

SAP on System z



Business Continuity for SAP on IBM System z

SAP on System z



Business Continuity for SAP on IBM System z

Note:

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 351.

Third Edition (October 2008)

This edition is a major update of *SAP on zSeries: High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33-8206-01. It applies to the following software components:

- SAP NetWeaver 7.0
- z/OS Release 1.8 (5694-A01) and higher supported z/OS Releases
- AIX Release 5.3 (5765-E61) and higher supported 5.x versions
- Linux on System z, formerly known as Linux on zSeries (for distribution details, see SAP Note 81737)
- Linux on System x, formerly known as Linux on xSeries
- IBM DB2 Universal Database for z/OS Version 8 (5625-DB2)
- IBM DB2 Universal Database for z/OS Version 9 (5625-DB2)
- IBM DB2 9 for z/OS (5635-DB2)
- IBM Tivoli System Automation for z/OS V3.2
- IBM Tivoli System Automation for Multiplatforms V3.1

and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

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 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

FAX (Germany): 07031-16-3456
FAX (Other Countries): (+49)+7031-16-3456

Internet e-mail: s390id@de.ibm.com
World Wide Web:
<http://www.ibm.com/servers/eserver/zseries/software/sap>
<http://www.ibm.com/servers/eserver/zseries/zos>

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 International Business Machines Corporation 2004, 2008.

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

Contents

Figures	ix
Tables	xi
Summary of changes	xiii
About this book	xv
Who should read this document	xv
Conventions and terminology used in this document	xv
Highlighting conventions	xvi
Syntax diagrams	xvii
Prerequisite and related information	xvii
How to send in your comments	xvii
Content of this document	xviii

Part 1. Concepts 1

Chapter 1. Introducing high availability and automation for SAP 3

High availability definitions	5
Degrees of availability	5
High availability	5
Continuous operation	6
Continuous availability	6
Types of outages	6
Planned outage	6
Unplanned outage	6
Tivoli System Automation's autonomic computing self-healing technologies	7
High availability and automation objectives for SAP	8
No planned outages	8
Failover support	9
Reduced operator errors	9
Health check for application problems	9
Overview of the high availability solution for SAP	9
High availability of an SAP system	9
Automation of an SAP system	10
Benefits of Tivoli System Automation	10

Chapter 2. Planning overview for high availability for SAP 13

Deciding which SA automation option you wish to implement	13
Option 1A: Automation via SA for z/OS, SA for MP, and SA AM (End-to-End): Variant A	14
Option 1B: Automation via SA for z/OS, SA for MP, and SA AM (End-to-End): Variant B	16
Option 2: Automation via SA for z/OS only	18
Technologies on System z used to gain highly available SAP solutions	19
1. Use DB2 data sharing on System z as database server	19

2. Configure the network	20
3. Exploit SAP features supporting high availability	20
4. Automate system operations for SAP	21
Making NFS highly available	23

Part 2. Database 25

Chapter 3. SAP availability benefits provided by System z 27

Features of the System z hardware architecture	27
Features of z/OS	27
Availability features and benefits with System z Parallel Sysplex	28
List of System z Parallel Sysplex availability features	28
Features of DB2 for z/OS	29
List of DB2 for z/OS availability features	29
List of DB2 for z/OS availability features with data sharing	35
DB2 data sharing	35
Non-disruptive software changes	35
DB2 for z/OS improvements	36
Group Buffer Pool (GBP) duplexing	36
Duplexing of SCA and lock structures	36
"Light" DB2 restart	36
SAP benefits and availability scenarios	37

Chapter 4. DB2 data sharing on System z Parallel Sysplex 39

Why Parallel Sysplex and data sharing for SAP?	39
Parallel Sysplex architecture	39
DB2 data sharing architecture	40
DB2 connection failover	41
Data sharing optimization for different SAP business applications	43

Chapter 5. Database architecture options 45

DB2 data sharing design options for SAP	45
Option 0: Single DB2 member with passive (inactive) standby member	46
Option 1: Two active DB2 members in active-standby mode	47
Option 2: Two active DB2 members, each with a passive standby member in the same LPAR	50
Option 3: Two active DB2 members, each with a passive standby member in an independent LPAR	51
How many data sharing groups?	51
How many sysplexes?	52
How many data sharing members?	52
Failover design	54

Chapter 6. Backup and recovery architecture in data sharing. 57

Data sharing considerations for disaster recovery.	57
Configuring the recovery site	57
Remote site recovery using archive logs	58
Using a tracker site for disaster recovery	60
Tracker site recovery	60
GDPS infrastructure for disaster recovery	60
Homogeneous system copy in data sharing.	64
Planning for homogeneous system copy in data sharing	64
Review of HSC in non-data-sharing	65
Requirements for data sharing	66
Designing homogeneous system copy in data sharing	66
Data sharing to data sharing.	67
Online copy design considerations.	67
Offline copy design considerations	68
Data sharing to non-data-sharing	68

Part 3. Network 69

Chapter 7. Network considerations for high availability 71

Introduction	71
General recommendations	72
Hardware considerations	72
z/OS communication software considerations	73
Considerations for the Linux on System z application server	73
Multiple Linux on System z guests under z/VM	74
DB2 connection failover recovery mechanism	75
OSPF protocol as a recovery mechanism.	76
Notes concerning AIX 5.3 and Path MTU discovery	77
Virtual IP Address (VIPA) as a recovery mechanism	77
Recommended setup for high availability connections between client and server	79
OSPF and subnet configuration aspects	79
VIPA and Source VIPA functions on remote application servers	80
Recommended setup for a high availability network	81
Alternative recovery mechanisms	83
z/OS VIPA usage for the high availability solution for SAP.	85
Timeout behavior of the client/server connection over TCP/IP	85
Timeout behavior of the AIX application server	86
Client connection timeout	86
Client transmission timeout	86
Recommended values	87
Client idle timeout	87
Recommended values	87
Timeout behavior of the Linux application server	88
Client connection timeout	88
Client transmission timeout	88
Recommended values	88
Client idle timeout	88
Recommended values	88

Timeout behavior of the Windows application server	89
Client connection timeout	89
Client transmission timeout	89
Recommended values	89
Client idle timeout	90
Recommended values	90
SAP maximum transaction time	90
Timeout behavior of the database server.	90
Server transmission timeout	90
Server idle timeout	91
DDF-specific keep-alive interval times	91
Resource timeout and deadlock detection interval.	91
Resource timeout	91
Deadlock detection interval	92

Part 4. SAP 93

Chapter 8. Concepts for a high availability SAP solution 95

Prerequisites and planning	95
SAP high availability installation	95
Sample two-node setup for a highly available SAP system	96
Architecture components	99
SAP Central Services	99
Old-style enqueue services with the central instance	100
Standalone enqueue server	100
Failover and recovery of SAP Central Services	101
Network	102
File system	106
Failover of the NFS server	107
Database	108
Non-data-sharing	108
Data sharing	109
Remote application server and DB2 connection failover support	109
General information	109
Failover with multiple DB2 members in the same LPAR when using DB2 Connect	110
Application design.	111
Failure scenarios and impact	111
Old-style central instance without data sharing	112
Data sharing, DB2 connection failover, double network (single central instance)	114
Enqueue replication and NFS failover: fully functional high availability	116

Chapter 9. Preparing a high availability SAP solution. 119

Software prerequisites	120
Naming conventions	121
Tivoli System Automation for z/OS	121
Conventions used in the SA z/OS policy	124
Tivoli System Automation for Multiplatforms	124
DB2	125

Using the SA z/OS 'DB2 - Best Practise Policy' to perform a light restart	125
File system setup	125
File systems	126
Setting up zFS filesystems	126
SAP directory definitions	127
SAP global transport directory.	127
SAP system-wide directories	127
SAP local directories	127
SAP administrator's home directory	128
SAPOSCOL/SAPCCMSR directory	128
NFS server on z/OS	128
How many NFS servers should I run?	128
Which NFS server security model (exports, safexp, or saf) should I use?	129
Security(exports)	129
security(safexp).	129
security(saf)	129
Mount handle databases and the remount site attribute.	130
The nlm site attribute.	130
NFS client root access	130
NFS Server automation	130
NFS Clients - General Information	130
NFS Client on Linux on System z	131
Tivoli System Automation	132
Setup of Tivoli NetView and Tivoli System Automation for z/OS.	132
Tivoli System Automation for Multiplatforms setup	132
Control of remote ABAP application server instances	132
SAP installation aspects	132
SAP license	133
SAP logon groups	133

Chapter 10. Customizing SAP for high availability. 135

Prerequisites.	135
Setting up an ABAP SCS instance and/or a Java SCS instance.	136
Rationale for enhancing the standard ASCS with additional SAP services	137
Rationale for not running the Enqueue Replication server as an ERS instance	137
General installation sequence	138
Installing and configuring ABAP SAP Central Services (ASCS)	139
Option 1: Installing ASCS with a virtual hostname.	139
Setting up Enqueue table replication.	141
Adding additional SAP services	142
Option 2: Installing ASCS with a physical hostname.	143
Installing SAP with classic CI	146
Verification of ABAP SCS with Enqueue Replication	148
Installing and configuring Java SAP Central Services (SCS)	149
1. Changes necessary if your system was installed with Option 2	150

2. Manually create the SCS instance directory on switch-over nodes.	152
3. Manually make the changes to enable enqueue table replication	153
Verification of Java SCS with Enqueue Replication	154
SAP profile parameters	155
Preparing SAP on z/OS for automation	157
C-shell and logon profiles	157
ABAP SAP Central Services (ASCS)	157
Java Central Services	158
ABAP application server instances	159
What the shell scripts do	159
startappsv_v5	159
stopappsv_v5	160
checkappsv_v4	160
rfcping	160
Remote execution	160
Remote control of Windows application servers	161
Java and double-stack application server instances	161
startjappsv	162
checkjappsv	163
saposcol	163
sapccmsr	163
Additional SAP setup for RFC connections	165
SAProuter	166
Summary of start, stop and monitoring commands	166
Other installation issues and recommendations	167

Part 5. System Automation 169

Chapter 11. Customizing Tivoli System Automation for z/OS. 171

Preparing SA z/OS for SAP high availability	171
Before you start	171
Setting initialization defaults for SA z/OS (AOFEXDEF)	171
Setting the region size for NetView to 2 GB	172
Sending UNIX messages to the syslog	172
Adapting the SA z/OS best practices policy for SAP	172
Overview of the resources	173
Description of the group structure	174
SAP system dependent groups	174
SAP infrastructure group	175
ABAP central services and enqueue replication server	177
Dependencies between the ABAP enqueue server and the enqueue replication server	178
Optional components of the ABAP central services	180
JAVA central services and enqueue replication server	180
DB2 policy	181
DB2 database server group (SAPHA1_DBX)	182
Classes	183

C_SAP_USS	183
SAP application servers	183
APAP-only application server	183
Double-stack (ABAP plus Java) application server	184
Java-only application server	185
SAP Application server groups	187
SAPHA1RAS	187
SAPHA1RASX	187
Overview of groups/applications.	189
Adding entries to the Automation Table	190
Adding the definitions for extension DFS/SMB	191
Additions to the SA z/OS policy	191
Application	191
DFS_SMB	191
Application group.	192
SMB_PLEX	192
Additions to the Automation Table for DFS/SMB	192

Chapter 12. Customizing the Tivoli System Automation for Multiplatforms (Base) 195

Introduction	195
Overview of Tivoli System Automation for Multiplatforms	196
Installing SA MP	197
Setting up SA MP cluster to manage SAP resources	199
Installing the high availability policy for SAP	199
Implementing Option 1A (Variant A)	201
Customizing the HA policy for a double-stack or Java-only SAP system	203
Step 1: Adapt the sample ABAP and Java configuration files	203
Step 2: Run the mksap script to create SA MP resources	205
Step 3: Perform a quick test of the SAP policy	206
Step 4: Save the policy	207
Step 5: Verify your SAP installation running under SA MP control	207
Implementing Option 1B (Variant B).	207
Making NFS highly available via SA MP	208
Run the NFS server in its own cluster	208
Special considerations for AIX.	209
Customize the sample SA MP high availability policy for SAP	209
Creating the SA MP resources for a SAP system	213
Customizing the high availability policy for a double-stack or Java-only SAP system	215
Step 1: Adapt the sample ABAP and Java configuration files	215
Step 2: Run the mksap script to create SA MP resources for components A, J and I	217
Step 3: Adapt the dependency file to create dependencies to the NFS server	219
Step 4: Run the commands in the dependency file	219
Step 5: Perform a quick test of the SAP policy	221

Step 6: Save the policy	221
Verify your highly available double-stack installation	221
Starting the SAP system	222
Verifying your high availability implementation with SA MP	223
Removing the HA policy	223
Using a tie breaker with SA MP	224
Using SA MP Quorums with Linux on System z under z/VM	225

Chapter 13. Customizing the Tivoli System Automation Application Manager (E2E) 227

Overview of end-to-end automation management	227
Sample high availability environment of the SAP on System z solution	228
Setting up the end-to-end product	230
Defining and installing the end-to-end high availability policy for SAP	237

Chapter 14. Change management. 241

Updating the SAP kernel	241
Updating the SAP kernel (release 6.40 or later)	242
Updating the SAP kernel	242
Updating the enqueue server or replication server, or changing the size of the enqueue table	242
Applying SAP/other maintenance when SAP is controlled by SA MP	243
Rolling kernel upgrade	243
Rolling update of DB2 Connect	244
Updating DB2 or z/OS	245

Part 6. Verification 247

Chapter 15. Verifying your implementation on z/OS. 249

Verification procedures and failover scenarios	249
Overview of the test scenarios.	249
Classification of the test scenarios	249
Test scenarios to verify the SA z/OS policy	250
Planned outage scenarios	250
Unplanned outage scenarios	250
Executed test scenarios	251
Planned outage scenarios	251
Unplanned outage scenarios	251
Test methodology	251
Purpose of the test	251
Expected behavior.	251
Setup of the test environment	252
Verification of resource status	252
Preparation for the test (unplanned outage only)	254
Execution of the test	256
Verifications after the test	257
Analyzing problems	257
Planned outage test scenarios	258
Stop and start of the entire SAP RED system	258

Startup of all LPARs one after the other	259
Shutdown and restart of an LPAR	260
Unplanned outage test scenarios	264
Failure of the enqueue server	264
Failure of the message server	267
Failure of the NFS server	269
Failure of a TCP/IP stack	270
Failure of an LPAR	273
Problem determination methodology	276
SA z/OS problem determination	276
NetView netlog.	276
z/OS syslog.	277
Message Processing Facility	277
Problem determination in SA z/OS	278
SDF or NMC	278
DISPINFO	278
INGINFO	279
UNIX messages	280
If nothing happens	280
When you are really lost	280
Getting help from the Web	281
Where to check for application problems	281
Checking the network	282
Checking the configuration.	282
Checking network devices	283
Dynamic VIPA	283
Routing tables and OSPF	283
Checking active connections	284
Checking the status of the Shared HFS and of NFS	284
Checking the status of DB2 and SAP connections	285
Check that DB2 is running	285
Check the SAP database connections	285
Availability test scenarios	286

Chapter 16. Verifying your implementation on Linux/AIX 289

Verification procedure and failover scenarios	289
Test setup	289
Scenarios	289
Testing an unplanned outage of an ABAP SCS	293
Testing an unplanned outage of a Java SCS	294

Part 7. Appendixes 295

Appendix A. Network setup 297

Network hardware components for the test setup	297
Networking software components for the test setup	298
z/OS network settings for the test setup	298
z/OS VIPAs	298
z/OS UNIX System Services setup - BPXPRMxx	298
z/OS LPAR SC42	298
File /etc/resolv.conf - SC42.	298
TCP/IP profile - SC42	300
OMPROUTE started task - SC42	302
ENVVARS - SC42	303
OSPF routing parameters - SC42	303

Linux on System z network settings for the test setup	306
Quagga setup - OSPF.	306
Zebra setup - Zebra	306
AIX OSPF definitions for the 'gated' daemon	307
Domain Name Server (DNS) definitions	307
Static VIPA definitions required for SLES-10	308

Appendix B. File system setup 311

NFS server procedure	311
NFS export file	311
NFS attribute file	313
Mount commands	313

Appendix C. Sample REXX sanity-check procedure 315

Sample REXX procedure.	315
SANCHK.	315

Appendix D. Description of the z/OS high availability scripts 319

Script availability	319
Script descriptions.	320
startappsrv_v5	320
stopappsrv_v5	321
checkappsrv_v4	322
sapctrl_em	322
startjappsrv	322
checkjappsrv	323

Appendix E. Sample Tivoli System Automation for Multiplatforms high availability policy for SAP 325

The ABAP SAP Central Services group (ASCS)	325
The ABAP ENQREP group for ASCS	326
The Java SAP Central Services group	326
The ENQREP group for the Java SCS	326
The application server groups	326
The SAProuter / SAP Web Dispatcher group	327
Interaction between ES and ERS	328
Application server (AS) resources	328
GetWebPage.	330
Service IP addresses (or VIPAs)	331
Setup scripts	331
Automation scripts	332
Removing the policy (rmsap)	332

List of abbreviations 335

Glossary 341

Bibliography. 347

IBM documents	347
SAP documents	349
SAP Notes	349
APARs	350

Notices 351

Trademarks and service marks 351

Index 355

Figures

1. The concept of autonomic computing	4	36. High-Availability System option within SAPinst	140
2. Causes of application downtime and appropriate response	7	37. SAP profile parameters relevant for the high availability solution	155
3. The closed loop of automation	8	38. Defining the gateway host for rfcoscol with SAP transaction SM59.	166
4. Option 1A: Automation via SA for z/OS, SA for MP, and SA AM (End-to-End), Variant A	14	39. SA z/OS best practices policy for SAP	174
5. Option 1B: Automation via SA for z/OS, SA for MP, and SA AM (End-to-End), Variant B.	16	40. Group SAPHA1_X belonging to SAP system HA1	175
6. Option 2: Automation via SA for z/OS only	18	41. SAP-system-independent groups and resources	176
7. System z Parallel Sysplex architecture elements	40	42. Lowest level in group structure of ABAP central services and enqueue replication server	177
8. DB2 data sharing in a Parallel Sysplex	41	43. Lowest level in group structure of JAVA central services and enqueue replication groups	181
9. SAP sysplex failover configuration: Option 0 example.	42	44. DB2 Best Practice Policy – adapted for SAP system HA1	182
10. Option 0: Single DB2 member with passive (inactive) standby member	46	45. SAPHA1RASX application group	188
11. Option 1: Two active DB2 members in active-standby mode	47	46. Overview of the resources	189
12. Large company using architecture options 0 and 1	49	47. Overview of the relationships between elements of the SAP policy (excluding DB2 elements)	190
13. Option 2: Two active DB2 members, each with a passive standby member in the same LPAR	50	48. SMB_PLEX application group	192
14. Option 3: Two active DB2 members, each with a passive standby member in an independent LPAR	51	49. Initial Setup: Linux on System z cluster	197
15. Example of high availability with GDPS configuration	62	50. Overview of the SAP policy definitions for double-stack system	202
16. Process for obtaining a non-disruptive volume backup without the BACKUP SYSTEM utility of DB2 V8	63	51. Entries to be adapted in the ABAP_instances.conf file.	204
17. Sample VSWITCH utilization	75	52. Entries to be adapted in the J2EE_instances.conf file	205
18. VIPA and OSPF recovery mechanisms under z/OS.	78	53. mksap options	205
19. Recommended setup for a high availability network.	82	54. Overview of the SAP policy definitions for double-stack system (ABAP part)	211
20. System setup with z/OS ARP takeover and Windows adapter teaming	84	55. Overview of the SAP policy definitions for double-stack system (Java part)	212
21. Sample two-node TSA domain	98	56. Option 1B: Linux on System z cluster	214
22. SAP enqueue services with the old central instance concept	100	57. Entries to be adapted in the ABAP_instances.conf file.	216
23. Initial startup of SCS	101	58. Entries to be adapted in the J2EE_instances.conf file	217
24. Failure of SCS and recovery of the enqueue table	102	59. mksap options	218
25. Movement of the enqueue replication server	102	60. List of SAP resources in SA MP policy	223
26. General concept of a fault-tolerant network	103	61. Landscape with sample HA solution of SAP on System z	229
27. Alternative paths in a duplicated network	104	62. Best Practice SAP policy adapted for SAP system HA1	231
28. Rerouting if a network adapter card fails	104	63. Groups and resources specific to SAP system HA1	232
29. Rerouting in a sysplex even in case of two failing network cards	104	64. SAP infrastructure group.	233
30. VIPA takeover and dynamic routing	105	65. ABAP central services and enqueue replication groups	234
31. Initial NFS client/server configuration	107	66. JAVA central services and enqueue replication groups	234
32. Failover of the NFS server	108		
33. Failover setup using DB2 Connect, with multiple DB2 members in the same LPAR	110		
34. High availability solution configuration for SAP.	120		
35. SAP directory structure and file systems	126		

67.	DB2 sysplex group – adapted for SAP system HA1	235	73.	Error handling menu	255
68.	Linux on System z cluster	235	74.	Enqueue test: start mass enqueue operations	256
69.	Overview of the end-to-end policy definitions	237	75.	List of entries in the enqueue table	256
70.	SA Operations Console view on SAP System Topology	238	76.	SAP system log (SM21)	267
71.	Graphical view of the SAP SA end-2-end policy	239	77.	SAP system log (SM21)	269
72.	SM12 primary panel	255	78.	Results of SDSF DA command	285
			79.	Results of DB2 Display Thread command	286
			80.	Networking configuration for the high availability solution for SAP	297

Tables

1. Parallel Sysplex availability features matrix	28	21. Examples of test scenarios	249
2. DB2 for z/OS availability features matrix	29	22. Stop of the entire SAP system with SA z/OS	258
3. Large company using architecture option 2	54	23. Start of the entire SAP system with SA z/OS	258
4. Recovery attributes of the recommended setup	82	24. Startup of the first LPAR.	260
5. Retransmission intervals	87	25. Startup of the second LPAR.	260
6. Simple configuration	112	26. Shutdown of the LPAR where the ES and NFS servers are running	260
7. DB2 sysplex data sharing configuration with double network	114	27. Restart of the LPAR where the ES and NFS servers are running	261
8. Fully implemented high availability solution for SAP	116	28. Failure of the enqueue server	265
9. Software requirements for the HA solution	120	29. Failure of the message server	267
10. SAP application server for Linux on System z	121	30. Failure of the NFS server	269
11. Recommended names for all z/OS-related components of a SAP system	122	31. Failure of a TCP/IP stack	271
12. Recommended names for all components of an individual SAP system	123	32. Failure of the LPAR where the ES and NFS servers are running	274
13. Naming conventions for SA z/OS resources	124	33. High availability test scenarios.	286
14. Using sapctrl_em to manage ABAP SCS and its resources	158	34. Planned outages	289
15. Using sapctrl_em to manage JAVA SCS and its resources	159	35. Unplanned outages	292
16. Summary of start/stop monitoring commands	166	36. Scripts for SA z/OS	319
17. Messages and User Data section from the SAPHA1ACV policy definition.	179	37. Setup scripts for the SA MP high availability policy for SAP	331
18. Startup section from the SAPHA1AER policy definition	179	38. Delivered SA MP automation scripts	332
19. Relationships section from the SAPHA1AER policy definition	179	39. Selected IBM Tivoli System Automation for z/OS publications	347
20. Components of the SA MP high availability policy for SAP	200	40. Selected IBM Tivoli System Automation for Multiplatforms publications.	347
		41. Other IBM reference documents	348
		42. IBM Redbooks and Redpapers covering related topics	348
		43. SAP publications	349
		44. Relevant SAP Notes	349

Summary of changes

The third edition of the manual (October 2008) contains this new information:

- Has a new title: *Business continuity for SAP on IBM System z*
- Is restructured and new titles given to some chapters and parts.
- Is based upon DB2 z/OS V9.1 and z/OS V1.9.
- Is based upon SAP NetWeaver 7.0.
- Is based upon the Tivoli System Automation's DB2 "Best Practise" policy that is supplied with APAR **OA26776** .
- Includes the use of a new script `sapctrl_em` which is used to start SAP resources.
- Contains a new chapter "Planning overview for high availability for SAP", which has details of the three *recommended* configuration options you should choose from (Options 1A, Option 1B, or Option 2).
- Includes the use of new versions (V5.3) of scripts for these components:
 - SA z/OS
 - SA MP
 - SA AM (end-to-end)

Summary of changes

About this book

This book describes the *IBM High Availability Solution for SAP on System z*, which provides the means for fully automating the management of all SAP components and related products running on z/OS, AIX®, or Linux®. The automation software monitors all resources and controls the restart and/or takeover of failing components, thereby ensuring near continuous availability of the SAP system.

This book has been updated to cover SAP Netweaver 7.0.

Major portions of this book were derived from the following publications by the IBM International Technical Support Organization:

- *SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations*, SG24-5690
- *SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution Using System Automation*, SG24-6836
- *SAP on DB2 for z/OS and OS/390: High Availability and Performance Monitoring with Data Sharing*, SG24-6950
- *mySAP Business Suite Managed by IBM Tivoli System Automation for Linux*, REDP-3717

The original documents are available at:

<http://www.redbooks.ibm.com>

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.

Conventions and terminology used in this document

In this document, the following naming conventions apply:

- IBM DB2 for z/OS is usually referred to as *DB2*.
- The SAP on DB2 for z/OS system is usually 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 5.x or AIX 6.x are usually referred to as *AIX*.
- Linux on System z (64-bit) is usually referred to as *Linux*.
- The term "Windows" is used to encompass Windows Server 2003 and its supported successors (32-bit version).
- The term "currently" refers to this document's edition date.
- The IBM products Tivoli System Automation for z/OS (SA z/OS), formerly known as System Automation for OS/390 (SA OS/390), and Tivoli System Automation for Multiplatforms (SA MP), formerly known as Tivoli System Automation for Linux (SA for Linux) are referred to collectively in this document as *Tivoli System Automation (SA)*.
- Tivoli System Automation Application Manager is referred to as *SA AM*.
- The term NetView® refers to the IBM product Tivoli NetView for z/OS (formerly Tivoli NetView for OS/390).

- DB2 documentation is usually cited in the text without a specific release or order number, since these numbers are different for DB2 V8 and V9. Refer to “Bibliography” on page 347 for specific information.
- *Planning Guide*:
 - *SAP Planning Guide for SAP NetWeaver on IBM DB2 for z/OS*.
 - *SAP NetWeaver 7.0 SR3* (or later).
 - Full titles for these publications (and numbers where applicable) are provided in SAP documents.
- USS refers to z/OS UNIX system services.
- The SAP documentation that is specific to the database implementation is usually referred to as the *SAP Database Administration Guide* (see “SAP documents” on page 349 for full titles). This is not to be confused with the IBM DB2 *Administration Guide* publication.
- The term *SAP installation guides* refers to the release-specific SAP installation documentation (“SAP documents” on page 349).
- 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, employing ASCS and *Java SCS* to designate the environment-dependent instances when required. See “SAP Central Services” on page 99.

Highlighting conventions

Italics are used for:

- document titles
- emphasis
- options, variables and parameters

Boldface is used for:

- check box labels
- choices in menus
- column headings
- entry fields
- field names in windows
- menu-bar choices
- menu names
- radio button names
- spin button names

Monospace is used for:

- coding examples
- commands and subcommands
- entered data
- file names
- group and user IDs
- message text
- path names

Underlined settings are:

- default values

Bold italics are used for:

- recommended values

Syntax diagrams

This document uses railroad syntax diagrams to illustrate how to use commands. This is how you read a syntax diagram:

A command or keyword that you must enter (a required command) is displayed like this:



An optional keyword is shown below the line, like this:



A default is shown over the line, like this:



An item that can be repeated (meaning that more than one optional keyword can be called) is shown like this:



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, z/OS, UNIX[®] System Services, RACF[®], JCL, RMF[™], WLM)
- DB2 administration (for example, SQL, SPUFI, and the utilities REORG and RUNSTATS)
- AIX, Linux on System z, or Windows (or all)

Refer to “Bibliography” on page 347 for a list of related documentation.

Additional information is available from SAP as part of the help system:

<http://help.sap.com>

How to send in 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 our home page at
<http://www.ibm.com/servers/eserver/zseries/software/sap>

Click on "Contact" at the bottom of the page.

- Send your comments by e-mail to s390id@de.ibm.com. Be sure to include the document's name and part number, the version of z/OS, and, if applicable, the specific location of the passage you are commenting on (for example, a page number or table number).
- 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 the activities that need to be completed before the actual SAP installation via the SAP system installation tool can be started, and administrative tasks that may have to be performed repeatedly during the lifetime of the system. Chapter descriptions follow below:

Part 1, "Concepts," on page 1

This part provides general information on high availability in a SAP environment.

Part 2, "Database," on page 25

This part lists the availability benefits provided by the System z hardware, z/OS, and DB2, discusses DB2 data sharing, identifies architecture options, and describes backup and recovery in a data sharing environment.

Part 3, "Network," on page 69

This part 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.

Part 4, "SAP," on page 93

This part 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.

Part 5, "System Automation," on page 169

This part discusses the customization of SA z/OS and the base and end-to-end management automation components of SA MP. It also discusses issues in updating and upgrading the system components.

Part 6, "Verification," on page 247

This part addresses how to confirm that the high-availability implementation is correct on z/OS and Linux, and, if not, how to determine where the problems lie and how to resolve them.

Part 7, “Appendixes”

Provide setup details for networking, file systems, and Tivoli System Automation. Also available are detailed descriptions of the scripts that support high availability on z/OS and how to obtain updates, and details of the high availability policy for SAP used with SA MP.

“List of abbreviations” on page 335

Contains a list of important abbreviations appearing in this document.

“Glossary” on page 341

Explains the meaning of the most important technical terms employed in this document.

“Bibliography” on page 347

Contains lists of the IBM and SAP documentation referred to elsewhere in this document, including SAP Notes and APARs.

Part 1. Concepts

Chapter 1. Introducing high availability and automation for SAP	3
High availability definitions	5
Degrees of availability	5
High availability	5
Continuous operation	6
Continuous availability	6
Types of outages	6
Planned outage	6
Unplanned outage	6
Tivoli System Automation's autonomic computing self-healing technologies	7
High availability and automation objectives for SAP	8
No planned outages	8
Failover support	9
Reduced operator errors	9
Health check for application problems.	9
Overview of the high availability solution for SAP.	9
High availability of an SAP system	9
Automation of an SAP system	10
Benefits of Tivoli System Automation.	10
Chapter 2. Planning overview for high availability for SAP	13
Deciding which SA automation option you wish to implement.	13
Option 1A: Automation via SA for z/OS, SA for MP, and SA AM (End-to-End): Variant A	14
Option 1B: Automation via SA for z/OS, SA for MP, and SA AM (End-to-End): Variant B.	16
Option 2: Automation via SA for z/OS only	18
Technologies on System z used to gain highly available SAP solutions	19
1. Use DB2 data sharing on System z as database server	19
2. Configure the network	20
3. Exploit SAP features supporting high availability	20
4. Automate system operations for SAP	21
Making NFS highly available	23

Chapter 1. Introducing high availability and automation for SAP

The Business Continuity of a SAP production system is a critical business factor. It requires the highest possible level of System availability. The solution documented in this book:

- combines high availability techniques with automation technologies of IBM Tivoli System Automation products.
- helps to avoid unplanned outages by eliminating *single points of failure*.
- helps to avoid *planned outages* such as administrative or maintenance work.
- provides Business Continuity for a SAP production system as close as possible to 24x365.

IBM Systems products incorporate a variety of advanced autonomic computing capabilities 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 re-configure autonomously both at boot time and during run time. This action can be initiated by the need to adjust the allocation of resources based on the current optimization criteria or in response to hardware or firmware faults. 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, platforms can detect hardware and firmware faults instantly and then contain the effects of the faults within defined boundaries. This allows platforms 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

This allows computing systems to protect against internal and external threats to the integrity and privacy of applications and data.

These four components are illustrated in the following graphic:



Figure 1. The concept of autonomic computing

Since the initial announcement of SAP on DB2 for z/OS, we have used DB2 Parallel Sysplex® data sharing combined with *DB2 connection failover* to remove the database server as a single point of failure. This also gave customers the ability to avoid planned and unplanned outages of the database server. See “Remote application server and DB2 connection failover support” on page 109.

The high availability solution presented in this book further enhances this capability by removing the SAP central instance as a single point of failure and providing a means to automate the management of all SAP components for planned and unplanned outages. This is achieved by combining the concepts of system automation and transparent failover in a Parallel Sysplex. Based on the IBM Tivoli System Automation (SA) products, together with a redesign of the SAP central instance concept, this high availability solution exploits the SAP standalone enqueue server, the enqueue replication server, dynamic virtual IP addresses (VIPA), shared file system, and DB2 data sharing to guarantee a minimum of SAP system outages along with a maximum of automation.

The implementation and customization of the complete HA solution highly depends on the customer configuration and requires SA skill. We strongly recommend that customers request support from IBM Global Services. Before going live, customers 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 running on z/OS, AIX, Windows, or Linux. The automation software monitors all resources and controls the restart and/or takeover of failing components, thereby ensuring near continuous availability of the SAP system.

The availability of the enqueue server is *critical* for an SAP system. If it fails, most SAP transactions will also fail. To address this single point of failure, SAP, in close cooperation with IBM, has changed the architecture of the enqueue 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 standalone process called the standalone enqueue server (which operates under the designation *SAP Central Services*, or SCS). The enqueue server transmits its replication data to an enqueue replication server, which normally resides on a different system. The enqueue replication server stores the replication data in a shadow enqueue table that resides in shared memory. The SAP developer network (SDN) has more information about this and about HA.

“SAP Central Services” on page 99 and the SAP publication *SAP Lock Concept*, which can be found via the SAP Marketplace Web page <http://service.sap.com/ha>

as follows: in the navigation pane open ‘High Availability’, ‘HA in Detail’, ‘Standalone enqueue server’, and then click on the link ‘SAP Lock Concept’.

When SAP introduced Java instances with SAP Netweaver 04, a similar central services mechanism has been available right from the beginning. The SAP installation program SAPInst under NetWeaver '04 supports the installation of a Java SCS for Java-only systems. Under NetWeaver 2004s, it supports the installation of an ABAP SCS (ASCS) for ABAP-only systems and both variants for ‘double-stack’ systems.

If the enqueue server fails, it is quickly restarted by Tivoli System Automation and uses the replicated data in the shadow enqueue table to rebuild the tables and data structures. This means that a failure of the enqueue server is transparent to the end user and the SAP application. For a more detailed description of this process, see Chapter 8, “Concepts for a high availability SAP solution,” on page 95.

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

The solution is applicable to a homogeneous z/OS environment as well as to a heterogeneous environment. The described implementation assumes that the database runs on z/OS. However, similar configurations are possible with DB2 or Oracle running on Linux or AIX. Of course, the overall availability characteristics depend heavily on the chosen hardware, operating system and database system.

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

High availability definitions

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

Degrees of availability

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

High availability

High availability refers to the ability to avoid unplanned outages by eliminating single points of failure. This is a measure of the reliability of the hardware, operating system, and database manager software. Another measure of high availability is the ability to minimize the effect of an unplanned outage by masking the outage from the end users. This can be accomplished by quickly restarting failed components using a tool such as SA 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 end users. This is accomplished by providing multiple servers and switching end users to an available server at times when one server is made unavailable. Using DB2 data sharing with DB2 connection failover is an example of how this is accomplished in an SAP environment. Part 2, "Database," on page 25 describes how a number of planned outages can be avoided by taking advantage of DB2 data sharing and DB2 connection failover.

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

Continuous availability

Continuous availability combines the characteristics of high availability and continuous operation to provide the ability to keep the SAP system running as close to 24x365 as possible. This is what most customers want to achieve.

Types of outages

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

Planned outage

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

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

Unplanned outage

Unplanned outages are unexpected outages that are caused by the failure of any SAP system component. They include hardware failures, software issues, or people and process issues.

In a report issued by Gartner Research, *Enterprise Guide to Gartner's High-Availability System Model for SAP*, R-13-8504 (December 2001), they discuss the causes of application downtime (see Figure 2 on page 7). According to Gartner, one-fifth of unplanned outages result from hardware failure, network components, operating system problems, or environmental problems. In the case of hardware or software failures, the reliability and resilience of these components determines the impact of unplanned outages on the SAP system.

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

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

longer than planned.

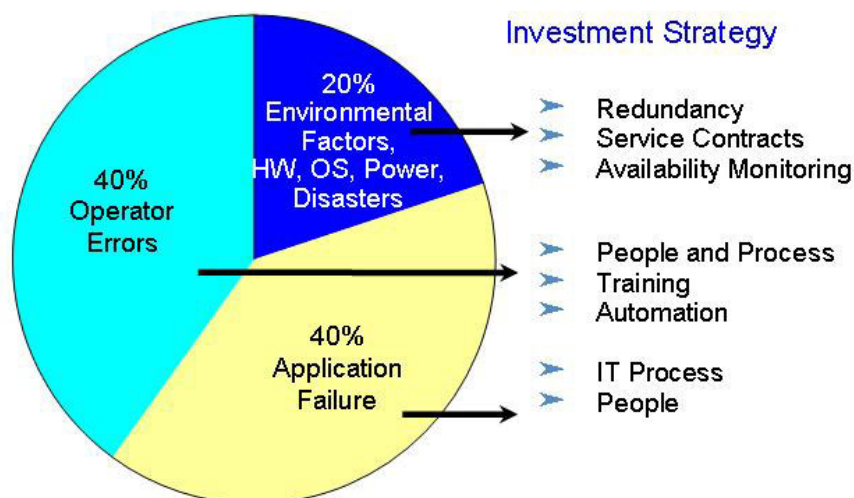


Figure 2. Causes of application downtime and appropriate response

Tivoli System Automation's autonomic computing self-healing technologies

In order to avoid all causes of outages, the high availability solution uses the autonomic computing self-healing technologies 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 availability goals to decide on corrective actions within the right context. Today, Tivoli System Automation manages availability of business applications running in single systems and clusters on z/OS and Linux on System z (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:

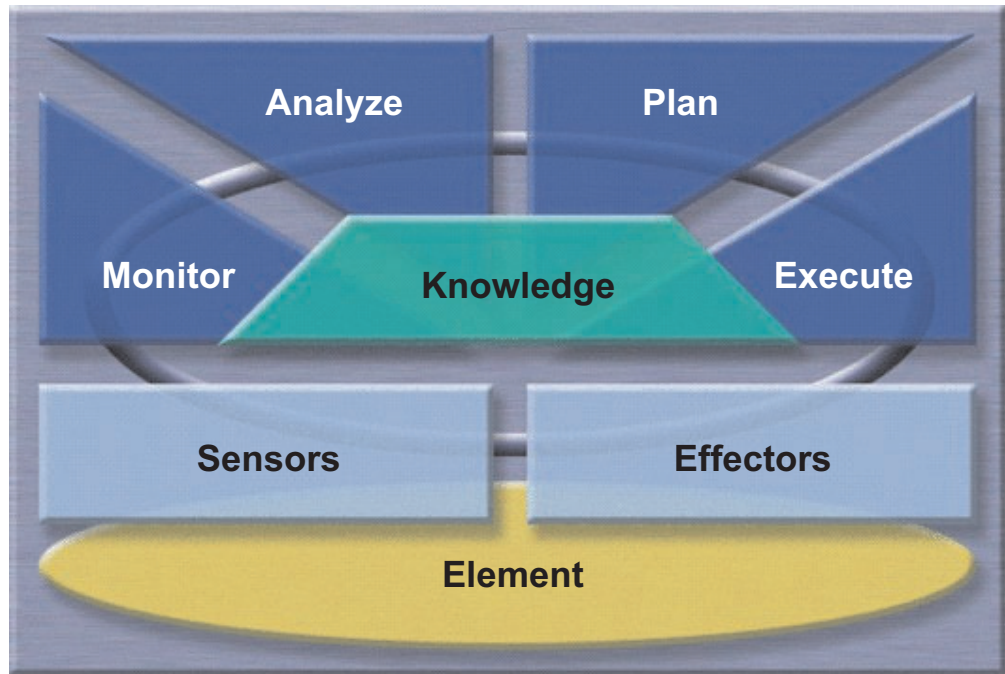


Figure 3. The closed loop of automation

Resource elements are monitored via 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 (which represents its knowledge) to deduce a plan to bring the resource and entire system into the desired state. The plan is executed 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 by:

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

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 which allows a “cascade” of moves. The procedures for doing this are documented in Part 2, “Database,” on page 25.

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

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

Failover support

The high availability solution for SAP has always had a failover capability for remote application server instances using Parallel Sysplex data sharing and DB2 connection failover. Because of the enqueue server, SCS can be moved or restarted transparent to the end users. SAP work processes automatically reconnect to SCS without failing any transactions.

Reduced operator errors

The high availability solution for SAP uses SA to automate the starting, stopping, and monitoring of all SAP components. By automating daily operations, there is less opportunity for error when starting or stopping SAP components. SA provides the ability to 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. SA also checks that a component is not stopped if there are child components still active. This ensures that an orderly start or stop of the SAP system is accomplished with little opportunity for operator error. See Chapter 10, “Customizing SAP for high availability,” on page 135 for a description of how this is set up.

Note that SA MP automates only those SAP components running outside of z/OS.

Health check for application problems

SAP provides a utility, *rfcping*, to monitor the status of *ABAP application servers* (AS). The high availability solution for SAP uses SA to invoke this *rfcping* monitoring task. For a more detailed description of *rfcping*, see “*rfcping*” on page 160.

IBM provides a simple utility, *GetWebPage*, to monitor the status of *Java application servers* in the same way. The monitor task *GetWebPage* tests the access to the main index Webpage of the Java application servers at regular intervals. Providing the access works, the monitor runs. If this Webpage is no longer accessible, the *GetWebPage* monitor stops. This signals to SA that the Java application server instance is down, because SA monitors the *GetWebPage* monitor.

For a more detailed description of *GetWebPage*, see “*GetWebPage*” on page 330.

Overview of the high availability solution for SAP

High availability of an SAP system

As described in “High availability” on page 5, elimination of single points of failure is required. We use DB2 data sharing to remove the database server as a single point of failure. Now, with SCS, the enqueue server has been removed as a single point of failure. The high availability solution for SAP also adds a movable NFS server and dynamic virtual IP addressing (under z/OS only) for moving application components. SA is used to monitor these components and quickly restart them if they should fail.

Automation of an SAP system

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

SA MP automates those SAP components which run outside of z/OS, that is SAP ABAP and/or Java application servers. Additionally, if you decided to run SAP Central Service and/or NFS server outside of z/OS *under Linux/AIX* (as described in “Option 1B: Automation via SA for z/OS, SA for MP, and SA AM (End-to-End): Variant B” on page 16), SA MP also automates them.

As the nature of the SAP on System z solution is heterogeneous, we recommend using *Tivoli System Automation Application Manager (SA AM)* as “glue” between the composite SAP application.

Benefits of Tivoli System Automation

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

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

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

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

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

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

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

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

More information on SA can be found on the Web at:

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

<http://www.ibm.com/software/tivoli/products/sys-auto-multi>

Introduction

Chapter 2. Planning overview for high availability for SAP

The solutions provided in this book use the autonomic computing technologies of IBM Systems products to provide automation and high availability for SAP systems. The purpose of this chapter is to provide an overview of planning for high availability and to indicate the appropriate section where further information and examples can be found. It is intended to give you a first overview of what has to be considered when making an SAP system highly available.

High availability means the ability to avoid planned and unplanned outages by eliminating single points of failure. This is a measure of the reliability of the hardware, operating system, and database manager software.

Another measure of high availability is the ability to minimize the effect of an unplanned outage by masking the outage from the end users. This can be accomplished by quickly restarting failed components using a tool from the *System Automation (SA) product family*. SA 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 an planned or unplanned outage takes place.

Deciding which SA automation option you wish to implement

This section provides an overview of the preferred configurations you can select when planning your high availability solution.

Option 1A: Automation via SA for z/OS, SA for MP, and SA AM (End-to-End): Variant A

In the first variant of Option 1, the *complete SAP environment* is automated using SA for z/OS, SA for MP (Multiplatform), and SA AM (End-to-End).

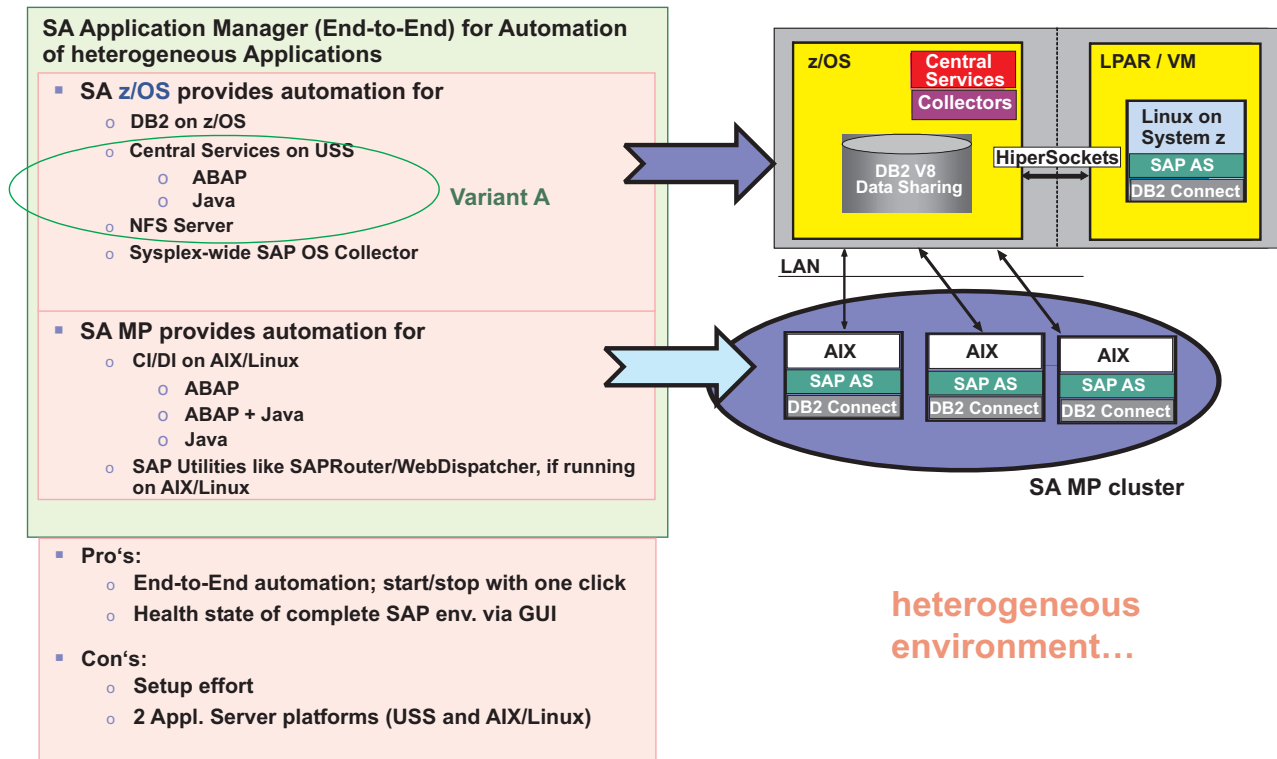


Figure 4. Option 1A: Automation via SA for z/OS, SA for MP, and SA AM (End-to-End), Variant A

Option 1A is the option that IBM recommends you should implement.

Option 1A lets *System Automation for z/OS* (SA z/OS) control and automate *all* SAP-related components which are supported to run under z/OS or USS:

- DB2 z/OS data sharing members
- SAP Central Services (for ABAP and/or for Java)
- NFS Server
- Sysplex-wide SAPOSCOL
- Other SAP utilities capable of running under z/OS or USS

Non-z/OS SAP resources like Application servers and other utilities (with an affinity to those Application servers) are automated in a cluster managed by *System Automation for Multiplatforms*.

As a complete SAP on System z solution is heterogeneous by nature, it is recommended to use *System Automation Application Manager* to create an End-to-End automation and high availability solution. End-to-End automation allows the definition of relations between the platforms:

- The Application servers under AIX/Linux are started after the NFS server running on z/OS is up, and if it is down it would be started automatically.
- The health status of the whole SAP solution can be verified from a *single point* in ISC (see Figure 70 on page 238). The SAP solution works well when no warning

sign is next to SAPHA1 icon. You should investigate when a warning sign is displayed. An operator can start or stop the complete SAP solution by clicking that SAPHA1 icon.

For details, see Chapter 13, “Customizing the Tivoli System Automation Application Manager (E2E),” on page 227.

Please note that setting up this option requires that you setup and configure:

- SA z/OS
- SA MP
- SA AM

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

- SAP Central Services on z/OS USS, and
- the Application server on AIX or Linux.

Option 1B: Automation via SA for z/OS, SA for MP, and SA AM (End-to-End): Variant B

In the second variant of Option 1, the *complete SAP environment* is again automated using SA for z/OS, SA for MP (Multiplatforms), and SA AM (End-to-End).

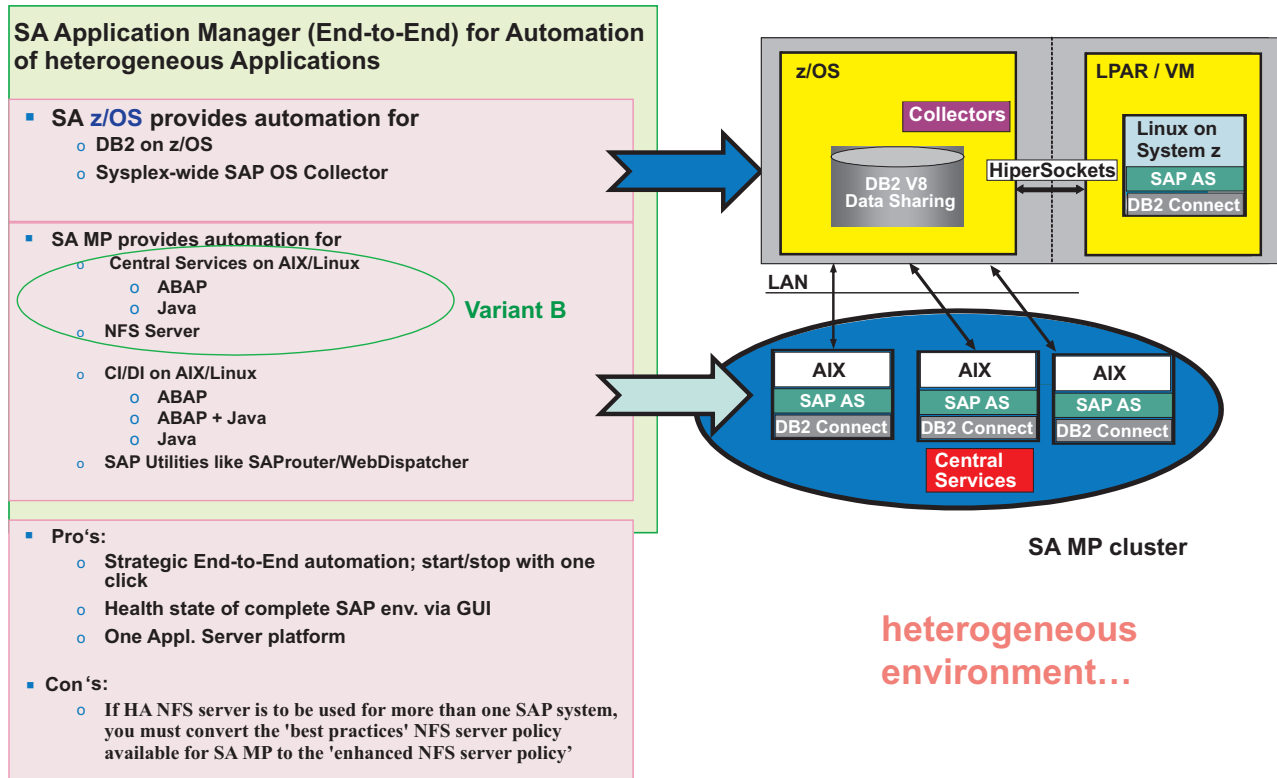


Figure 5. Option 1B: Automation via SA for z/OS, SA for MP, and SA AM (End-to-End), Variant B

Option 1B differs from 1A by having the SAP Central Services and NFS server run under the control of System Automation for MP.

In Option 1B, SA z/OS manages:

- DB2 z/OS data sharing members
- Sysplex-wide SAPOSCOL (optional)
- Other SAP utilities capable of running under z/OS or USS.

In Option 1B, SA MP manages the following SAP components:

- SAP Central Services (for ABAP and/or for Java)
- Application Servers
- NFS Server

In addition, other utilities (with an affinity to Application servers) are automated in a cluster managed by SA MP.

As a complete SAP on System z solution is heterogeneous by nature, you must use System Automation Application Manager to create an End-to-End automation solution:

- End-to-End automation allows the grouping of resources from different platforms.
- All z/OS and AIX/Linux resources which belong to one SAP system can be grouped together within End-to-End.

- The health status of the whole SAP solution can be verified from a *single point* in ISC (see Figure 70 on page 238). The SAP solution works well when no warning sign is next to SAPHA1 icon. You should investigate when a warning sign is displayed. An operator can start or stop the complete SAP solution by clicking that SAPHA1 icon.

For details, see Chapter 13, “Customizing the Tivoli System Automation Application Manager (E2E),” on page 227.

Regarding the highly available NFS server there are *two* setup options:

- The first option is to run for each SAP system its ‘own’ NFS server within the SA MP cluster for SAP.
- The second option is to define a *separate* SA MP NFS cluster serving more than one SAP system. This option should be implemented using the ‘enhanced NFS Server’ best practices policy.

Setting up option 1B requires that you setup and configure SA z/OS, SA MP and SA AM.

This configuration implies that *all SAP components* (SAP Central Services and the Application server) run on *one platform*: AIX or Linux.

Option 2: Automation via SA for z/OS only

In Option 2, the environment is automated using SA for z/OS only.

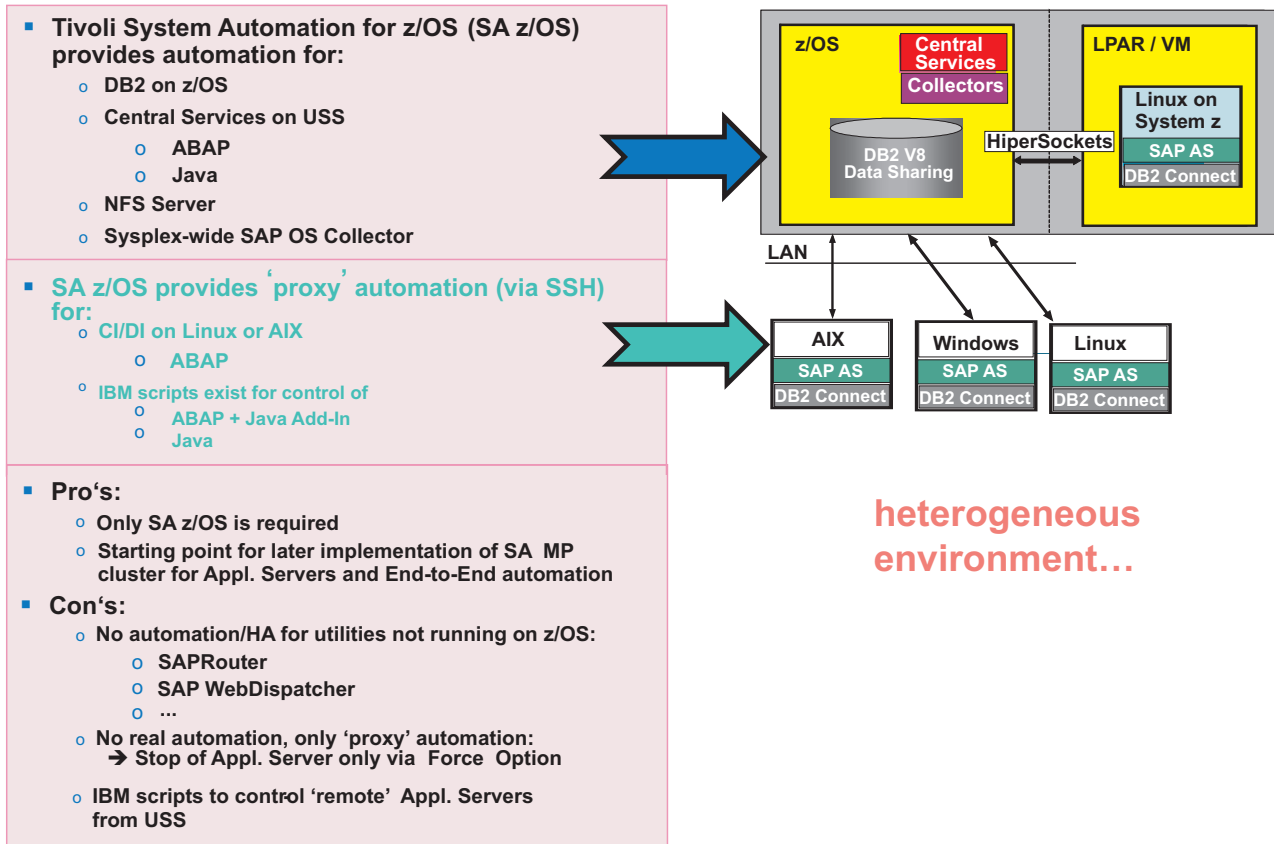


Figure 6. Option 2: Automation via SA for z/OS only

The ABAP/JAVA application servers are controlled via "proxy" resources. The Option 2 setup can be seen as an "intermediate solution" on a path to implementing the recommended option 1A.

Options 2 allows SA for z/OS to control and automate all SAP-related components which can run under z/OS or USS. Furthermore, it controls 'remote' Application server instances via 'proxy' resources. This means, the Application server resources under SA z/OS:

- issue (for example) SSH commands to the 'remote' machines to start and stop the Application servers, and
- use TCP/IP-based 'monitoring' via the rfcping and GetWebPage utilities (see "ABAP application server instances" on page 159 and "Java and double-stack application server instances" on page 161).

Therefore, SA z/OS controls:

- DB2 z/OS data sharing members
- SAP Central Services (for ABAP and/or for Java)
- Application Server on AIX/Linux via 'proxy' resources
- NFS Server
- Sysplex wide SAPOSCOL
- Other SAP utilities that can run under z/OS or USS

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 used to control the Application servers on AIX/Linux are quite simple compared to the SA MP scripts. The SA MP scripts include far more 'complex logic' that is used to distinguish between short living error situations and more severe errors.

This option requires that you have to set up and configure *SA z/OS only*.

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

- SAP Central Services on z/OS USS, and
- the Application server on AIX or Linux.

Technologies on System z used to gain highly available SAP solutions

There are 4 steps in planning and implementing a high availability solution:

1. You must use DB2 on System z in datasharing mode as 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) like Cisco's Virtual Switching System.
 - Software-based, by using features like z/OS VIPA (Virtual IP Address) and dynamic routing (OSPF) to get a fault-tolerant, highly available network.
3. Exploit SAP features supporting high availability such as:
 - 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 notify the operator accordingly

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

When implementing 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. This guarantees redundancy of the database at the highest level. Additionally, online backup and online recovery features, coupled with the features of System z platform, provide high availability as well as DB2 rolling maintenance and release upgrade. For further information on these features, see Chapter 3, "SAP availability benefits provided by System z," on page 27.

This manual describes *four* major data sharing on System z parallel sysplex failover configurations that provide a highly available environment for an SAP database on DB2 for z/OS. These are (in increasing levels of 'availability'):

- Option 0: Single DB2 member with passive (inactive) standby member
- Option 1: Two active DB2 members in active-standby mode
- Option 2: Two active DB2 members, each with a passive standby member in same LPAR
- Option 3: Two active DB2 members, each with a passive standby member in a standby LPAR

For further information about each of these options, see Chapter 4, "DB2 data sharing on System z Parallel Sysplex," on page 39.

2. Configure the network

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

The duplicate paths should be exploited in failure situations in a transparent way from an application point of view. This can be done hardware-based, if 'paired' switch equipment is used, or software-based by using features like z/OS VIPA (Virtual IP Address) and dynamic routing (OSPF). Such a software-based setup is mandatory, if you want to exploit *HiperSockets*, because HiperSockets cannot be used for communication between different physical machines.

Software-based network high availability builds on implementing:

- OSPF protocol as a recovery mechanism
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.
- 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. 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 (physical) Network Interfaces fail, the VIPA can still be reached via one of the other Network Interfaces. Therefore, a Network Interface is no longer a SPOF (single point of failure).

Chapter 7, "Network considerations for high availability," on page 71 covers 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 79

3. Exploit SAP features supporting high availability

DB2 connection failover provides the capability to redirect application servers to a standby database server in case the primary database server becomes inaccessible. By exploiting the DB2 data sharing function in a sysplex, you can provide redundancy at the database service layer. See Chapter 4, "DB2 data sharing on System z Parallel Sysplex," on page 39 for more information.

Install and run the Standalone enqueue and enqueue replication server for ABAP and Java. This means, you should use SAP Central Services for ABAP and Java (ASCS and SCS). For details, see Chapter 10, "Customizing SAP for high availability," on page 135.

Install each SAP Central Service instance with its own virtual hostname. This allows each instance to be moved separately within a cluster, if necessary. Standalone enqueue servers avoid enqueue data loss and Database rollbacks in case of a failure of the central enqueue server. It allows fast failover of those services to a standby machine where the replication server runs. It also reduces planned outages thru rolling kernel update (see SAP note 953653 - "Rolling Kernel Switch").

Another SAP function we highly recommend exploiting is the use of *SAP logon groups*. This will enable the distribution of SAP users across available application servers based on requirements for workgroup service and utilization. In case an AS becomes unavailable, a SAP user will drop out of her/his GUI session and has just to re-logon in order to be transferred to an AS still running.

It is also possible to check if an application server is running correctly. For ABAP stack, we use the `rfcping` utility, which does a dummy transaction for a specified Appl. server indicating, if a SAP user can do real work on that AS. For Java stack we use a small java executable (`GetWebPage.class`) which checks the availability of a web site, indicating that the Java stack is running.

4. Automate system operations for SAP

IBM Tivoli System Automation for z/OS (SA z/OS) as well as for Multiplatforms (SA MP) is a product family that provides high availability (HA) 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 in order to avoid for example operator errors resulting from this complexity, we highly recommend automating system operations for SAP with the System Automation product family. Automation also allows the system to react on detection of failed resources, 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 desired high availability.

A list of such resources and components appears below. Note that it is *not* a complete list, but instead gives an idea of what has to be considered. However, you are not required to define these resources and components “from scratch”. The SA products come with so-called ‘best practices’ policy samples, which you have to adapt to your real environment.

Operating system components

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

Database

- DB2

Parallel sysplex

- Integration with GDPS

SAP components

- ABAP Central services:
 - Enqueue server
 - Message server
 - Gateway (optional)
 - Syslog Collector (optional)

Planning

- Java Central services:
 - Enqueue server
 - Message server
- ABAP Enqueue Replication server
- Java Enqueue Replication server
- Application server
 - Remote ABAP or Java instances
- Other components:
 - SAPOSCOL
 - SAPCCMSR
 - SAP Router

Running SAP under the control of System Automation currently requires some customizing of SAP during and after installation. You must:

1. Install SAP in a high availability setup, for example by installing all SAP Central Services (SCS) with their *own virtual hostname*. Even for a double-stack with ABAP and Java SCS, each instance must be installed with its own virtual hostname.
2. Add SAP components that are required for achieving high availability of your SAP system but which are currently (as of SAP Netweaver 7.0) *not supported by the SAP installer*, such as the enqueue replication server.

Additionally, be aware that setting up the resources listed above to create a specific SAP environment for a customer is a *manual task*. For System Automation, 'best practice' policies are available for modeling a SAP system. You can use these 'best practise' policies as a starting point, but since each environment is unique these 'best practise' policies require modification and subsequent testing.

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

1. Install and setup SA z/OS and SA MP, and define all z/OS 'base' subsystems to SA z/OS. See Chapter 9, "Preparing a high availability SAP solution," on page 119.
2. Make the NFS server highly available via SA:
 - For details of how to setup a HA NFS server *under z/OS*, see Chapter 9, "Preparing a high availability SAP solution," on page 119.
 - For details of how to setup a HA NFS server *under AIX/Linux*, see Chapter 12, "Customizing the Tivoli System Automation for Multiplatforms (Base)," on page 195.
3. Install and customize SAP with HA setup using virtual host names for each SAP Central Service instance. See Chapter 10, "Customizing SAP for high availability," on page 135.
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 SA MP using the best practices SAP policy according to your *selected automation option* (described in "Deciding which SA automation option you wish to implement" on page 13). Finally, set up SA end-to-end. See Part 5, "System Automation," on page 169.
6. Verify that the installation is performing as expected. See Part 6, "Verification," on page 247.

Making NFS highly available

NFS is an integral part of a distributed SAP system and as such *must also be highly available*. We recommend running NFS Server under z/OS and make it highly available with SA z/OS.

Alternatively, it is also possible to run the NFS server *outside* z/OS. This might be done when no SAP components apart from the database run under z/OS. You have *two* options for setting up a highly available NFS server under SA MP

- Option 1 is to run the NFS server within the *same SA MP domain* as the SAP system. Use this option if you only want to use the NFS server for that single SAP system.
- Option 2 is to define for the NFS server its *own SA MP cluster*. Use this option if the NFS server will serve *several SAP Systems*. In this case, you should use the 'enhanced' NFS server policy. It is recommended that you setup the SA MP end-to-end component so that you can define a StartAfter relationship between the NFS server cluster and the SAP clusters.

Planning

Part 2. Database

Chapter 3. SAP availability benefits provided by System z 27

Features of the System z hardware architecture	27
Features of z/OS	27
Availability features and benefits with System z Parallel Sysplex	28
List of System z Parallel Sysplex availability features.	28
Features of DB2 for z/OS.	29
List of DB2 for z/OS availability features	29
List of DB2 for z/OS availability features with data sharing	35
DB2 data sharing	35
Non-disruptive software changes	35
DB2 for z/OS improvements	36
Group Buffer Pool (GBP) duplexing	36
Duplexing of SCA and lock structures	36
"Light" DB2 restart	36
SAP benefits and availability scenarios	37

Chapter 4. DB2 data sharing on System z Parallel Sysplex 39

Why Parallel Sysplex and data sharing for SAP?	39
Parallel Sysplex architecture	39
DB2 data sharing architecture	40
DB2 connection failover	41
Data sharing optimization for different SAP business applications	43

Chapter 5. Database architecture options 45

DB2 data sharing design options for SAP	45
Option 0: Single DB2 member with passive (inactive) standby member	46
Option 1: Two active DB2 members in active-standby mode	47
Option 2: Two active DB2 members, each with a passive standby member in the same LPAR	50
Option 3: Two active DB2 members, each with a passive standby member in an independent LPAR	51
How many data sharing groups?	51
How many sysplexes?	52
How many data sharing members?	52
Failover design	54

Chapter 6. Backup and recovery architecture in data sharing. 57

Data sharing considerations for disaster recovery.	57
Configuring the recovery site	57
Remote site recovery using archive logs	58
Using a tracker site for disaster recovery	60
Tracker site recovery	60
GDPS infrastructure for disaster recovery	60
Homogeneous system copy in data sharing.	64

Planning for homogeneous system copy in data sharing	64
Review of HSC in non-data-sharing	65
Requirements for data sharing	66
Designing homogeneous system copy in data sharing	66
Data sharing to data sharing.	67
Online copy design considerations.	67
Offline copy design considerations	68
Data sharing to non-data-sharing	68

Chapter 3. SAP availability benefits provided by System z

The IBM System z platform incorporates a variety of advanced autonomic computing capabilities. As discussed in Chapter 1, “Introducing high availability and automation for SAP,” on page 3, self-managing systems are:

- self-configuring
- self-healing
- self-optimizing
- self-protecting

While reading through the lists of high availability features below, you can check which characteristic applies in each case.

SAP on System z (single system or Parallel Sysplex) inherits all the intrinsic high availability features of the System z platform. These include hardware features as well as features of the software components involved. They provide a hardware and software infrastructure with the 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, it should be written not only to avoid failures but also to identify and recover those that occur. Automation also eliminates failures by ensuring that procedures are followed accurately and quickly every time.

The availability features of the System z platform are derived from these concepts. System z was designed with the reliability, availability and serviceability (RAS) philosophy. Its availability features result from 40 years of evolution and are incorporated in the System z hardware, the z/OS operating system, and DB2 for z/OS.

Features of the System z hardware architecture

For a complete and detailed explanation of the high availability features that are built in to System z servers, refer to the respective hardware reference guides.

For details of the current (as of the date this book was made available) System z hardware and features, refer to the following books:

- *IBM System z10 Enterprise Class (z10 EC) Reference Guide*, which you can find on: <http://www.ibm.com/systems/z/hardware/z10ec/index.html>
- *IBM System z10 System Overview*, SA22-1084
- *IBM System z10 Technical Introduction*, SG24-7515
- *IBM System z10 Technical Guide*, SG24-7516
- *IBM System z10 Capacity on Demand*, SG24-7504

Features of z/OS

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

SAP availability benefits provided by System z

of software reliability and availability features, far beyond that 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 continuing.
- 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 *IBM z/OS Reference Guide*. You can obtain a copy of this document at:

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

Availability features and benefits with System z Parallel Sysplex

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

For a complete list of advantages and benefits see:

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

List of System z 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**

Refer to “List of DB2 for z/OS availability features with data sharing” on page 35

- **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 have the ability to support non-disruptive software changes. This means that 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. This demands that both the old and new versions co-exist and work together within the Parallel Sysplex. For more information on this release tolerance, see “Updating DB2 or z/OS” on page 245.

For details on DB2 for z/OS, see “Features of DB2 for z/OS” and in particular “List of DB2 for z/OS availability features with data sharing” on page 35.

- **Non-disruptive policy changes**

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

Features of DB2 for z/OS

DB2 for z/OS was designed so that you should not have to take DB2 down in order to perform traditional database activities. Every new version of DB2 delivers new functions that are designed to ensure high availability. In this section we discuss the main features that are built into DB2 for z/OS to improve high availability and continuous operation of the database.

List of DB2 for z/OS availability features

In this section we only discuss DB2 for z/OS availability features that apply to standalone DB2s. For DB2 data sharing and related features that apply to Parallel Sysplex configurations, and which are probably the most important DB2 availability features, see “List of DB2 for z/OS availability features with data sharing” on page 35.

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 2. DB2 for z/OS availability features matrix

Availability feature	Reduces outage frequency	Reduces outage duration	Reduces outage scope	Planned outage	Unplanned outage
System-level point-in-time backup and recovery	X	X		X	X
Suspend log write		X		X	

SAP availability benefits provided by System z

Table 2. DB2 for z/OS availability features matrix (continued)

Availability feature	Reduces outage frequency	Reduces outage duration	Reduces outage scope	Planned outage	Unplanned outage
Variable control interval (CI) size			X	X	
Online backup with SHRLEVEL CHANGE option		X		X	
CONCURRENT Option in COPY		X		X	
CHANGELIMIT option in COPY	X	X		X	
COPYDDN and RECOVERYDDN	X			X	
Backing up indexes		X		X	X
Fast log apply		X			X
Online REBUILD INDEX (as of DB2 V9.1)	X			X	X
Automated point-in-time recovery by RESTORE SYSTEM		X			X
Embedded and online FlashCopy by BACKUP SYSTEM	X	X		X	
Online system parameters	X			X	
Alter checkpoint frequency		X			X
Parallel COPY and Parallel RECOVER		X		X	X
Automatic recovery at restart			X		X
Online reorg		X		X	
COPYDDN option in LOAD/REORG	X	X		X	X
Inline statistics		X		X	
Automatic space management	X			X	X
Automated LPL recovery		X	X		X
Virtual storage monitoring	X				X
Online schema evolution	X			X	
Partition independence		X	X	X	
Fast reorganization of partitioned tablespaces		X		X	

Table 2. DB2 for z/OS availability features matrix (continued)

Availability feature	Reduces outage frequency	Reduces outage duration	Reduces outage scope	Planned outage	Unplanned outage
Data-partitioned secondary indexes (DPSI)	X			X	
Partition Rebalancing	X			X	
X = applies					

The main DB2 for z/OS availability features are in detail:

- **System-level point-in-time backup and recovery**

The system level point in time recovery enhancement that was introduced in DB2 V8 provides the capability to recover the DB2 system to any point in time in the shortest amount of time. The DB2 utilities BACKUP SYSTEM and RESTORE SYSTEM have been introduced in DB2 V8 for this purpose. The incremental FlashCopy capability, which is supported starting with z/OS 1.8, can help you reduce the elapsed time of taking a physical backup and minimizes the impact on the I/O performance of concurrent applications. DB2 9 helps you automating the management of backups as BACKUP SYSTEM can directly dump backups to tape. Moreover, the BACKUP SYSTEM utility allows taking system-wide backups with unrestricted read and write activity of the SAP workload.

- **Variable control interval (CI) size**

DB2 V8 introduced support for CI sizes of 8, 16, and 32 KB. CI sizes that match DB2 page sizes avoid intra-page inconsistencies. Therefore, this feature allows taking volume-based backups without suspending write activity on pages with a size of 32 KB.

- **Backing up indexes**

In earlier DB2 versions, you could not make image copies of indexes. Therefore, you could recover indexes only by rebuilding the indexes from existing data. This process could be lengthy, especially if index recovery had to wait until the data was recovered, making those indexes unavailable until the rebuild was complete. Today, you can take a full image copy or a concurrent copy of an index, just as you have always done for tablespaces. To recover those indexes, you use the RECOVER utility. This restores the image copy and applies log records (or they are recovered as part of RESTORE SYSTEM processing).

- **Fast log apply**

A faster log apply process improves restart and recovery times up to 5 times in order to reduce unplanned outages. The process sorts out log records so that changes that are to be applied to the same page or same set of pages are together. Then, using several log-apply tasks, DB2 can apply those changes in parallel. This feature requires fewer I/O operations for the log apply and can reduce CPU time.

- **Recovery of individual tablespaces from system-level backup**

An enhancement of the RECOVER utility in DB2 V9.1 is the ability to recover an individual tablespace from a system-level backup which was created by BACKUP SYSTEM. This allows you to rely on system-level backups for both system-level and object-level recoveries. Only in specific situations is it necessary to take image copies in addition to system-level backups. For example, an inline image copy is required during online REORG processing. In addition, DB2

SAP availability benefits provided by System z

V9.1's RECOVER utility recovers consistently if the options TOLOGPOINT or TORBA are specified. This means that uncommitted transactions at the target point of the recovery are automatically rolled back.

- **Fast log-only recovery due to more frequent HPGRBRBA updates**

Log-only recovery is used when a database object is recovered based on a volume level backup (as opposed to image copies). The speed of the recovery depends on the amount of log data that needs to be scanned and applied. If DB2 V8's RESTORE SYSTEM utility is not utilized, DB2 uses the so-called Recover Base RBA (HPGRBRBA) that is recorded in the object's header page to determine how far in the log it needs to go.

Prior to V7, the HPGRBRBA was updated (moved forward) at the object's physical or pseudo-close time. Some heavily accessed objects might not be closed for a very long time and consequently their HPGRBRBA can get very old. As a result, the log-only recovery is likely to take very long. Starting with DB2 V7, the HPGRBRBA is updated more frequently, which results in less log data scanned and faster log-only recoveries.

- **Online system parameters**

DB2 V7 introduced online modifications of a large number of system parameters. This is beneficial for SAP systems both to correct some settings that inadvertently do not match values highly recommended by SAP and to adjust some parameter values for which SAP gave initial recommendations and which need to be modified to better suit specific the customer's workload, such as buffer pool sizes.

The system parameter values can be changed by means of the DB2 command SET SYSPARM.

- **Alter checkpoint frequency**

The SET LOG LOGLOAD(n) command allows you to dynamically change the LOGLOAD system parameter. This parameter controls the frequency of checkpoints. The more frequent checkpoints, the faster the DB2 restart after abnormal termination. On the other hand, too frequent checkpoints negatively affect performance. The command allows you to adjust the frequency according to your site objectives and do it dynamically, without restarting the system. Another interesting aspect of this command is initiating checkpoint on demand by specifying SET LOG LOGLOAD(0). For example, if the CI size does not match the page size of objects, it is recommended to issue this command before suspending log writes in order to reduce the number of database writes during the log write suspension and consequently the risk of generating inconsistent 32K size pages.

- **Automatic recovery at restart**

When a subsystem failure occurs, a restart of DB2 automatically restores data to a consistent state by backing out uncommitted changes and completing the processing of the committed changes.

- **Online reorg**

The REORG utility allows the reorganization of a tablespace or index during online operation. The keyword SHRLEVEL allows you to choose standard, read-only online, or read-write online reorganization. With the SHRLEVEL CHANGE option, you have both read and write through almost the entire reorganization process.

The process involves the following activities:

1. The utility unloads the data from the original tablespace, sorts the data by clustering key, and reloads the data into a shadow copy. Concurrently, SAP has read and write access to the original tablespace, and changes are recorded in the log.
2. The utility reads the log records and applies them to the shadow copy iteratively. During the last iteration, SAP has only read access.
3. The utility switches the application access to the shadow copy. Starting with DB2 V7, the renaming of data sets can be avoided, which improves performance considerably.
4. DB2 V9.1 can considerably speed up the reorganization of a partitioned tablespace by essentially processing the different partitions in parallel.
5. SAP reads and writes to the new data sets.

DB2 V7 introduced the REORG options `RETRY`, `DRAIN_WAIT`, and `RETRY_DELAY`, which control the strategy of the REORG utility to drain a tablespace. They allow execution of the REORG utility more concurrently with the application workload.

- **COPYDDN option in LOAD/REORG**

If you run REORG or LOAD REPLACE and use `LOG(NO)`, then an image copy is required for data integrity. By default, DB2 will place the tablespace in copy-pending status, and you have to perform an image copy before you can further change the tablespace. If you run REORG or LOAD REPLACE with the `COPYDDN` option, a full image copy is produced during the execution of the utility and DB2 does not place the tablespace in copy-pending status. This eliminates the period of time when the tablespace is in copy-pending status and a separate COPY step is not required. Therefore the data is available sooner.

- **Inline statistics**

Prior releases of DB2 require the user to update statistics by executing `RUNSTATS` after common utility operations on tablespaces such as `LOAD`, `REORG` and `REBUILD INDEX`. Today, you can include `RUNSTATS` within the execution of those utility operations. This avoids the need for separate `RUNSTATS` jobs and uses less processing resources by making fewer passes of the data. Furthermore, tablespaces will be made available sooner.

- **Automatic space management**

DB2 V8 introduced automatic space management that ensures that the data sets used for DB2 objects do not reach the maximum number of extents. For SAP, this feature is highly valuable for continuous availability because it provides adaptability when data growth patterns are not predictable or do not follow those expected. To let DB2 override secondary quantities that are too small, set the system parameter `MGEXTSZ` to `YES`.

- **Automated LPL recovery**

Prior to DB2 V8, pages that DB2 puts into the logical page list (LPL) needed to be recovered manually, which includes draining the entire page set or partition. The complete page set or partition is unavailable for the duration of the LPL recovery process. DB2 V8 recovers LPL pages without draining page sets or partitions. It only locks the LPL pages during the recovery process, leaving the remaining pages in the page set or partition accessible to applications. This significantly improves system performance and enhances data availability. Moreover, DB2 V8 attempts to automatically recover pages when they are added to the LPL.

- **Virtual storage monitoring**

DB2 V7 introduced two new instrumentation records, 217 and 225, that externalize a snapshot of the current virtual storage usage by DB2. This enables

SAP availability benefits provided by System z

monitors such as DB2 Performance Expert to show the virtual storage map and precisely report on how much storage is used by different DB2 resources, such as thread storage, local and global cache, buffer pools etc. Analyzing this data can help avoid inefficient memory over-allocation.

- **Online schema evolution**

DB2 V8 introduced major improvements in the area of online schema evolution. Online schema evolution allows for table, index, and tablespace attribute changes while maximizing application availability. For example, you can change column types and lengths, add columns to an index, add, rotate, or rebalance partitions, and specify which index you want to use as the clustering index. Even before DB2 V8, support was provided to enlarge columns of type VARCHAR in an online operation. DB2 V9.1 further enhance the online schema evolution capabilities and provides support (for example) for online RENAME INDEX, online RENAME COLUMN and online ALTER COLUMN SET DEFAULT.

- **Partition independence**

A key availability feature of DB2 for z/OS is the ability to partition a DB2 tablespace into as many as 256 partitions prior to DB2 V8 and 4096 partitions starting with DB2 V8. The maximum size of a partition is 64 GB. The partition size determines the maximum number of partitions that is possible.

- **Fast reorganization of partitioned tablespaces**

There are two features introduced in DB2 V7 that provide for faster reorganization of partitioned tablespaces: avoiding partitioning index key sort and using parallelism in the BUILD2 phase. When reorganizing a partitioned tablespace and when parallel index build is used, REORG will build the partitioning index during the RELOAD phase instead of piping it afterwards to a sort subtask. This is possible because for a partitioned tablespace the rows are reloaded in the PI order, so the PI keys are already sorted. Nevertheless the number of sort and build tasks will not be changed, but one of the sort tasks will not be called at all. Therefore, if you allocate the sort work data sets yourself, be sure that there are some for the task for processing the PI (although with minimal space allocation). The BUILD2 phase is downtime for online reorg. With parallelizing it (and assuming a single partition reorg) each original NPI is assigned a subtask to update its RIDs based on the shadow copy of the logical partition. When reorganizing a range of partitions, the same subtask will update the NPI for all the logical partitions.

- **Data-partitioned secondary indexes (DPSI)**

DB2 V8 introduced data-partitioned secondary indexes to improve data availability during partition level utility operations and facilitate partition level operations such as roll on/off a partition or rotate a partition. The improved availability is accomplished by allowing the secondary indexes on partitioned tables to be partitioned according to the partitioning of the underlying data. There is no BUILD2 phase in REORG SHRLEVEL CHANGE when all secondary indexes are so partitioned.

- **Partition rebalancing**

When data in a partitioned tablespace becomes heavily skewed, performance can be negatively affected because of contention for I/O and other resources. In this case you might want to shift data among partitions. DB2 enables you to rebalance those partitions more efficiently. Partitioning key ranges can be altered while applications that access data not affected by the rebalance continue to run. The actual redistribution is accomplished by running the REORG utility for affected partitions after adjusting the key ranges. The REORG both reorganizes and rebalances the range of partitions. As of DB2 V8, you can also specify that

the rows in the tablespace or the partition range being reorganized should be evenly distributed for each partition range when they are reloaded (option REBALANCE). Thus, you do not have to execute an ALTER INDEX statement before executing the REORG utility.

- **Partition-by-growth Universal Tablespace**

DB2 V9.1 introduces a new tablespace type, which is called partition-by-growth Universal tablespace. Such a tablespace always contains a single table and dynamically grows as needed. This means that a table can virtually grow without any limits. Therefore, no downtime is needed to modify a table that has grown considerably.

List of DB2 for z/OS availability features with data sharing

DB2 data sharing

Data sharing is a key element in the Parallel Sysplex continuous availability design. It allows the redundancy needed to overcome both the failure of a member which is processing database updates, and allows to schedule the stopping of a member for service or a similar action. In each case, the service provided by the DB2 data sharing group is unaffected. Failover mechanisms direct application servers to a surviving DB2 system in case their primary system becomes unavailable for any reason (including planned outages such as software maintenance).

DB2 data sharing is based on the “shared everything” approach: There can be multiple DB2 subsystems belonging to one DB2 data sharing group. Each DB2 member of the data sharing group is assigned both read and write access to all data in the database. Therefore, all data must reside on shared DASD. The members of a DB2 data sharing group use coupling facility services to communicate and move data between each other. The Coupling Facility (CF) is further used by the individual members of a data sharing group to exchange locking information, system and data object status information, and to cache shared data pages with the other members of the group.

DB2 uses special data sharing locking and caching mechanisms to ensure data consistency. When one or more members of a data sharing group have opened the same tablespace, index space, or partition, and at least one of them has been opened for writing, then the data is said to be of “inter-DB2 R/W interest” to the members. To control access to data that is of inter-DB2 interest, DB2 uses the locking capability provided by the CF. As already mentioned, DB2 also caches the data in a storage area in the CF called a group buffer pool structure. Group buffer pools are used for caching data of interest to more than one DB2. There is one group buffer pool for all local buffer pools of the same name. When a particular page of data is changed by one DB2 subsystem, DB2 caches that page in the group buffer pool. The CF invalidates any image of the page in the local buffer pools of all the members. Then, when a request for that same data is subsequently made by another DB2, it looks for the data in the group buffer pool. The access from each DB2 to the CF is handled in a matter of microseconds, so that overall linear scalability is reached.

Non-disruptive software changes

DB2 for z/OS has the ability to support non-disruptive software changes. This means that an individual data sharing member can be upgraded by removing that member from the sysplex and adding the upgraded member back when ready.

SAP availability benefits provided by System z

This demands that both the old and new versions co-exist and work together within the Parallel Sysplex. For more information on this release tolerance, refer to “Updating DB2 or z/OS” on page 245.

DB2 for z/OS improvements

The initial delivery of DB2 data sharing came with Version 4 of DB2 for OS/390. With each new DB2 release, high availability and performance characteristics of data sharing have been improved. The following data sharing features particularly benefit availability:

Group Buffer Pool (GBP) duplexing

The option to duplex group buffer pools enables you to have a hot standby copy of a GBP ready and waiting. Each GBP is allocated in a different CF. Changed pages to both a primary and a secondary group buffer pool are written at the same time, where overlapped writes to both GBPs provide for good performance. If the primary GBP fails, DB2 for z/OS can recover quickly by switching over to the secondary GBP. Also if the secondary GBP fails, DB2 just drops back to simplex mode. If both the primary and secondary GBPs are damaged, DB2 can still use automatic GBP recovery for a duplexed GBP. GBP duplexing allows for faster recovery in the unlikely event of a CF failure and also for partial or 100% loss of connectivity. It makes the data sharing subsystems more reliable and reduces the time to recover the GBP.

Duplexing of SCA and lock structures

DB2 V7 introduced support for system-managed duplexing of SCA (shared communication area) and lock structures. Duplexing SCA and lock structures enables you to use ICFs (internal coupling facilities) to achieve levels of availability comparable to external CFs, because single points of failure are eliminated. To achieve availability benefits, both of these structures need to be duplexed. Duplexing only one of them does not provide any benefit.

Be aware that duplexing the SCA and lock structures impacts performance, particularly if the CF links are slow and if an excessive number of locks are propagated to the CF at once. This can happen if intersystem read/write interest occurs for a page set that is updated by a long-running transaction.

If you decide to duplex the SCA and lock structures, consider making DB2 propagate child locks prematurely to the lock structure. This can be controlled by the DB2 system parameter SPRMFLKT, which has been introduced in APAR PK01667.

“Light” DB2 restart

If the primary DB2 system fails and the application load moves to another system, it is very important to resolve any outstanding locks (so called retained locks) still held by the failing system. Starting with DB2 V7, a data sharing member can be optionally started with a special purpose: to resolve the retained locks as soon as possible. Such a DB2 start is called ‘light start’ and is requested by means of the LIGHT option of the START DB2 command. In this case the DB2 system starts with a minimal storage footprint and minimum functionality necessary to resolve the retained locks, after which it terminates. Therefore this system cannot be used for any other purpose.

The System Automation “Best Practise” policy for DB2 delivered with APAR OA26776 includes this light DB2 restart functionality. For details, see Chapter 10, “Customizing SAP for high availability,” on page 135.

SAP benefits and availability scenarios

System z Parallel Sysplex and DB2 data sharing will avoid SAP system downtime in the following hardware and software failure scenarios:

- Outage of a CEC in the Parallel Sysplex
- Coupling Facility outage
- Coupling Facility Link failure
- z/OS outage on a sysplex member
- DB2 subsystem outage on a sysplex member
- System z hardware upgrade in the Parallel Sysplex
- Installation of an additional Coupling Facility
- z/OS upgrade on a sysplex member
- DB2 upgrade on a sysplex member

For a description of tests for some of the failure-scenarios listed above, see:

- Chapter 15, "Verifying your implementation on z/OS," on page 249.
- The IBM Redbook *SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations*, SG24-5690,

SAP availability benefits provided by System z

Chapter 4. DB2 data sharing on System z Parallel Sysplex

This chapter discusses:

- The benefits that result from implementing an SAP-DB2 data sharing solution.
- The building blocks for a continuously available and scalable system.

We describe the following topics:

- Why Parallel Sysplex and data sharing for SAP
- Parallel Sysplex[®] architecture
- DB2 data sharing architecture
- SAP sysplex failover architecture

Why Parallel Sysplex and data sharing for SAP?

There are several motivations for pursuing an SAP implementation based on System z Parallel Sysplex and DB2 data sharing. Many customers are deploying SAP applications in support of business-critical operations. Typical business drivers include a desire to run a single global SAP instance, customer and supplier access to Web-based SAP applications around the clock, and support of 24x365 manufacturing and distribution operations or 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, “Introducing high availability and automation for SAP,” on page 3.
- Central processor scalability through horizontal growth

The infrastructure for the high availability solution 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. It was also constrained by the amount of data and control information that could be held in the primary DB2 address space (the DBM1 address space). The SAP DB2 Parallel Sysplex architecture enables us to overcome these constraints and cluster multiple CECs in a single DB2 data sharing group. This enables horizontal growth of both processor power (MIPS) and virtual storage. Data sharing also gives us another means in workload management as we 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 operating against a single copy of the data. Figure 7 on page 40 introduces a high-level picture of the elements of a System z Parallel Sysplex environment.

DB2 data sharing

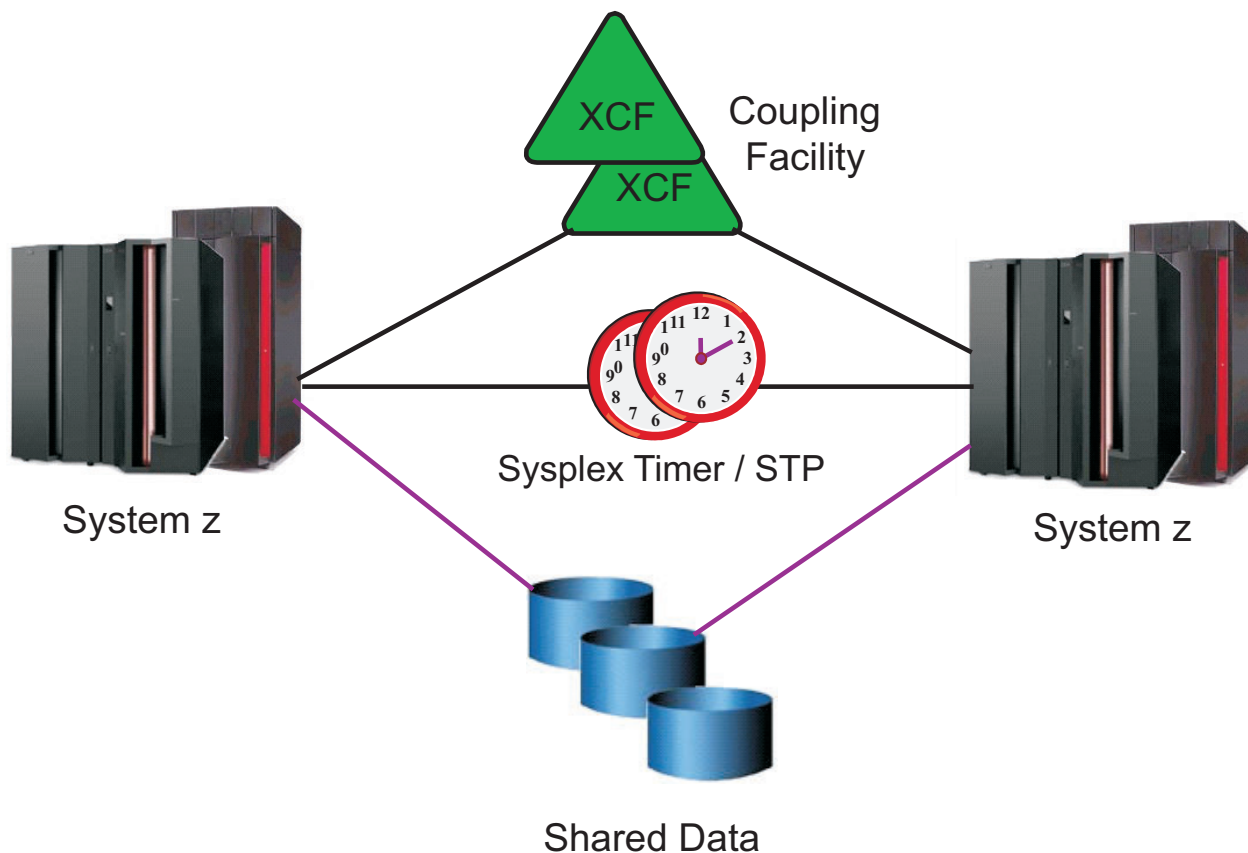


Figure 7. System z Parallel Sysplex architecture elements

- In this figure, we see that a Parallel Sysplex system is typically made up of:
- 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 called Sysplex Timers or STP,
 - sysplex-wide shared data, and
 - multiple high-speed, duplexed links connecting the components.

This implementation employs hardware, software, and microcode.

DB2 data sharing architecture

Figure 8 on page 41 completes the picture by laying multiple DB2 data sharing members on top of the Parallel Sysplex infrastructure.

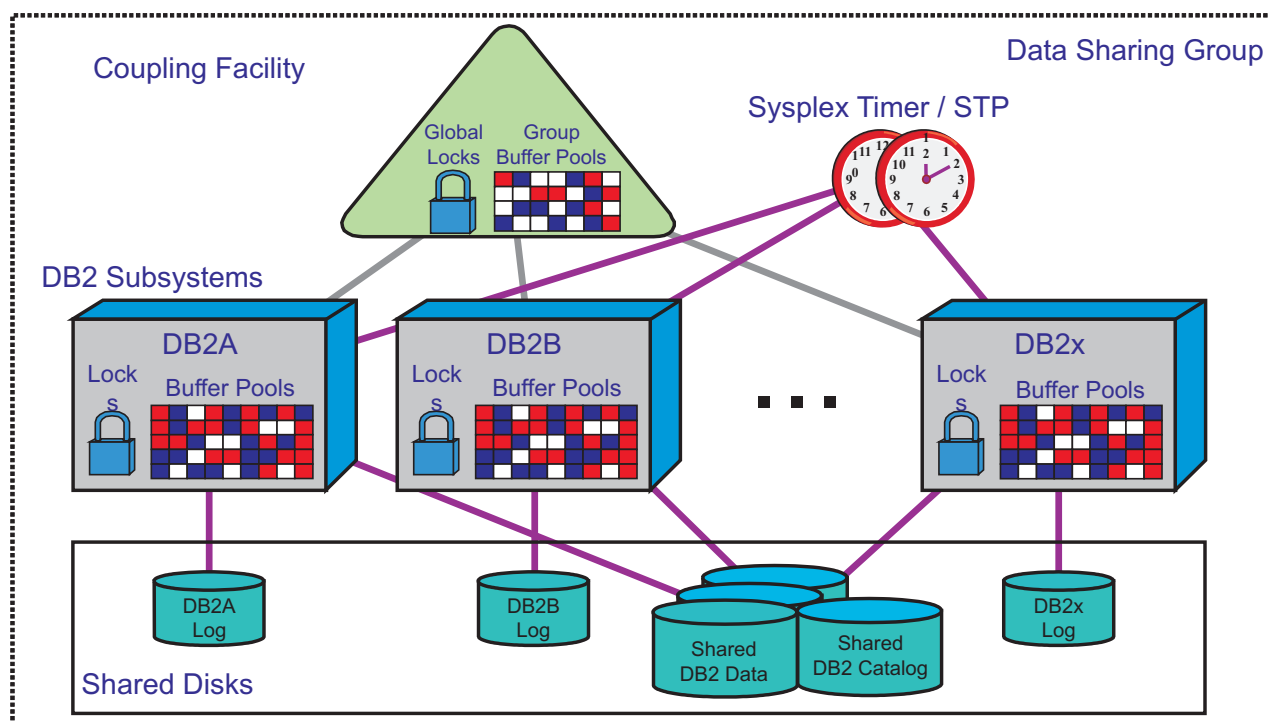


Figure 8. DB2 data sharing in a Parallel Sysplex

In Figure 8 we 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 managed by a companion IRLM. The DB2 data sharing group shares the DB2 databases, 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 DB2 document *Data Sharing: Planning and Administration*.

DB2 connection failover

This section provides an overview of *DB2 connection failover*, which completes the description of the DB2 data sharing infrastructure.

If a planned or unplanned outage of a DB2 data sharing member occurs, we use the term *DB2 connection failover* to refer to the capability of an SAP system to switch its DB2 database connection to another DB2 data sharing member. This avoids a (partial or even complete) SAP system outage.

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

ABAP instances

- DB2 connection failover is implemented here as a *SAP functionality* (in earlier editions of this manual, this functionality was referred to as *SAP sysplex failover*).
- SAP employs a configuration file called **connect.ini**, that you need to customize. For details of how to perform this customization, refer to the *SAP Database Administration Guide*.

DB2 data sharing

- SAP's failover mechanism provides support for the reconnect of an ABAP instance to a standby DB2 data sharing member in case of an *unplanned outage* of the primary DB2 data-sharing member.
- For a seamless failover, which does not roll back any transactions when a *planned outage* occurs:
 - Prior to the planned downtime of a DB2 member, SAP transactions ST04 or DBACOCKPIT can be used to direct the DB2 connections of an ABAP instance to a *different* DB2 member.
 - Alternatively, you can automate the failover to a different DB2 member via ABAP function `STU3_ADMIN_SWITCH_DB_CON`. For details, refer to SAP Note 915482.

JAVA instances

- DB2 connection failover is based on the native database failover feature of the *IBM Data Server Driver for JDBC*, which is packaged as part of DB2 Connect.
- A basic variant of the connection failover (delivered by DB2 Connect V9.1 FP3) can only deal with unplanned outages.
- The extended (and recommended) variant in addition is able to provide seamless failover in the planned outage case. Support for this variant was initially delivered by DB2 Connect V9.5 FP1.
- For both variants, refer to SAP note 1085521 for implementation details and required software levels.

Figure 9 introduces the basic building blocks needed for DB2 connection failover.

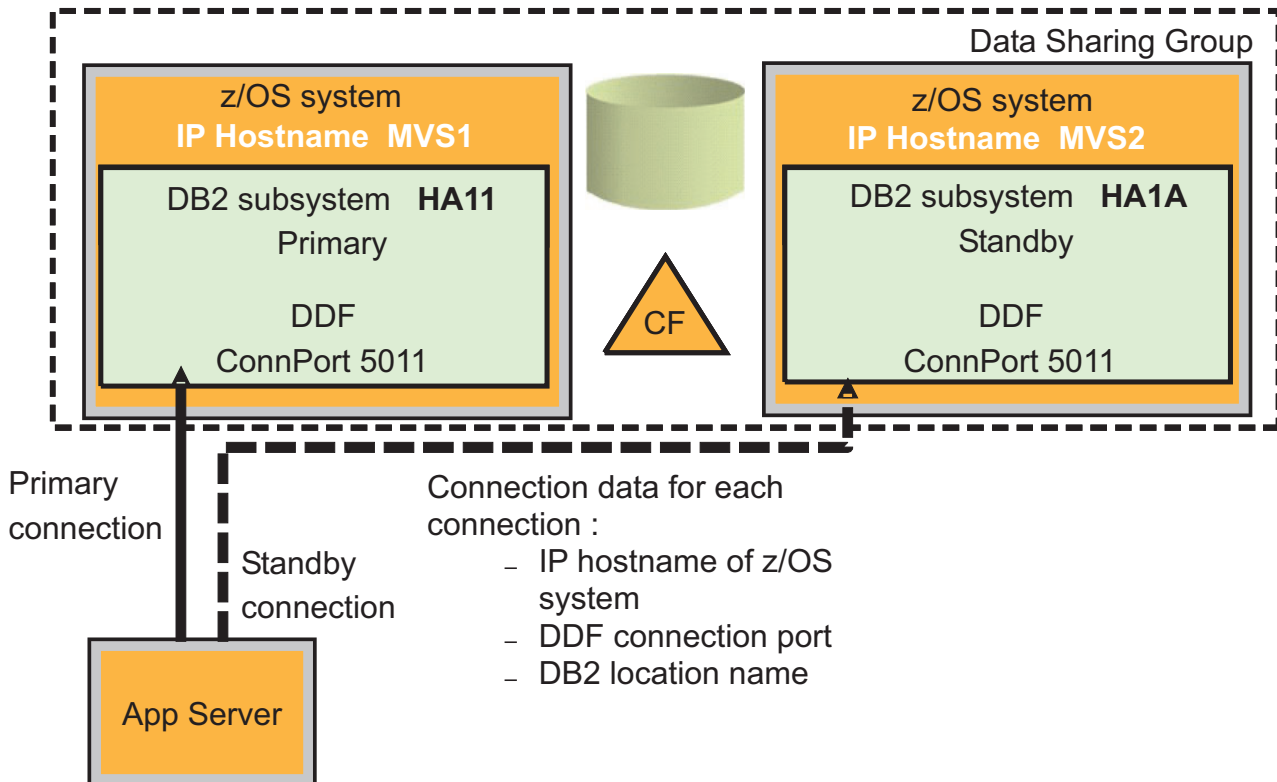


Figure 9. SAP sysplex failover configuration: Option 0 example

There are four major data sharing failover configurations providing a highly available environment for an SAP database on DB2 for z/OS:

- Option 0: Single DB2 member with passive (inactive) standby member
- Option 1: Two active DB2 members in active-standby mode
- Option 2: Two active DB2 members, each with a passive standby member in same LPAR
- Option 3: Two active DB2 members, each with a passive standby member in a standby LPAR

We discuss these options in Chapter 5, “Database architecture options,” on page 45.

The sample SAP system name (SAPSID) HA1 is used in Figure 9 on page 42. In the example, we also use HA1 as the DB2 group name.

We also introduce the notion of *primary* DB2 members, which normally have application servers attached doing productive work, and *standby* DB2 members, which can be 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 will be the DB2 group name plus a letter (**HA1A**).

This figure 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).

For all ABAP application servers a general file, the *connect.ini* file exists which contains this DB2 connection information:

- the DB2 location name,
- the z/OS LPAR virtual IP host name,
- the connection port, and
- the DB2 member name of the primary DB2 member, as well as for standby DB2s of each application server.

In the event of a planned or unplanned incident, the SAP Database Services Layer (DBSL) recognizes the need to fail over, looks for standby information in the control file, and connects the application server to the standby DB2 member.

For details on *connect.ini*, see the *SAP Database Administration Guide*.

Data sharing optimization for different SAP business applications

In general, the DB2 data sharing performance and scalability is excellent with SAP business applications. Nevertheless, there are still some data sharing tuning opportunities (such as partitioning the SAP asynchronous update protocol tables) which are described in the *SAP NetWeaver DBA Guide for DB2 for z/OS*.

In addition, there are specific tuning steps to further optimize data sharing performance that can be accomplished for specific applications:

- The data sharing optimizations for SAP banking applications are summarized in SAP note 496904.
- The data sharing optimizations for SAP Business Intelligence are summarized in SAP note 1239127.

DB2 data sharing

- The data sharing optimizations for SAP PI are summarized in SAP note 1260453.

Chapter 5. Database architecture options

This chapter explains SAP design considerations in a Parallel Sysplex data sharing environment. We describe:

- DB2 data sharing design options for SAP
- Failover design
- High availability considerations with DB2 Connect and DDF

DB2 data sharing design options for SAP

There are four basic data sharing options for providing a highly available environment for an SAP database using DB2 on z/OS.

For review, the options are:

- Option 0: Single DB2 member with passive (inactive) standby member
- Option 1: Two active DB2 members without passive standby members
- Option 2: Two active DB2 members, each with a passive standby member in the same LPAR
- Option 3: Two active DB2 members, each with a passive standby member in an independent LPAR

We do not go into detail about each option's configuration. This is already described in the IBM Redbook *SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations*, SG24-5690. We only discuss any applicable performance and monitoring aspects of the choice of options.

SAP Note 915482 describes ABAP function modules that you can use in automation procedures to schedule the redirection of application servers to another DB2 member at a specific time.

Option 0: Single DB2 member with passive (inactive) standby member

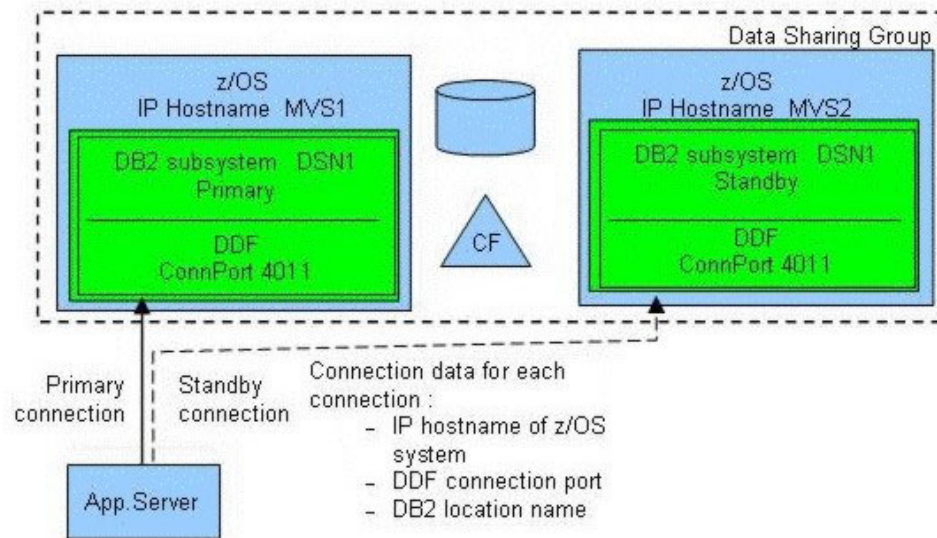


Figure 10. Option 0: Single DB2 member with passive (inactive) standby member

This option is chosen most often when high availability is the main concern and the current hardware is sufficient to handle all database SAP requirements identified from an SAP sizing or Insight for SAP report. In other words, your SAP database workload can be handled by one DB2 data sharing member in one CEC.

However, you should also be aware that the fastest failover performance when a DB2 member goes down is achieved with *two or more active members*. This is because the P-locks held by the different members are usually more granular with two or more active members.

Under normal conditions (with every component working properly), the passive DB2 member should not use any system resources except as needed to start each component. Even though the idea of high availability is to eliminate human intervention, system programmers (both z/OS and SAP) should check the status of their systems periodically.

To check the current state, you can use the data sharing view that SAP transactions ST04 or DBACockpit provide. From a z/OS perspective, one of the easiest ways to check the current state interactively is to use the DA screen in SDSF. On this screen, one can view the CPU activity of individual address spaces. All that is required is to look for CPU activity on the passive standby DB2 address spaces.

From an SAP system programming (SAP Basis) perspective, it is not always possible to get access to a user ID with a TSO segment on the z/OS system to perform such monitoring. SAP transaction 'DB2' enables administrators to initiate a failover of application servers from one DB2 member to another. Implicit in this functionality is the ability to determine which DB2 member is currently being used by an application server. From the main screen of transaction DB2, click *Data Sharing Topology* to see the current state. Click *DB Connection List* to move application servers to the other DB2 member.

Also, the SAP system log (transaction SM21) indicates when a DB2 connection failover occurred

Option 1: Two active DB2 members in active-standby mode

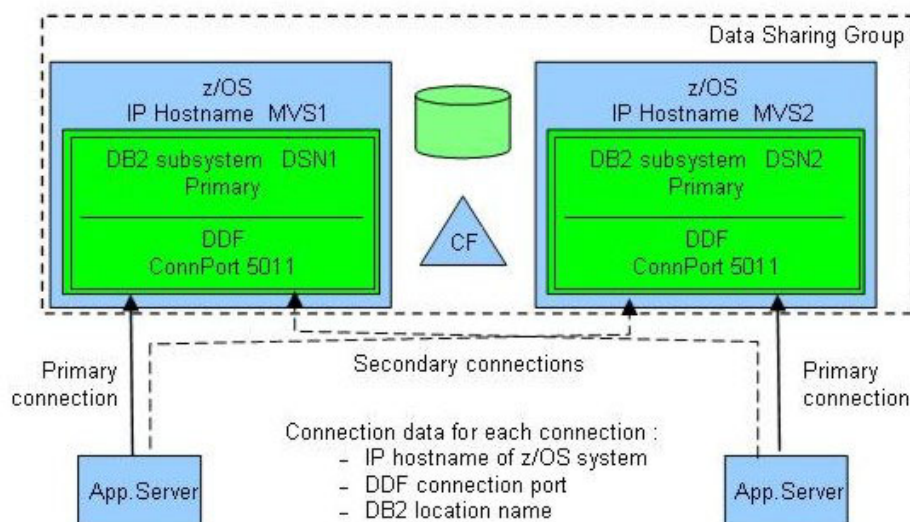


Figure 11. Option 1: Two active DB2 members in active-standby mode

This option is considered most often when one DB2 member running in one CEC or LPAR cannot handle the SAP database workload from a CPU capacity or virtual storage standpoint. Before DB2 V8, the DBM1 address space of DB2 was limited to 2 GB virtual storage. To alleviate virtual storage shortages in DBM1, data spaces can be implemented to accommodate buffer pools and the dynamic statement cache part of the EDM pool. DB2 V8 improves the situation by moving most of DB2's storage structures above the 2 GB bar.

When thinking about configuring DB2 to support more workload, this is most often the first thought that comes to mind. In this configuration, *DB2 connection failover* is set up so that application servers will move (connect) from the failing active DB2 member to the other active DB2 member. If the respective machines supporting these DB2 members are sized just to fit the normal workload, then one should expect degraded performance when all of the SAP database workload is concentrated on the surviving DB. This degraded performance could come about due to lack of CPU capacity or lack of memory. Consider using the System z capacity on demand options such as CBU (Capacity Backup Upgrade).

The monitoring possibilities of this configuration are essentially the same as with option 0. Monitoring from z/OS can be accomplished with SDSF or the z/OS console. If you had more than one user ID with TSO segment, then you could be logged on to both LPARs and view the DA screen simultaneously.

Actually, it is possible to display the information about all address spaces in the sysplex from one SDSF screen. For ease of use, it would be beneficial to name the address spaces in such a manner as to make them easily discernable from each other.

DB architecture

Monitoring within SAP is similar to what is described for option 0 above. It is possible to execute SAP transaction 'DB2' from any application server and check the status of any DB2 member.

In the case of failure for option 0, only one LPAR is in use, so there is no increase in database workload. Note that there is the possibility for one subset of application server work processes to be connected to the primary DB2 member and another subset of work processes in the same application server to be connected to the standby DB2 member. There is no real concern, because the workload is still in one LPAR. It only matters for monitoring purposes.

If you decide to use option 1, we remind you to give careful consideration to sizing the hardware properly and configuring Workload Manager (WLM). If you require the same level of performance no matter what state the system is in, then each system should have enough CPU and memory capacity reserved to handle the maximum additional workload on each system. Fortunately, one of the great strengths of z/OS on System z is the capability to support multiple workloads simultaneously. This is where WLM is important, because it enables you to assign importance to each workload. So in the event of a failover of workload to one surviving DB2 member, WLM can be configured to ensure that the SAP workload receives priority over the other workload, even if it is non-production SAP workload.

If it is non-production SAP workload, then extra definitions in WLM are required for WLM to distinguish between the SAP systems. Those familiar with the SAP on DB2 for z/OS series of planning guides should note that the sample WLM definitions assume that you are running one and only one SAP system per LPAR. All of the service classes begin with the prefix SAP. If you want to mix production and non-production workload or run multiple production workloads in the same LPAR, the sample definitions must be extended to control these workloads. One way is to create service classes for each SAP system. For example, you could create PR1HIGH, PR1MED, and PR1LOW for SAP production system 'PR1' and DR1HIGH, DR1MED, and DR1LOW for the SAP development system 'DR1'. A more flexible naming strategy would be to put the SAP system name in the service classes.

The following figure shows how one large company with multiple SAP workloads has selected the data sharing architecture options best suited to each workload.

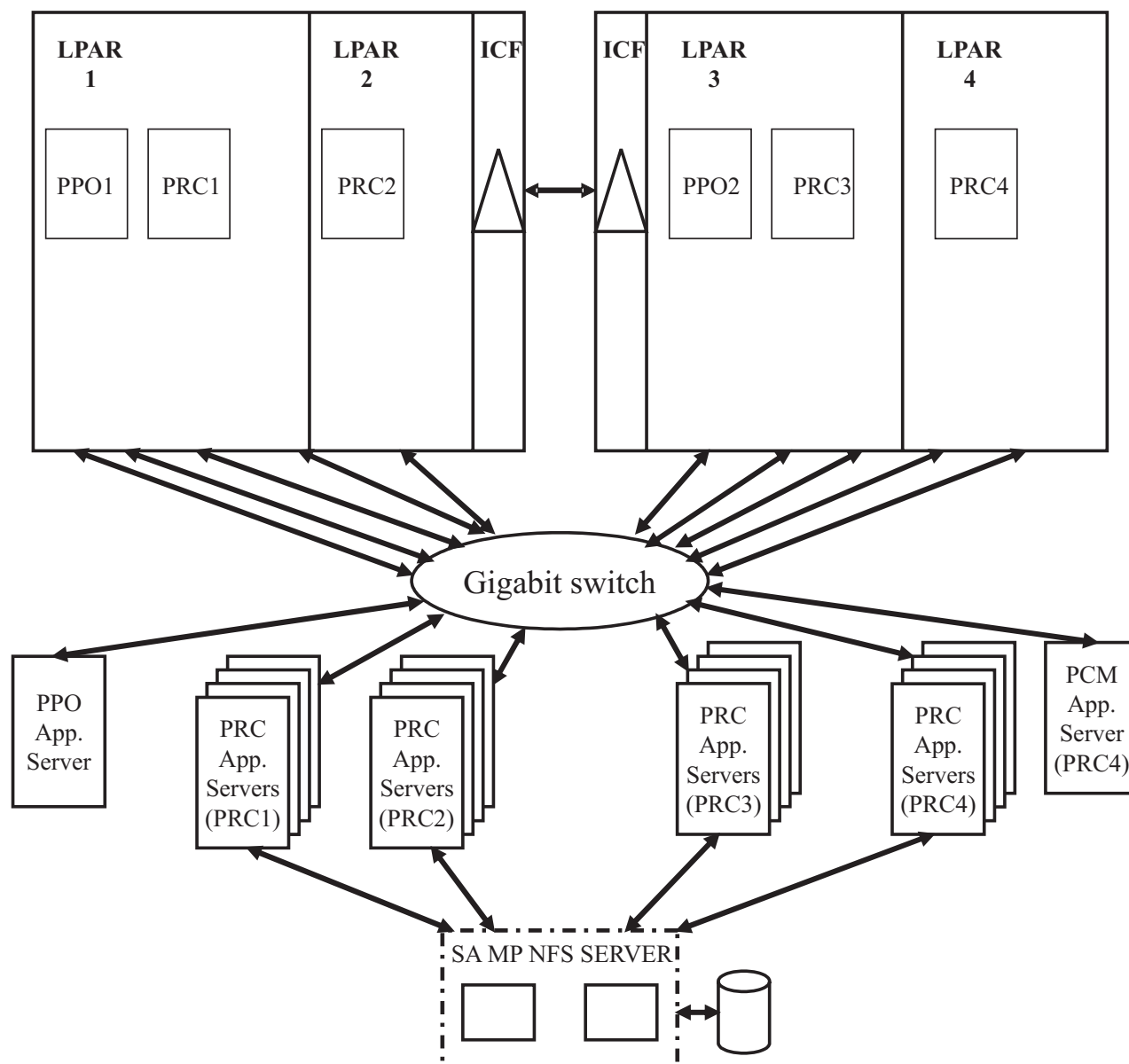


Figure 12. Large company using architecture options 0 and 1

This example shows a variation on data sharing options 0 and 1. The production sysplex has four LPARs spread across two mainframe servers. Each server has an internal coupling facility defined.

Two production DB2s are running, supporting:

- SAP ERP 6.0 (SAPSID=PRC)
- SCM Release 5.0 (SAPSID=PPO)
- CRM 2007 (SAPSID=PCM)

The SAP ERP and the CRM system are running in the same DB2: a configuration that SAP designates as *Multiple Components in One Database (MCOB)*.

ERP runs four-way active data sharing. Servers attached to PRC1 fail over to PRC4 and vice versa. PRC2 servers fail over to PRC3 and vice versa.

DB architecture

Each DB2 data sharing member has the capacity to handle the extra load for failover.

SCM runs two-way active/passive data sharing.

CRM runs data sharing option 0, because the CRM server is attached to a single member within the data sharing group, with failover to a second member.

With this setup, we can apply maintenance to z/OS and DB2 by controlled failover of the SAP systems during productive operation.

Changes for increased high availability are:

- A second gigabit switch (although there is built-in redundancy for all components within the switch).
- Stand-alone enqueue server to replace the SAP CI as the single point of failure.
- Move the NFS server to z/OS and make it highly available via SA z/OS. Alternatively, Tivoli System Automation for Multiplatforms can be used to make the NFS server highly available on other platforms.
- CRM in its own data sharing member. If we find that the CRM load affects ERP, or if we need different ZPARM settings for CRM, we can give CRM its own data sharing member within the PRC data sharing group.

Option 2: Two active DB2 members, each with a passive standby member in the same LPAR

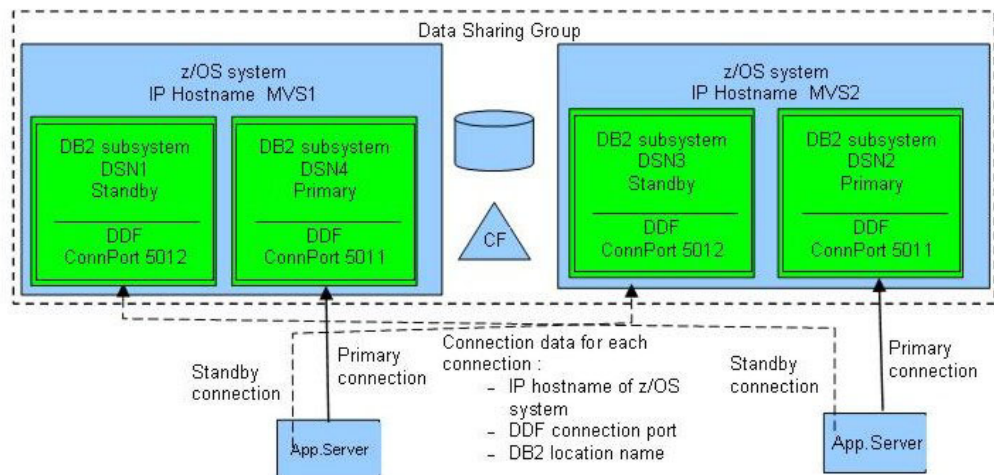


Figure 13. Option 2: Two active DB2 members, each with a passive standby member in the same LPAR

Option 2 is really just a variation of option 0. In both options you have an active and standby DB2 member. Option 2 just has more pairs of active and passive members. This option is recommended or required to support any SAP requirement that exceeds the capacity of a single machine.

This option also helps the falling back of SAP Java application servers:

- If two or more Java instances concurrently connect to different DB2 members and seamless failover is triggered for one DB2 member, then having the Java

engines failover to a DB2 member that does not already serve Java connections allows you to easily fall back these Java connections to the original DB2 member later.

- If there were already other Java connections established to the failover member, then falling back the Java connections would also cause the Java connections of other Java instances to be redirected.

Another use of this option would be to isolate SAP business components from each other. This is the logical extension of having separate application servers to run specific SAP modules.

Option 3: Two active DB2 members, each with a passive standby member in an independent LPAR

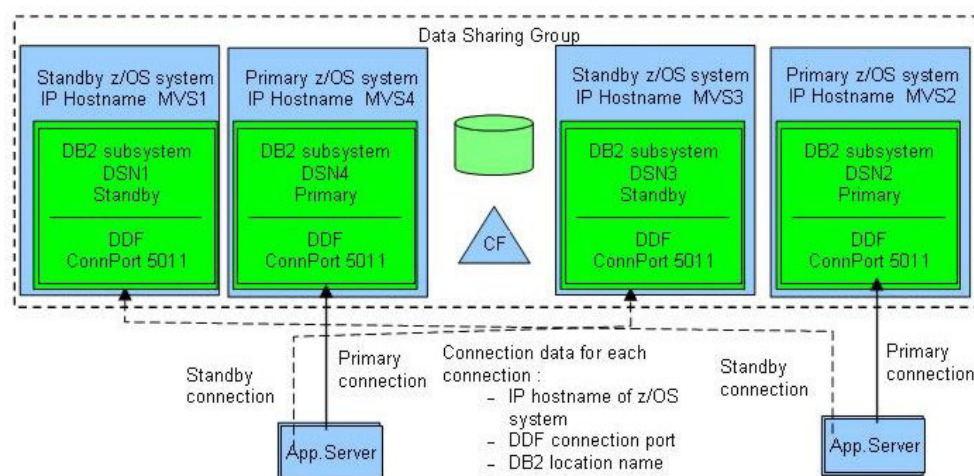


Figure 14. Option 3: Two active DB2 members, each with a passive standby member in an independent LPAR

Option 3 represents the option 2 solution carried to the next level. The inactive standby data sharing members reside in independent LPARs.

Data sharing architecture option 3 was used in early installations that were memory-constrained by the 2 GB of central storage per LPAR. In this configuration each primary DB2 and each standby DB2 would be in separate LPARs. Availability of System z 64-bit hardware and the 64-bit Real Storage Manager in z/OS have effectively eliminated the need for this option, as more than 2 GB of central storage can be made available to a single LPAR.

How many data sharing groups?

A typical SAP core landscape consists of a development system, a quality control system, a stress test system, and a production system. Optionally, one might decide to have a training system, a technical sandbox, or a production support system.

It is common these days for businesses to roll out the other SAP technology components such as SAP Business Intelligence (BI), and Customer Relationship Management (CRM) and so on to support the next generation of mySAP Business Suite solutions. So it is quite common to have separate SAP landscapes for each SAP component.

Whatever SAP components or solutions you are implementing, the total number of SAP systems to build and maintain can add up quickly. It seems that every group involved in implementing an SAP system or solution wants their very own system to work with. Of all those systems, how many should be configured for high availability?

You already may have decided that your SAP production system must be configured to be highly available. Therefore, the production must be configured at the very least to run in DB2 data sharing mode.

What about the non-production SAP systems? The answer is not so easy. It depends on your service level agreement (SLA). Some SLAs require that even the development system be highly available. It is very costly to have developers that cannot work because the system is unavailable. Whatever your SLA, we recommend that you configure at a minimum one other SAP system for DB2 data sharing in your promote-to-production landscape. This system is where you would test your code or configuration changes to verify that there are no problems related to running them in your data sharing setup. Your production system is not the place you want to learn that your changes do not work with DB2 data sharing.

Which non-production system should be configured for data sharing? It depends on how soon or late you want to test your changes with data sharing. Applications will run just fine with data sharing when doing single-user or component-level tests. The story might be quite different for stress tests. As the number of users running against different systems increases, you might have a bigger potential for resource contention, so we recommend that your other data sharing system is either your quality control system or your stress test system if you have one.

We recommend further that you consider having at a minimum one additional data sharing system in each of your SAP landscapes where your business needs require that you have high availability for your production system. Each SAP component shares common technology, but there is also non-common functionality. A more important thing to keep in mind is specific landscape configuration work. So it is recommended that you have one additional DB data sharing system per SAP Landscape.

How many sysplexes?

So far we have concentrated on figuring the number of data sharing systems to ensure that the application changes and SAP basis changes do not cause any problems with data sharing. What about the infrastructure changes such as coupling facility changes? What system should the infrastructure group use to test their changes? The infrastructure group should consider building a Parallel Sysplex with a data sharing system that is independent of the production and non-production SAP systems. It is sufficient to have one Parallel Sysplex for the infrastructure group. There is no need nor benefit to have one technical sandbox system per SAP landscape. This approach, while consistent, would be cost-prohibitive. Some large customers run the production and non-production SAP systems in separate sysplexes. Such a configuration allows separating the shared file systems and the scope of SA. It also avoids encountering the limit on the group buffer pools per Coupling Facility.

How many data sharing members?

After you have decided that you need DB2 data sharing, the next question is how many data sharing members are required for each highly available system. The

answer to this question depends on the data sharing option you are implementing. The option you choose depends on the sizing estimate for your proposed production system or systems.

For an option 0 production system, you need only two data sharing members per data sharing group. The primary data sharing member does all of the work, and the secondary data sharing member is only a standby. We call this *passive data sharing*. This option is valid as long as your workload does not exceed the capacity of the System z box or production LPAR for SAP.

For an option 1 production system you have active workload distributed between two or more members of a data sharing group. Assume that you are configuring a two-way data sharing system. If one system fails or becomes unreachable, the workload will be moved to the surviving data sharing member. If you want the system to perform in failover mode with same level of throughput as in non-failover mode, then you must ensure that there be sufficient extra CPU capacity, memory capacity, and I/O bandwidth to handle the failed over work in one System z box or LPAR. Basically, you must double the capacity of each System z box. A System z box fails so rarely that it may not be so important to have all of that extra capacity ready for a failover situation that will rarely happen.

While DB2 V8 and V9.1 have moved a number of storage structures above the 2 GB bar in the DBM1 address space, please keep in mind that some structures like thread storage still reside below the 2 GB bar. Every SAP work process that connects to the surviving data sharing member will consume from 1.3 to 2.5 megabytes or more, so you must plan for the maximum number of DB2 threads per data sharing member. For details on virtual storage planning in DBM1, see the *SAP Database Administration Guide for SAP NetWeaver on IBM DB2 for z/OS*.

Another possible option 1 production system could have three data sharing members in a group. If one data sharing system fails with this version of option 1, you have the option to redistribute the workload to one of the surviving members or to redistribute the workload evenly among the surviving members. When making this decision, the main choices are to minimize DB2 inter-systems interest or not overallocate DB2 DBM1 virtual storage.

To minimize DB2 inter-systems interest, ideally you would move all of the workload from the failing data sharing member to one of the surviving data sharing members. This might lead to overallocation on DBM1 virtual storage. On the other hand, to prevent possible overallocation of memory, you could evenly distribute the workload among the surviving members. However, this might increase DB2 inter-systems interest between the two surviving members. Which is the best choice to make? It is better to have the system available providing reduced throughput than to over allocate DBM1 virtual storage and risk another abend that would reduce throughput even more. Therefore, we recommend that option 1 systems redistribute the workload evenly among the surviving data sharing members.

To minimize DB2 inter-systems interest and prevent the overallocation of memory in a failover situation, we recommend that you implement option 2 instead of option 1. Option 2 eliminates DB2 inter-systems interest, because all workload from the failing DB2 data sharing member would move to a standby member. Also, no part of the surviving primary data sharing members would be affected. This standby member could be started, ready, and waiting in the same LPAR as the primary or in another LPAR. Option 2 eliminates the possibility of overallocation of DBM1 virtual storage, because the failover happens to an empty data sharing

member that is configured exactly the same as the primary. In the event that only some of the SAP work processes failover to the standby data sharing member, a corresponding number of SAP work processes would be eliminated from the primary data sharing member. There would be no overallocation of virtual storage, but there could be inter-systems interest, but only for that portion of workload running on those data sharing members.

We recommend that you configure at least one of your non-production data sharing systems the same way as your production system. On the one hand, this makes management of your system landscape easier, and on the other hand you might detect potential bottlenecks or setup problems within your test environment, if it is an exact copy.

Failover design

As explained earlier, the notion of standby DB2 members was introduced for several reasons: it is less disruptive to surviving DB2 members (no competition for buffer pools or dynamic statement cache), and it reduces the need to ensure that there is sufficient DBM1 virtual storage to absorb the movement of a large number of SAP work processes (hence DB2 threads). A standby DB2 member also benefits the falling back of Java connections to their primary DB2 member.

There are two groups of companies doing SAP DB2 sysplex Data sharing:

- Those who were motivated primarily by near-continuous availability (secondarily the ability to level load across the multiple CECs required for no single point of failure).
- Those who had both a scalability and availability objective.

The first group typically implements option 0 and frequently is characterized by having many SAP production systems (that is, many production SIDs). Generally they will place half of the production DB2 members in a production LPAR on one CEC and the other half of the production DB2 members in a production LPAR on the other CEC. Each production LPAR will have the standby DB2 members for the other CEC.

Many of the companies pursuing both scalability and high availability have three or more CECs in their sysplex. A typical configuration in a three-CEC sysplex would have a primary DB2 on each CEC with a standby DB2 member residing in each production LPAR that contains a primary DB2, as shown in the following table:

Table 3. Large company using architecture option 2

Machine	CEC1		CEC2		CEC3	
LPAR	MVS1		MVS2		MVS3	
DB2 member	PR11	PR1A	PR12	PR1B	PR13	PR1C
App. Srvr. Grp. 1	Primary			Standby		
App. Srvr. Grp. 2	Primary					Standby
App. Srvr. Grp. 3			Primary			Standby
App. Srvr. Grp. 4		Standby	Primary			
App. Srvr. Grp. 5				Standby	Primary	
App. Srvr. Grp. 6		Standby			Primary	

In the event of planned or unplanned loss of one of the production environments (for example, CEC1), half of the application servers will reconnect to the standby

DB2 member on CEC2 and half will connect to the standby member on CEC3. Although this is more complex than simply moving all of the application servers to one standby DB2, it does offer workload management benefits. Assuming that each of the three primary DB2s were servicing one-third of the total workload, half of this (or one-sixth of the total workload) will be moved to each standby member in the surviving CECs. When coupled with the WLM ability to differentiate priorities (goals, importance, velocity) based on SAP work process type, very high interactive service levels can be maintained with minimized disruption to the surviving CECs. This enables us to minimize the purchase of extra capacity to support failover.

Chapter 6. Backup and recovery architecture in data sharing

In this chapter, we focus on how our usual SAP backup recovery procedures can be affected when we start working in a data sharing environment, and the adjustments we must make on these procedures. We focus on Disaster Recovery and Homogeneous System Copy in an SAP environment.

A comprehensive description of the SAP backup and recovery procedures, both for data sharing and non-data sharing, is given in the SAP SDN document *Casebook - DB2 Backup, Recovery and Cloning for SAP Environments*, which is available at:

<https://www.sdn.sap.com/irj/sdn/db2>

This chapter includes the following sections:

- Disaster recovery considerations
- Homogeneous system copy considerations

Data sharing considerations for disaster recovery

Another important aspect to consider, after deciding to enable data sharing for SAP on DB2 on z/OS, is the need to introduce changes in the disaster recovery procedures to accommodate the new configuration. We describe the most important concepts and different options for implementing a disaster recovery strategy with data sharing, 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.

Detailed descriptions of disaster recovery options can be found in the IBM DB2 *Administration Guide*. Specific information about data sharing is available in the DB2 publication *Data Sharing: Planning and Administration*. For SAP, good references are the IBM Redbook *SAP R/3 on DB2 for OS/390: Disaster Recovery*, SG24-5343, and documentation about split mirror backup/recovery solutions, which can be found at:

<http://www.storage.ibm.com/hardsoft/diskdr1s/technology.htm>

Configuring the recovery site

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 MVS systems as you want.

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

- Run a multi-system data sharing group.

Backup and recovery

The local site is most likely configured this way, with a Parallel Sysplex containing many CPCs, MVS systems, and DB2s. This configuration requires a coupling facility, the requisite coupling facility channels, and the Sysplex Timer or STP.

The advantage of this method having the same availability and growth options as on the local site.

- Run a single-system data sharing group.

In this configuration, all DB2 processing is centralized within a single System z server capable of supporting the expected workload. With even a single CPC, a multi-member data sharing group using an internal coupling facility must be installed. After the DB2 group restart, all but one of the DB2s are shut down, and data is accessed through that single DB2.

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

- Neither a Sysplex Timer nor STP is needed, as the CPC 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 inter-DB2 R/W interest, and the requirements for the coupling facility are:

- A LOCK structure (which can be smaller)
- An SCA

Group buffer pools are not needed for running a single-system data sharing group. However, small (at least) group buffer pools are needed for the initial startup of the group so that DB2 can allocate them and do damage-assessment processing. When it is time to do single-system data sharing, remove the group buffer pools by stopping all members and then restarting the member that is handling the workload at the disaster recovery site.

Remote site recovery using archive logs

Apart from these configuration issues, the disaster recovery procedural considerations do not greatly affect the procedures already put in place for a single DB2 when enabling data sharing. All steps are comprehensively documented in the *IBM DB2 Administration Guide*.

The procedure for DB2 data sharing group restart at the recovery site differs in that there are steps ensuring 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.

In order to deallocate the LOCK structure and the 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.

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

First, 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 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 groupwide point of consistency whose 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 we are going to conditionally restart all the members of the group, we must find a common point in time on the log in order to provide for consistency throughout the group. We will have to find the lowest ENDLRSNs 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, it is important to remember that each member's BSDS data sets and logs are available. Also, 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, as with any other user database, exist only once in the data sharing group and only have to be defined and recovered once from any of the active members.

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

After all members are successfully restarted, 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 listed above. Start the members that run in LIGHT(NO) mode first, followed by the LIGHT(YES) members.

Backup and recovery

You can continue with all of the steps described in “Remote site recovery from disaster at a local site” in the *DB2 Administration Guide*.

Using a 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.

Tracker site recovery

Using DB2 for z/OS V8 or later, we can take advantage of the new utilities to perform tracker site recovery. These steps can be used:

- Use Backup System to establish a tracker site.
- Periodically send active, BSDS, and archive logs to tracker site (Metro Mirror, Global Mirror, z/OS Global Mirror, FTP, or tapes).
- Send image copies after load/reorg log(no).
- 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 IBM DB2 publications *Administration Guide* and *Data Sharing: Planning and Administration*, and in the IBM Redbook *SAP R/3 on DB2 for OS/390: Disaster Recovery*, SG24-5343.

GDPS[®] infrastructure for disaster recovery

GDPS stands for Geographically Dispersed Parallel Sysplex[™]. It is a multisite application that provides the capability to manage:

- The remote copy configuration and storage subsystems
- Automated Parallel Sysplex tasks
- Failure recovery

Its main function is providing an automated recovery for planned and unplanned site outages. GDPS maintains Multi-Site Sysplex, in which some of the MVS images can be separated by a limited distance. GDPS follows the sysplex specification of being an application independent solution.

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

GDPS supports three data mirroring technologies:

1. Metro Mirror (formerly 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 executes failover.
 - Distance between sites up to 40 km (fiber).
 - Provides for both: 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 manages secondary data consistency.
 - GDPS executes Parallel Sysplex restart.
 - Supports any distance.
 - Provides only a disaster recovery solution.
3. Global Mirror with:
 - Asynchronous data mirroring.
 - Supports any distance.
 - Disk-based technology
 - Supports mix of CKD and FBA data

The following is an example of multi-functional disaster recovery infrastructure using GDPS and Metro Mirror to provide all the elements of a backup and recovery architecture. It takes the capabilities of both DB2 V8 and older DB2 releases into account and includes:

- Conventional recovery, to current and to a prior point in time
- Disaster recovery
- Fast system copy capability to clone systems for testing or reporting
- A corrective system as a “toolbox” in case of application disaster
- Compliance with the high availability requirements of a true 24x7 transaction environment based on SAP

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

By using the command to freeze GDPS, the need for SET LOG SUSPEND is avoided even if the non-disruptive DB2 BACKUP SYSTEM utility is not used. In this way, data backup is obtained without production disruption. No loss of transactions and data is encountered, even during split mirror phase. The infrastructure provides for a corrective system as a snapshot of production that can be obtained repeatedly throughout the day.

The components of this sample solution are IBM System z, 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 functionality, and SAP/IBM replication server for application high availability.

The following figure shows the GDPS solution landscape.

Backup and recovery

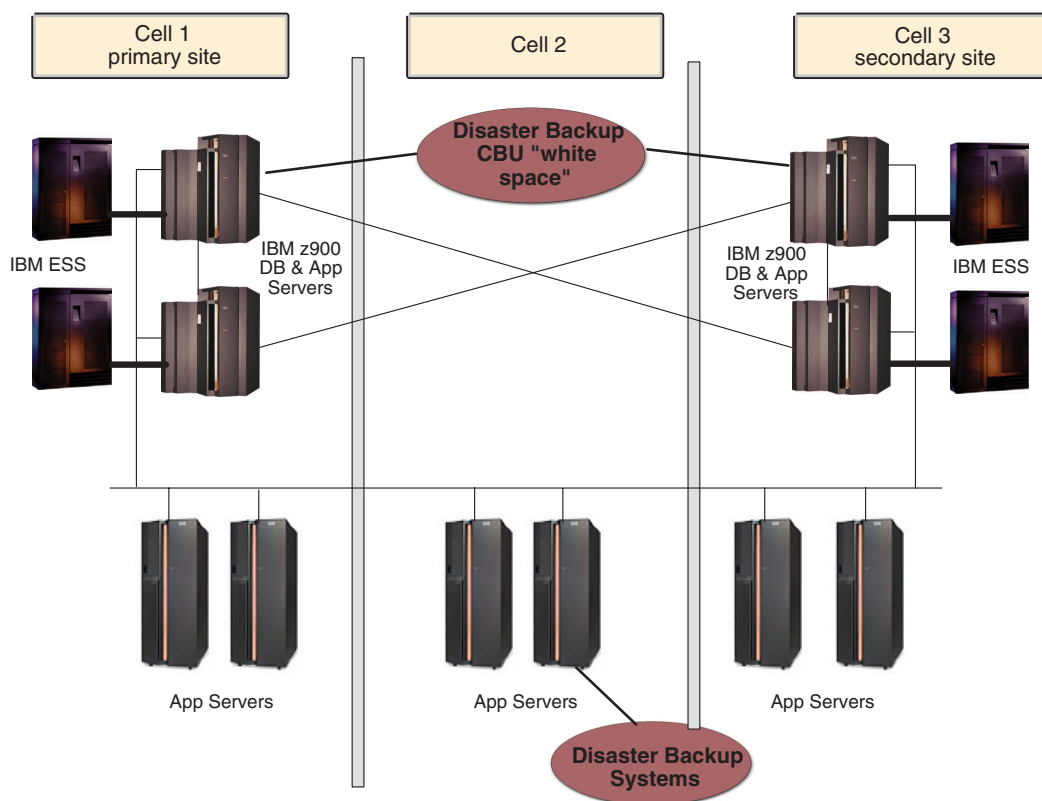


Figure 15. Example of high availability with GDPS configuration

This configuration is made up of two sites and three cells. (Cell 2 is where the corrective system is started.) The three cells are totally encapsulated and safe against floods, earthquakes, and so on. 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 in order to keep the same DB2 data set names for the corrective system. In future versions of FlashCopy, this will not be a requirement.

DS8000 primary and active set of disks is located on the primary site and, using Metro Mirror, they are mirrored to the secondary site. If the BACKUP SYSTEM utility is employed to copy the data, the DS8000 FlashCopy activity takes place at the primary site. Otherwise all DS8000 activity takes place at the secondary site. The design keeps symmetry between both sites, having 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.

If BACKUP SYSTEM is not used, this infrastructure uses the GDPS *freeze* command to suspend all data storage operations temporarily, and initiates FlashCopy at the secondary site. The mirror is split until the end of FlashCopy. By using the BACKUP SYSTEM utility introduced in DB2 V8, it is not necessary to split the mirror to get a non-disruptive backup.

Figure 16 illustrates the process of obtaining a non-disruptive volume backup if the DB2 V8 BACKUP SYSTEM utility is not used.

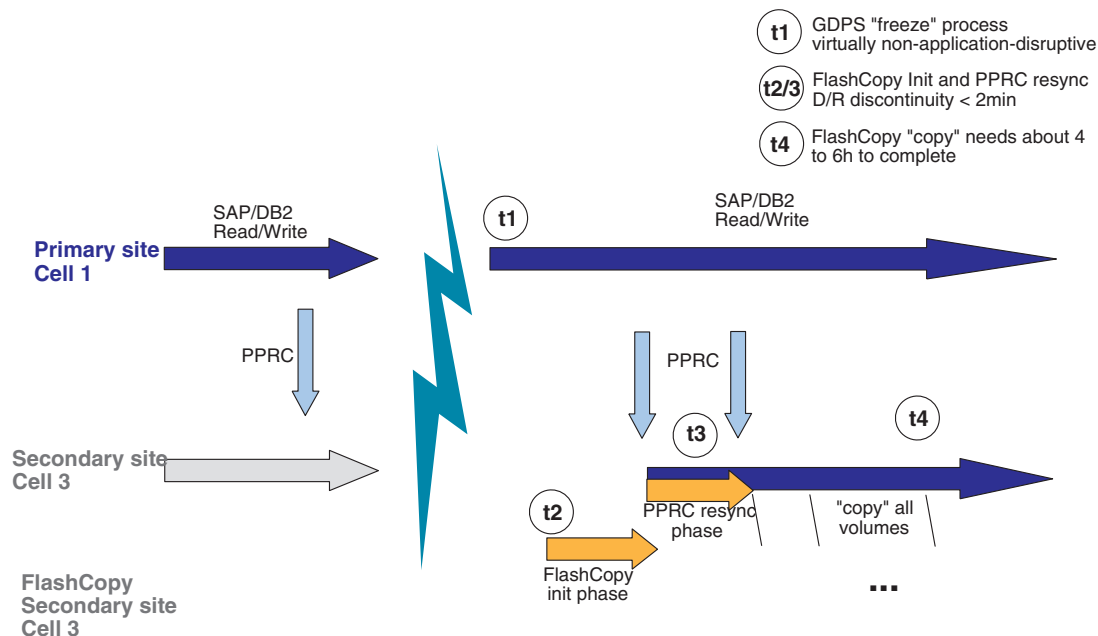


Figure 16. Process for obtaining a non-disruptive volume backup without the BACKUP SYSTEM utility of DB2 V8

Unlike the DB2 log suspend method, t1 in the GDPS freeze process is just a moment, not a duration. The *freeze* command may keep the primary site volumes frozen for one second. During this time frame DB2 looks stopped and Metro Mirror is split between both sites. Immediately, activity continues normally on primary site while, at the secondary site, the initial FlashCopy phase is taking place.

At the end of the FlashCopy initial phase (t2), Metro Mirror synchronizes the volumes on both sites.

Between t1 and t3 (several minutes for large databases) there is a possibility of losing transactional data in the event of disaster failure. The reason is that the mirroring is not active during this interval. Using DB2 V8's BACKUP SYSTEM utility, this gap has been closed, because it is not necessary to split the mirror.

One way to solve this problem is to exclude the second active log copy of all members (in the primary site) from this mirroring, and enable some kind of backup of this active log. A recovery using this backup could provide transactional data until the last moment.

The approach that exploits the BACKUP SYSTEM utility with DB2 V8 is slightly different. Since BACKUP SYSTEM is non-disruptive, the Metro Mirror relationship

Backup and recovery

between the primary and secondary sites does not need to be split. At the primary site, volume-based copies can be taken at any time with the BACKUP SYSTEM utility. Due to the Metro Mirror secondary status of volumes at the secondary site, the copies cannot be taken there. To have the backups available at both the primary and secondary sites, the Copy Pool Backup storage group, which contains the backup target volumes, can be mirrored to the secondary site using Metro Mirror.

Homogeneous system copy in data sharing

Under normal conditions, sooner or later every SAP installation finds the need to perform an efficient homogeneous system copy (HSC). Customers use SAP homogeneous system copy for various reasons:

- Application testing and quality assurance
- System function test
- Production maintenance
- Reporting
- Data mining
- Training

SAP supports two methods for performing an HSC:

- Using SAP export/import tools
- Using database-specific tools

The SAP export/import procedure uses a standard SAP-supplied transaction to export the data from the source database to a flat file and then import the data into the target database. This process is not recommended for large production SAP environments. Typically, it is used with small systems that are being used in pilot projects or some development efforts. The time it takes to accomplish the export-import process with large production systems makes this process prohibitive.

Therefore, we focus on the second method, which is the most commonly used in SAP installations. Our starting point will be the standard procedure for DB2 for z/OS described in the *SAP System Copy Guide* documentation (which is available from <http://service.sap.com/systemcopy>). In the following, we discuss the procedural changes required to support a database server that has enabled DB2 data sharing.

The aim of this section is to present concepts, not to be exhaustive in the steps sequence. For a complete review of the procedure, reference the detailed steps and considerations, including data sharing, given in the Redbook *SAP on DB2 for z/OS and OS/390: DB2 System Cloning*, SG24-6287.

For a detailed description on how to take advantage of the *DB2 Cloning Tool for z/OS* for automated and accelerated DB2 subsystem cloning, refer to the SAP SDN document *Casebook - DB2 Backup, Recovery and Cloning for SAP Environments*, which is available at:

<https://www.sdn.sap.com/irj/sdn/db2>

Planning for homogeneous system copy in data sharing

When planning for homogeneous system copy for a source system that is a data sharing group, consider the following issues:

- What is the DB2 data sharing configuration of the target system?
- Which method are you going to use to obtain the copy?

It is not uncommon to find in some installations that the production DB2 system has been configured for high availability, while the non-production DB2 systems have not. Usually this is done to conserve resources. There could be instances of a non-production system being non-data sharing or, if it is data sharing, not having the same number of members as the production system. In this case, we could find a different group configuration between source and target system.

However, if it is determined for availability reasons to obtain the source system copy using online processes (fast copy volume solution and—before DB2 V8—set log suspend), the target system configuration has specific requirements for facilitating group restart and retained lock resolution. If the source system DB2 data sharing group is going to be quiesced and stopped while obtaining the copy, the requirements on the target configuration are not as stringent. Be aware that certain SAP Java applications may support offline copies online because they need to keep the database and file system in sync. Check the SAP documentation for details.

Review of HSC in non-data-sharing

In order to understand the implications of the issues involving source and target systems configuration and whether the source system copy is obtained online or offline, we first must review the normal homogeneous system copy method for non-data sharing to non-data sharing.

The HSC method is based on copying the entire DB2 system from one environment to the other. If the copy is performed *offline*, all objects need to be quiesced (no uncommitted units of recovery) prior to the copy process. If the copy is performed *online*, we must perform a SET LOG SUSPEND, take the fast volume copy, and perform SET LOG RESUME to continue running, or use the BACKUP SYSTEM utility introduced in DB2 V8.

At some point there must be a step to rename the data sets to the target environment HLQ. This rename can be done:

- During the DFDSS logical copy if using the offline method
- With DFDSS and an interim LPAR if using an online copy
- With a tool from an independent software vendor (ISV)
- With DS8000 disks, using the new features of FlashCopy at the data set level

In the copy we must include the following data sets:

- DB2 log data sets
- DB2 BSDS data sets
- DB2 system data sets
- SAP data sets
- (Optionally) SMPE target libraries

Now, assuming that all of the procedures, parameter libraries, and MVS definitions have been established for the target DB2 environment, prepare the start of the DB2 target system.

In a non-data sharing to non-data sharing HSC, the source system BSDS data sets can be copied into the target system BSDS data sets and used for restart of the target system. However, the VCAT alias and the active log data sets must be

Backup and recovery

changed to the target system's VCAT and active log data set names. The modifications can be performed with the stand-alone utility DSNJU003. The only other modification that might be required is the DDF information. There is no requirement for a conditional restart card.

The restart of the target system varies depending on whether the source system copy was obtained online or offline. Restarting from an online copy requires access to the DB2 catalog and directory and SAP tablespaces in order to recover any outstanding units of recovery or externalize unwritten pages that existed at the time of the log suspend. At the time of target system restart, the VCAT stored in the DB2 catalog tables SYSSTOGROUP, SYSTABLEPART, and SYSINDEXPART is still the VCAT from the source system. To avoid access to the source system's VSAM data sets, you must restart the target system with DSNZPARM DEFER ALL.

During the restart of the target system from a source system offline copy, there should not be any units of recovery to resolve or unwritten pages to externalize. However, it is still recommended to start with DEFER ALL to ensure that the target system does not try to open any of the source system VSAM data sets.

After the DB2 target system has restarted, the temporary workspace tablespaces for the target system must be defined and created. Then all of the DB2 steps necessary to alter the VCAT alias, in all of the defined storage groups, must be performed. Starting with DB2 V9.1, the CATMAINT utility can rename the VCAT alias (described in more detail in the *SAP System Copy Guide*).

After the VCAT alias has been altered to the VCAT for the target system, DB2 opens the VSAM data sets for the target system, instead of the VSAM data sets for the source system. When performing a homogeneous system copy with a Java stack of release NetWeaver '04, be aware that the database schema name needs to reflect the SAP system name. With NetWeaver 2004s, this restriction has been lifted. For details on these steps, refer to the *SAP Homogeneous System Copy* documentation.

Requirements for data sharing

Data sharing introduces the following changes to the procedure:

- Coupling facility structures information cannot be included in the source system copy. For online copy, some committed data pages in the group buffer pools will have to be recovered in the target system.
- BSDSs cannot be exported with the copy because it contains data sharing group information that cannot be changed.
- To move from data sharing to non-data sharing, or to a data sharing group with a different number of members, perform a cold restart. This is only possible when using an offline copy of the source system's database. This means that all members were quiesced and stopped prior to the copy being obtained.

Designing homogeneous system copy in data sharing

In order to apply the modification to the procedure introduced by the data sharing conditionings, we consider two cases:

- Data sharing to data sharing (with the same number of members) copy
- Data sharing to non-data-sharing copy

In either case, when the target system is also data sharing group there is no other option than performing a target system group restart to allocate new structures in

the coupling facility. Therefore, preparation steps must be performed to assure good CFRM structure definitions and enough space in the coupling facility for the structures.

Data sharing to data sharing

We now describe the two copy possibilities: online and offline.

Online copy design considerations: If an online copy is used to restart a DB2 data sharing group at the target, an equivalent number of DB2 members must be restarted at the target system to ensure that the log information from all members at the source can be processed. This is necessary to roll back transactions that are in process on the source system at the time the online copy is taken.

In order to support group restart via the coupling facility, it is necessary to have the same number of members in the target system as in the source system. However, not all of the members in the target system have to be configured as robustly as a member that actually supports a workload. In other words, the active logging configuration must be sufficient to support group restart and nothing else. The configuration to support group restart consists of each target member having BSDS data sets, and the current active log from the source member available and registered in the target member's BSDS.

It may not be necessary to restart all members in the target system. If a member, or members, of the source system were quiesced and stopped at the time of the copy, these members will not need to be restarted in the target system. However, all active source members must be restarted. This is required in order to resolve local locks held by an active member. The members that are restarted will read the BSDS and the registered active logs of the members that are not restarted and will perform group restart for these peer members.

The restart process can use active or archive logs from the source system. The active log configuration for each member of the target data sharing group can be different from that of the source system members and different from each other.

Many things can be changed in the BSDS via the change log inventory utility (DSNJU003). However, the information about the data sharing group and its members cannot be changed, so it is necessary to keep all BSDSs, belonging to all members, of the target data sharing group intact. That means that we do not use the BSDSs from the source system to perform the restart of the target system. However, there is information in the source system BSDSs that must be recorded in the target system BSDSs in order to accomplish the restart in the target system. Depending on whether the restart is being done with the active logs versus the archive logs, the required BSDS information will vary. This information may include some, but not all, of the following items:

- The suspend LRSN, to be used as the conditional restart LRSN
- The checkpoint taken just prior to the suspend
- The archive log containing the suspend LRSN and the checkpoint
- The active log containing the suspend LRSN and the checkpoint
- The highest written RBA

To ensure the successful use of this information during the restart of the target system, consider creating a skeleton BSDS. See "Creating the skeleton BSDS" in *SAP on DB2 for z/OS and OS/390: DB2 System Cloning*, SG24-6287.

Backup and recovery

Offline copy design considerations: During the offline copy, all members of the source data sharing group are stopped. There should not be any outstanding units-of-recovery, and all data pages in the virtual buffer pools should have been externalized (written to disks). In other words, all data managed by the source system is quiesced and consistent.

The process is similar to the online copy procedure except that the copy is made with the DB2 group stopped, and the BSDSs print log map from each source member should be obtained while the group is stopped. With this information we define the restart of the target DB2 data sharing group. The restart process should be faster, for there are no page sets to recover.

Data sharing to non-data-sharing

This DB2 system cloning configuration involves moving the data from a DB2 data sharing group to a non-data sharing DB2.

This step is similar to disabling data sharing in one DB2 environment. There is no other way than performing a cold restart. For this reason the database must be copied in a state of consistency, which can only be achieved with offline copy.

Because the target system is non-data sharing, the DB2 system is managed by RBA and not LRSN. The target system original BSDS and active logs are used. The information required to perform the cold start would be registered in the BSDSs of the target system.

As an example, suppose our source DB2 system has a two-member data sharing group. The target system is a non-data sharing DB2. The highest used LRSN of our source system could be used as the restart RBA of our target system. Example 11-5 shows the highest used LRSN in the source system.

```
TIME OF CHECKPOINT 18:00:08 JUNE 18,2001
BEGIN CHECKPOINT RBA 0012F391263C
END CHECKPOINT RBA 0012F391448C
END CHECKPOINT LRSN B6016DA1E435
```

The following example shows the cold start at the target system with the source LRSN used as target start RBA.

```
//ACTLOG EXEC PGM=DSNJU003
//STEPLIB DD DISP=SHR,DSN=DSN610.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DB2V610B.BSDS01
//SYSUT2 DD DISP=OLD,DSN=DB2V610B.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
CRESTART CREATE,STARTRBA=B6016DA1F000 ,ENDRBA=B6016DA1F000
/*
```

As previously noted, testing environments with all of the details to plan and prepare the procedures and recommendations can be found in the Redbook *SAP on DB2 for z/OS and OS/390: DB2 System Cloning*, SG24-6287.

Part 3. Network

Chapter 7. Network considerations for high availability	71
Introduction	71
General recommendations	72
Hardware considerations	72
z/OS communication software considerations	73
Considerations for the Linux on System z application server	73
Multiple Linux on System z guests under z/VM	74
DB2 connection failover recovery mechanism	75
OSPF protocol as a recovery mechanism	76
Notes concerning AIX 5.3 and Path MTU discovery	77
Virtual IP Address (VIPA) as a recovery mechanism	77
Recommended setup for high availability connections between client and server	79
OSPF and subnet configuration aspects	79
VIPA and Source VIPA functions on remote application servers	80
Recommended setup for a high availability network	81
Alternative recovery mechanisms	83
z/OS VIPA usage for the high availability solution for SAP	85
Timeout behavior of the client/server connection over TCP/IP	85
Timeout behavior of the AIX application server	86
Client connection timeout	86
Client transmission timeout	86
Recommended values	87
Client idle timeout	87
Recommended values	87
Timeout behavior of the Linux application server	88
Client connection timeout	88
Client transmission timeout	88
Recommended values	88
Client idle timeout	88
Recommended values	88
Timeout behavior of the Windows application server	89
Client connection timeout	89
Client transmission timeout	89
Recommended values	89
Client idle timeout	90
Recommended values	90
SAP maximum transaction time	90
Timeout behavior of the database server	90
Server transmission timeout	90
Server idle timeout	91
DDF-specific keep-alive interval times	91
Resource timeout and deadlock detection interval	91
Resource timeout	91
Deadlock detection interval	92

Chapter 7. Network considerations for high availability

This chapter describes high availability aspects of the network between a remote SAP application server and the SAP database server. In our solution, this means 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 are given. Then, the three recovery mechanisms that 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 gathered by our test team. For the sample definitions of this test scenario, see Appendix A, “Network setup,” on page 297. These sample definitions give you an impression of the necessary implementation tasks.

The chapter concludes with discussions of a description of an alternative recovery mechanism, z/OS VIPA usage, and timeout behavior.

Introduction

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 a System z 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.

Network considerations

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 5.x, and Linux on System z, 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)

In order 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 chapter provides hints and recommendations on how to achieve the transparent level of recovery on z/OS, AIX, and Linux on System z, utilizing communication network options such as:

- HiperSockets – System z 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 HA 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 a SAP outage, read “Alternative recovery mechanisms” on page 83, which describes a much simpler setup.

General recommendations

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.

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

We recommend having 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.

Notes:

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*, SC31-8775.

Considerations for the Linux on System z application server

If a Linux on System z application server runs in one LPAR and the SAP on DB2 database server runs in another LPAR within a single System z 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, System z hardware model, and OSA card under **Networking at:**
http://www.ibm.com/developerworks/linux/linux390/development_restrictions.html.

Network considerations

Make sure you have read SAP Note **1263782** with the title *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 System z and z/OS DB2 Database Server and/or SAP Enqueue Server.

Multiple Linux on System z guests under z/VM®

If you are running several Linux on System z guests (as SAP application servers) under z/VM, we recommend setting 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 System z 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's 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 trunked together 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. For failures that are handled and recovered by z/VM, then this is transparent to the Guest. But remember that z/VM is not handling the TCP traffic only the OSA card. If z/VM does not detect an error then no VSWITCH failover will take 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 (and avoids the use of routers and gateways for local traffic). This may be true of local SAP GUI end users which you may wish to connect to the SAP applications servers running on Linux on System z. 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.

The following figure depicts the use of VSWITCH:

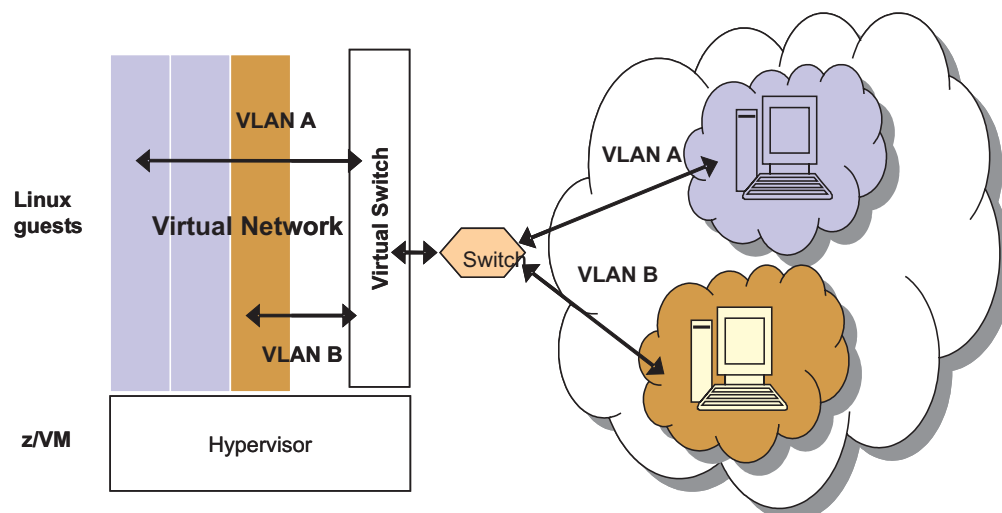


Figure 17. Sample VSWITCH utilization

For a detailed description of how the above features can be utilized by Linux guests, and how your virtual networking configurations can be greatly simplified through the use of these new functions, read the IBM Redpaper *Linux on IBM 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.

DB2 connection failover recovery mechanism

DB2 connection failover (described in "DB2 connection failover" on page 41) 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 datasharing in a sysplex, you can thereby provide redundancy at the database service layer.

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.

In order to implement the recommended solution (see "Recommended setup for high availability connections between client and server" on page 79), DB2 connection failover must be exploited and the following preconditions need to 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.

Network considerations

- All network components need to be duplicated.
- The SAP failover parameters are set up correctly:
 - For ABAP instances, configure the connect.ini profile as described in the *SAP Database Administration Guide*.
 - For Java instances, use the setup described in SAP note 1085521.

It is possible to define different system configurations to handle the failure of one or several components. In the configuration shown in Figure 19 on page 82, 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

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 resends 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 5.x, and Linux on System z. 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 System z

Note: Quagga is a derivative of zebra.

- Routing and Remote Access Services (RRAS) under Windows

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

Notes concerning AIX 5.3 and Path MTU discovery

Beginning with AIX 5.3, path MTU (PMTU) discovery can be used on duplicate routes.

Starting with AIX 5L Version 5.1, you are permitted to use multiple routes to the same destination (including multiple default routes) through the Multipath Routing feature. The cost (hopcount) 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 fastpath`.

Virtual IP Address (VIPA) as a recovery mechanism

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.

We highly recommend running a dynamic routing protocol like OSPF when exploiting VIPAs. We also highly recommend defining a subnet for the VIPAs, different from the subnets of the other IP addresses of the NICs. The figure below illustrates how a VIPA and OSPF work together under z/OS to achieve transparent recoveries from z/OS device or NIC (feature) failures:

Network considerations

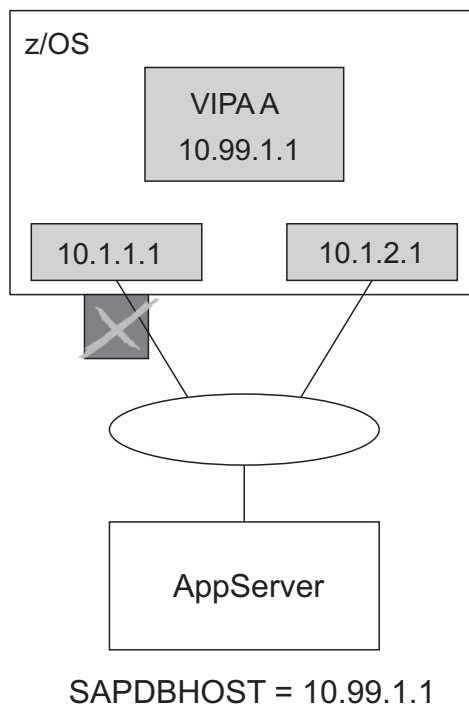


Figure 18. VIPA and OSPF recovery mechanisms under z/OS

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 80 for details).

VIPAs are supported on z/OS, AIX 5.x, and Linux on System z; VIPAs on AIX 5.x are always Source VIPAs. For information on alternative recovery mechanisms on Windows, see “Alternative recovery mechanisms” on page 83.

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 the scenario above. We recommend using static VIPAs for database connections, whereas dynamic VIPAs should be used for movable applications like the NFS server and SAP Central Services.

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

OSPF and subnet configuration aspects

In an SAP on System z 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 will experience 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 19 on page 82, 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 5.x, and Linux on System z.

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 5.x. You need to be aware that a VIPA on AIX 5.x is automatically a Source VIPA. This means that every packet sent out from AIX 5.x on any real interface has the VIPA as its source IP Address. With AIX 5.1, this may cause problems with your current network structure. For example, if you want to use a 10.x.x.x address for the VIPA subnet, then you need to ensure that the 10.x.x.x address can be routed within all networks to which the AIX application server is connected.

With AIX 5.2, the VIPA feature has been enhanced to give the administrator greater control to select the source address for outgoing packets, and the above problem has been resolved.

The VIPA function is available on Linux on System z via the so-called dummy device. For detailed information concerning the definition of a VIPA under Linux on System z, see "VIPA – minimize outage due to adapter failure" in *Linux on System z - Device Drivers, Features and Commands - May 2008*, SC33-8411-00 available from

<http://www.ibm.com/developerworks/linux/linux390/index.html>

We do not recommend the described use of the Source VIPA utility 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 we be used as gateways for the VIPAs on the local interfaces. For example if we have a dummy0 interface with a VIPA of 10.1.100.1 and 2 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 above two 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. Quagga on SLES-10-SP1 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 new "set src" sub-command.

For this example let us assume that all z/OS VIPAs are in the subnet 10.1.100.0/24, and that we 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

The following figure shows the recommended setup for a high availability network between the SAP application server and the z/OS database server (or NFS, SCS, etc.) that results from the considerations above:

- 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.

Note that this recommended HA network setup allows transparent recovery of most kinds of network outages. If you do not have such requirements for your network availability and if you are willing take the risk that a switch outage means a network outage, and therefore an SAP outage, please read "Alternative recovery mechanisms" on page 83, which describes a much simpler setup.

Network considerations

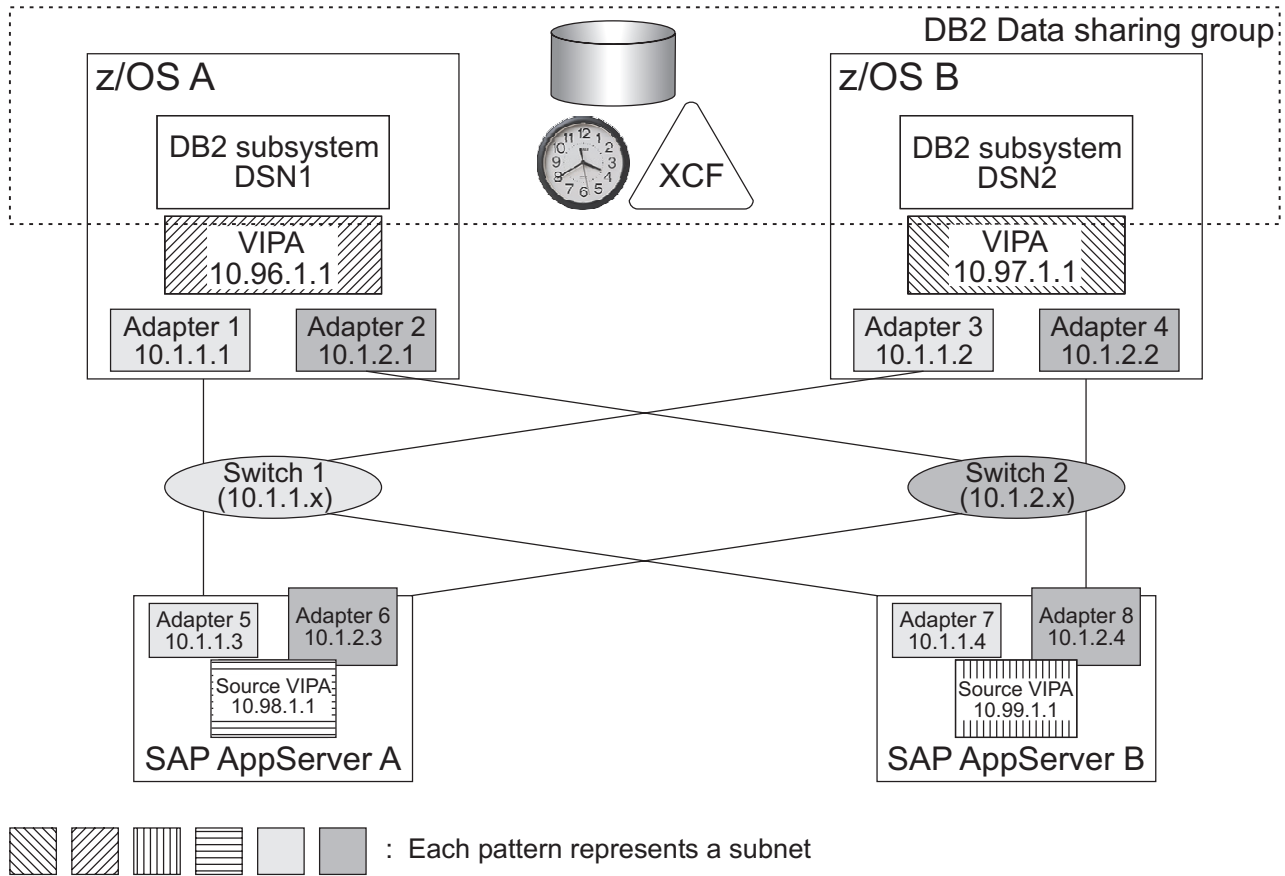


Figure 19. Recommended setup for a high availability network

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.

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

Table 4. 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, we recommend using a value of 10 seconds (or a different value which fits your requirements after careful investigation).

Alternative recovery mechanisms

The following describes a much simpler network setup. You may run with such a setup:

- If you are willing to take the risk that a switch outage means a network outage and therefore SAP outage under AIX or Linux on System z/x
- If you run under Windows platform. Windows does not support the VIPA recovery mechanism. Therefore, other mechanisms have to be employed to recover from Windows adapter failures.

We recommend the following solution:

- exploit the ARP takeover function of the OSA-Express features on z/OS
- utilize the 'adapter teaming' function of Windows adapters on Windows (for example, the teaming function of the IBM Netfinity® Gigabit Ethernet SX adapter)
- use for example etherchannel under AIX
- use for example bonding under Linux.

You implement this solution by connecting two OSA-Express features on z/OS and two adapters on Windows (application server side) to the same network or subnet; a dynamic routing protocol (such as OSPF) or VIPA is not required.

Network considerations

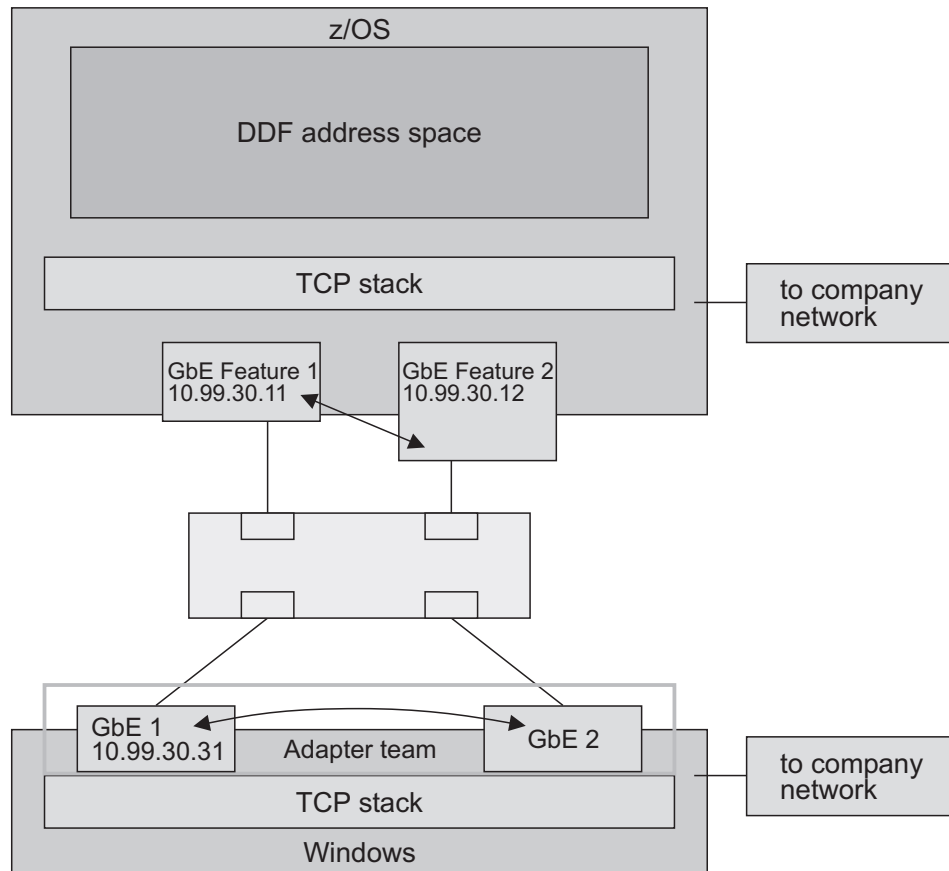


Figure 20. System setup with z/OS ARP takeover and Windows adapter teaming

In such a setup:

- The failure of an OSA-Express feature is handled by the ARP takeover function (MAC and IP address takeover).
- The failure of a Windows adapter is recovered by activating the Windows IP address on the standby adapter of the adapter team.

ARP recovery solutions rely on the fact that hosts will 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. Using a Dynamic routing protocol, *any* network media can be used in combination to provide alternate routes.

If a single network switch is used, this becomes a single point of failure. However, certain switch equipment can be “paired” to create a single LAN segment. In such a configuration, even the loss of one of the two physical switches does not lead to a loss of network connectivity. This type of configuration is discussed in an IBM z/OS Newsletter which you can find at:

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

In particular, refer to the section “No Dynamic Routing protocol? No Problem!” of this document.

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

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

- Static virtual IP address (VIPA) definitions for:
 - z/OS systems hosting DB2 data sharing members
- Dynamic VIPA definitions for:
 - SCS
 - NFS server and/or DFS[™] SMB
 - SAP network interface router (saprouter)

The dynamic VIPA is to be defined as VIPARANGE with the attributes MOVEABLE and DISRUPTIVE:

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

Furthermore, the SOURCEVIPA attribute is needed for all VIPAs.

Normally with the SAP application server running on a non z/OS system, all network session connects are inbound to z/OS and z/OS will honor 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. As we recommend using a Dynamic VIPA for the NFS Server (because it can move across LPARs) then we must set a special Source VIPA just for the NFS Server. This can be done in the z/OS TCP PROFILE(s) by using the SRCIP statement. For example:

```
SRCIP JOBNAME MVSNFSHA 10.101.5.193
ENDSRCIP
```

In the example above MVSNFSHA is the z/OS jobname of our NFS Server, and 10.101.5.193 is the Dynamic VIPA associated with our 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'
//VIPAO00 EXEC  PGM=MODDVIPA,
//          PARM='POSIX(ON) ALL31(ON)/-p TCPIPA -c &VIPA'
```

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

In this section, the timeout behavior of a client/server connection with the TCP/IP communication protocol is described for each of the platforms AIX, Linux on System z, and Windows. In conclusion, platform-independent information is then presented on the maximum transaction time. The timeout behavior applies 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 we recommend to change the default TCP/IP timeouts as described below.

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

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

It is recommended that, to avoid negative effects on system performance, default values be changed 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.

Client transmission timeout

Each time the client sends data to the database server, TCP/IP waits for acknowledgement of this data. TCP/IP retransmits data if acknowledgements are missing. The time period that TCP/IP waits for the acknowledgement 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.

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 "doublings" 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 5. 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 will acknowledge 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).

Network considerations

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

Timeout behavior of the Linux application server

On Linux on System z, 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 the default values be changed 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 System z connection request times out after 180 seconds.

Client transmission timeout

Each time the client sends data to the server, TCP/IP waits for acknowledgement of this data. TCP/IP retransmits data if acknowledgements are missing. The time period that TCP/IP waits for the acknowledgement before it times out is variable and dynamically calculated. This calculation uses, among other factors, the round-trip 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 System z, 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 System z, 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_interval` (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_interval* to 6 seconds.

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

Timeout behavior of the Windows application server

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` and `<Adapter Name>\Parameters\Tcpip`, where `<Adapter Name>` refers to the subkey for a network adapter to which TCP/IP is bound. Values under the latter key(s) are adapter-specific. The parameters mentioned below normally do not exist in the registry. They may be created to modify the default behavior of the TCP/IP protocol driver.

It is recommended that, to avoid negative effects on system performance, default values be changed only after *careful study*.

For more information on these registry values, refer to the online documentation of Windows 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.

Client transmission timeout

Each time the client sends data to the server, TCP/IP waits for acknowledgement of this data. TCP/IP retransmits data if acknowledgements are missing. The time period that TCP/IP waits for the acknowledgement 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 round-trip 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: We recommend running with the default value.

Network considerations

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 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 2 hours and 6 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.

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:

600 + 60 = 660

seconds.

When this time elapses, an ABAP/4 short dump is issued.

Timeout behavior of the database server

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 acknowledgements are missing, TCP/IP retransmits data. The time period that TCP/IP waits for the acknowledgement 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 values of

- *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

which are parameters of the GATEWAY statement in the PROFILE.TCPIP data set.

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 60 is used, meaning that the first keep-alive probe is sent after 60 minutes:

```
KEEPALIVEOPTIONS
INTERVAL 60
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 above, 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. We recommend running 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

Network considerations

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.

Part 4. SAP

Chapter 8. Concepts for a high availability SAP solution

Prerequisites and planning	95
SAP high availability installation	95
Sample two-node setup for a highly available SAP system	96
Architecture components	99
SAP Central Services	99
Old-style enqueue services with the central instance	100
Standalone enqueue server	100
Failover and recovery of SAP Central Services	101
Network	102
File system	106
Failover of the NFS server	107
Database	108
Non-data-sharing	108
Data sharing	109
Remote application server and DB2 connection failover support	109
General information	109
Failover with multiple DB2 members in the same LPAR when using DB2 Connect	110
Application design	111
Failure scenarios and impact	111
Old-style central instance without data sharing	112
Data sharing, DB2 connection failover, double network (single central instance)	114
Enqueue replication and NFS failover: fully functional high availability	116

Chapter 9. Preparing a high availability SAP solution

Software prerequisites	120
Naming conventions	121
Tivoli System Automation for z/OS	121
Conventions used in the SA z/OS policy	124
Tivoli System Automation for Multiplatforms	124
DB2	125
Using the SA z/OS 'DB2 - Best Practise Policy' to perform a light restart	125
File system setup	125
File systems	126
Setting up zFS filesystems	126
SAP directory definitions	127
SAP global transport directory	127
SAP system-wide directories	127
SAP local directories	127
SAP administrator's home directory	128
SAPOSCOL/SAPCCMSR directory	128
NFS server on z/OS	128
How many NFS servers should I run?	128
Which NFS server security model (exports, safexp, or saf) should I use?	129
Security(exports)	129

security(safexp)	129
security(saf)	129
Mount handle databases and the remount site attribute	130
The nlm site attribute	130
NFS client root access	130
NFS Server automation	130
NFS Clients - General Information	130
NFS Client on Linux on System z	131
Tivoli System Automation	132
Setup of Tivoli NetView and Tivoli System Automation for z/OS	132
Tivoli System Automation for Multiplatforms setup	132
Control of remote ABAP application server instances	132
SAP installation aspects	132
SAP license	133
SAP logon groups	133

Chapter 10. Customizing SAP for high availability

Prerequisites	135
Setting up an ABAP SCS instance and/or a Java SCS instance	136
Rationale for enhancing the standard ASCS with additional SAP services	137
Rationale for not running the Enqueue Replication server as an ERS instance	137
General installation sequence	138
Installing and configuring ABAP SAP Central Services (ASCS)	139
Option 1: Installing ASCS with a virtual hostname	139
Setting up Enqueue table replication	141
Adding additional SAP services	142
Option 2: Installing ASCS with a physical hostname	143
Installing SAP with classic CI	146
Verification of ABAP SCS with Enqueue Replication	148
Installing and configuring Java SAP Central Services (SCS)	149
1. Changes necessary if your system was installed with Option 2	150
2. Manually create the SCS instance directory on switch-over nodes	152
3. Manually make the changes to enable enqueue table replication	153
Verification of Java SCS with Enqueue Replication	154
SAP profile parameters	155
Preparing SAP on z/OS for automation	157
C-shell and logon profiles	157
ABAP SAP Central Services (ASCS)	157
Java Central Services	158

ABAP application server instances	159
What the shell scripts do	159
startappsrv_v5	159
stopappsrv_v5	160
checkappsrv_v4	160
rfcping	160
Remote execution	160
Remote control of Windows application servers	161
Java and double-stack application server instances	161
startjappsrv	162
checkjappsrv	163
saposcol	163
sapccmsr	163
Additional SAP setup for RFC connections	165
SAProuter	166
Summary of start, stop and monitoring commands	166
Other installation issues and recommendations	167

Chapter 8. Concepts for a high availability SAP solution

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

We discuss the following:

- Prerequisites and planning
- Architecture components
- Failure scenarios and impact

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 setup redundancy. This means to run one or more (standby) servers where each of the SPOF services can be 'failed over' and restarted independently.

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

SAP high availability installation

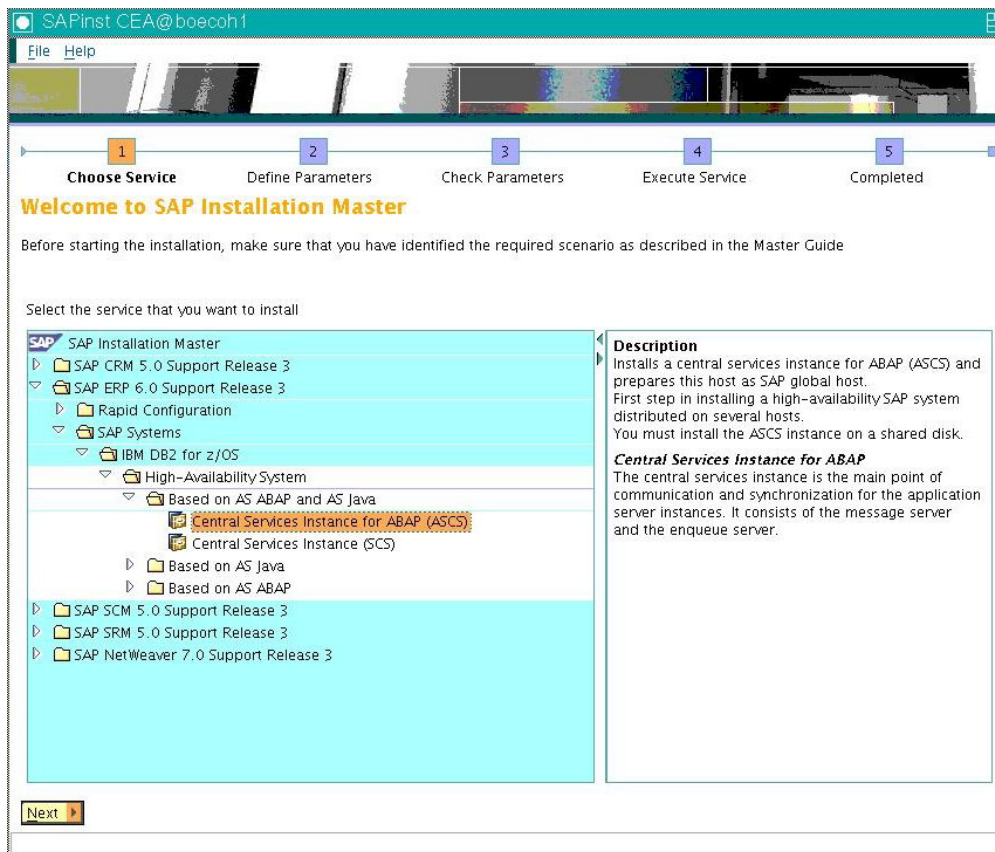
With this background it is a MUST to install the DB2 Database in datasharing mode. Also we highly recommend that you install the ABAP Central Services and the Java Central Services directly with SAPinst using for each service its own virtual hostname.

For example, *for ABAP* you would use the ABAP virtual host name and run in the install directory of the SAP DVD:

```
./sapinst SAPINST_USE_HOSTNAME=ha1ascsv
```

for a SAP system with <SAPSID> HA1 and for installation of the ASCS (the ABAP SAP Central Services).

Use the **High-Availability System** menu option within SAPinst:



Above is an example for an ERP 2005 SR3 installation on SAP on System z running with DB2 z/OS and installing the ABAP Central Services Instance with virtual hostname ha1ascsv. Select the appropriate **High-Availability System** option according to your DB and OS platform.

Now *stop the SAPInst* after ASCS installation and start it again for SCS installation using the *Java virtual host name*:

```
./sapinst SAPINST_USE_HOSTNAME=ha1jscsv
```

That way, the Java SCS is installed with its own virtual host ha1jscsv.

Sample two-node setup for a highly available SAP system

The *minimum* hardware setup consists of a two-node cluster. However, the *preferred and recommended* configuration is a three-node cluster, which is described later in this chapter.

The two nodes are either two *physical machines*, or two *LPARs* where each LPAR runs on a different physical machine. The machines must be connected together via a network, for example via Gigabit Ethernet. Each machine needs access to the DB2 database.

High availability for Application server (AS) is achieved by having at least two application server instances.

1. The 'rest' Central instance
2. Additional Dialog instance on the other node

This 2nd AS, the DI, must be manually configured so that all SAP services which run on the CI installation are also on it. These are:

- Batch service
- Update/Update 2 service
- Spool service

In this way all non-unique SAP services are running on each of the AS and are no longer SPOFs. Use SAP's group logon feature to easily use the redundancy from a SAP user perspective.

Note: in older, or non-HA installations, the only way to make the unique SAP Enqueue and Messages services highly available was to failover and restart the complete Central Instance. This may suggest that it is a good idea to also failover the 'rest' CI or the additional DI Application server. This is not recommended as such a failover or restart takes a long time once the Java stack is part of the AS. In fact the long (re)starting time of a Java application server was the reason why SAP introduced with the Java AS the Java SAP Central Services in order to get rid of the need to failover the CI.

Additionally, an SAP system needs utility programs. If the SAP Router or SAP Web Dispatcher program is used, it must also be made highly available.

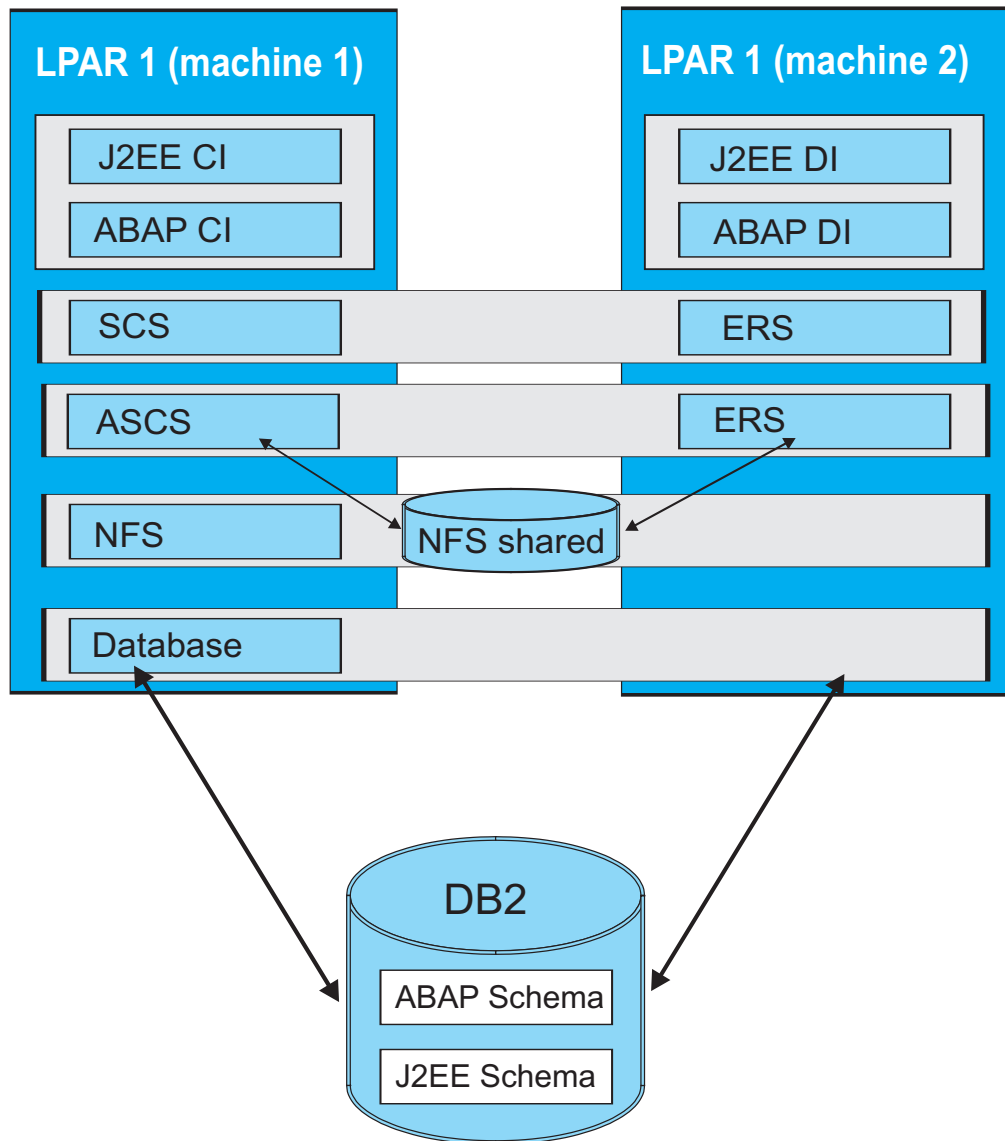


Figure 21. Sample two-node TSA domain

Notes:

1. A *double-stack* application server is physically one instance which runs both the ABAP and the Java stacks.
2. Although it is physically one instance, the SA MP policy separates it into its two logical parts, the ABAP application server and the Java application server. In other words, within a TSA domain, one 'Add-In' application server instance is automated as two logical application server instances, an ABAP application server instance and a Java application server instance.
3. There is a tight relationship between those two logical application servers. The Java instance is always started *after* the ABAP instance. This StartAfter relationship guarantees that starting the Java instance automatically triggers the start of the ABAP instance before.
4. On the other hand, stopping the Java application server does not stop any of the 'Add-In' server processes, it only stops the monitoring java program 'java GetWebPage' which does a primitive health check of Java application server.

Note that using the Tivoli SA MP sample policy (the most current is version 5.2) assumes that the standard naming conventions are used. Those conventions are described in “Tivoli System Automation for Multiplatforms” on page 124. If you have made changes to the mksap script or wrote your own script(s) to set up the SAP resources, you will encounter difficulties with the provided sapctrl scripts because they rely on the standard naming conventions.

Architecture components

The high availability solution for SAP involves the following architecture components:

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

SAP Central Services

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.

In earlier designs related to SAP high availability support, the central instance 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 a SAP gateway process and serves as primary destination for RFC connections.

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

For the high availability solution, the central instance has been disassembled and redesigned into standalone components that operate as SAP Central Services (SCS). The independence of the components allows for more efficient recovery should a component become unavailable, and provides better performance of the enqueue services.

For the sake of simplicity, the following standalone components have been grouped together as SCS:

- Enqueue server
- Message server
- SAP gateway (optional)
- Syslog collector (optional)

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.

HA SAP concepts

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.

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

Old-style enqueue services with the central instance

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

As shown in Figure 22, the enqueue server resides inside a work process. The message flow goes from the requesting work process to its dispatcher, via the message server and the dispatcher of the central instance to the enqueue work process. The response message is sent back the same way.

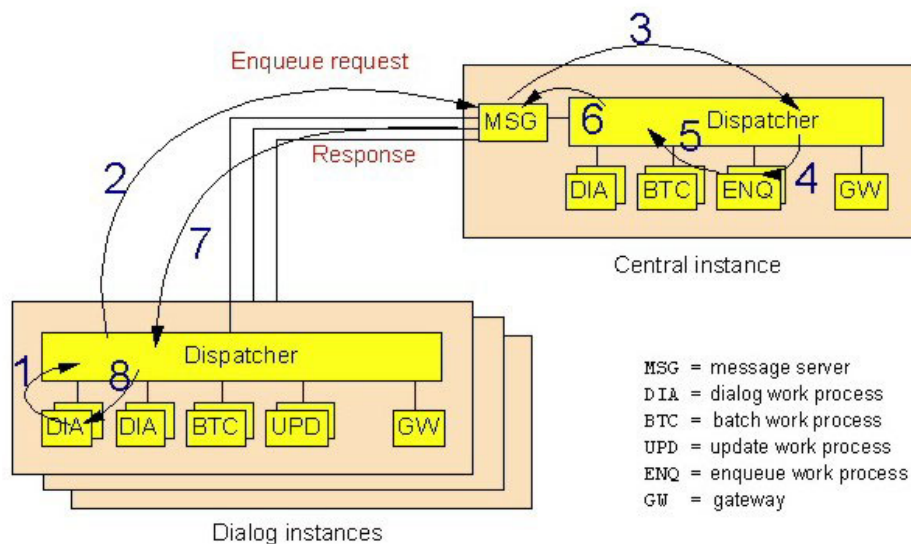


Figure 22. SAP enqueue services with the old central instance concept

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

Standalone enqueue server

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

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

An application server instance connects directly to the enqueue server by using a virtual IP address (VIPA). The message server is no longer in the communication path. See Figure 23.

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 because it runs on a different system, though it may share the same instance directory and instance profile, providing that a shared file system is used.

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

Failover and recovery of SAP Central Services

Figure 23 shows the principal TCP/IP communication paths between the application server instances and the enqueue and message servers. The other SAP components of SCS (gateway, syslog collector and sender) are not shown because they are of minor relevance for the failover scenario.

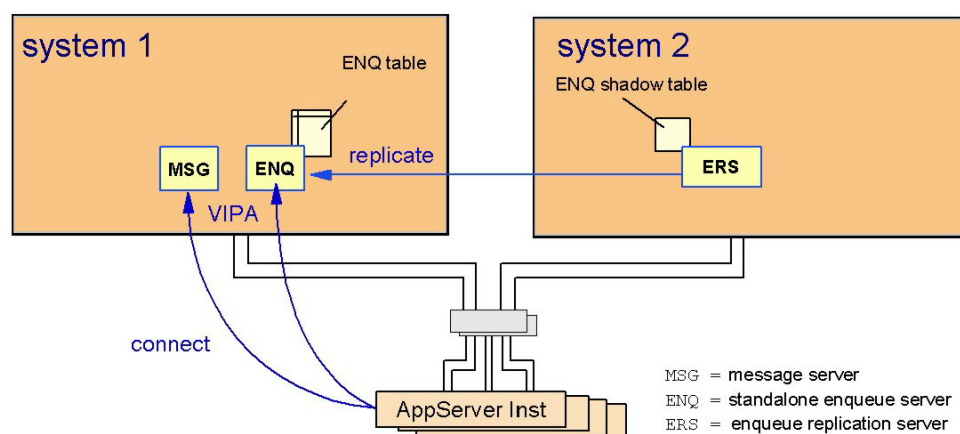


Figure 23. Initial startup of SCS

If the first system fails, the second system takes over the role of the first one, as shown in Figure 24 on page 102:

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.

HA SAP concepts

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.

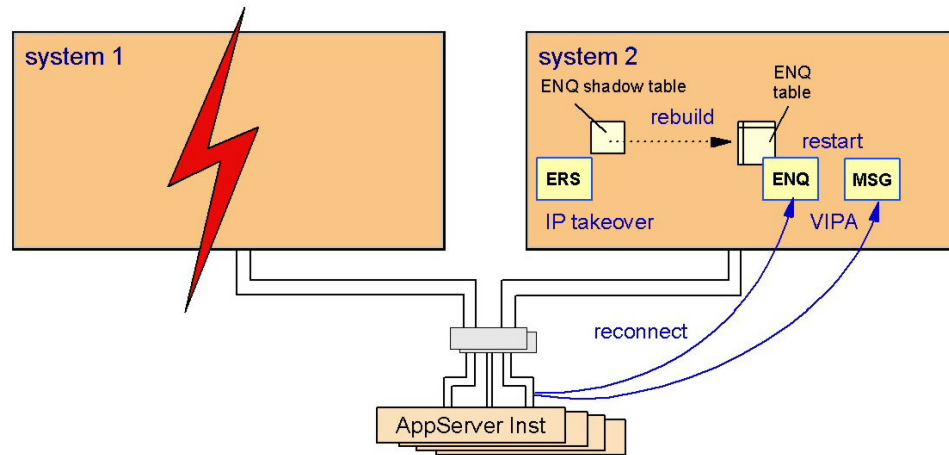


Figure 24. Failure of SCS and recovery of the enqueue table

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 25.

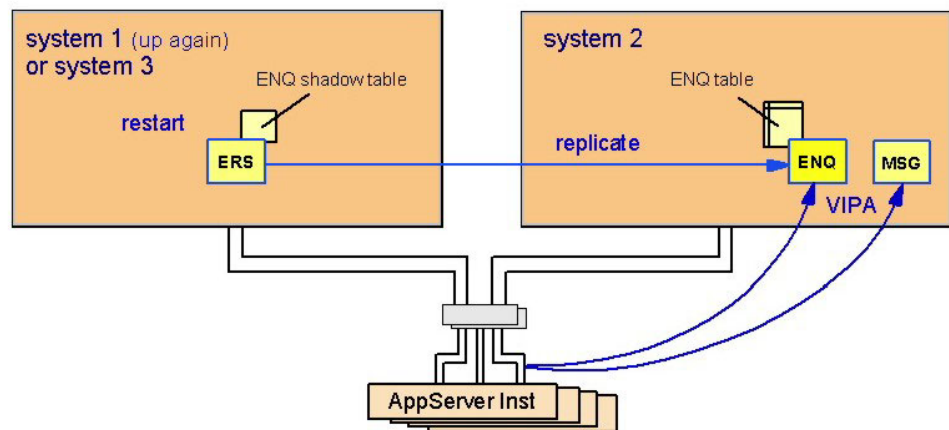


Figure 25. Movement of the enqueue replication server

Network

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

Static VIPAs are used to locate a host while *dynamic VIPAs* 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 26 shows the general concept of a fault-tolerant network with duplicated network components and VIPA.

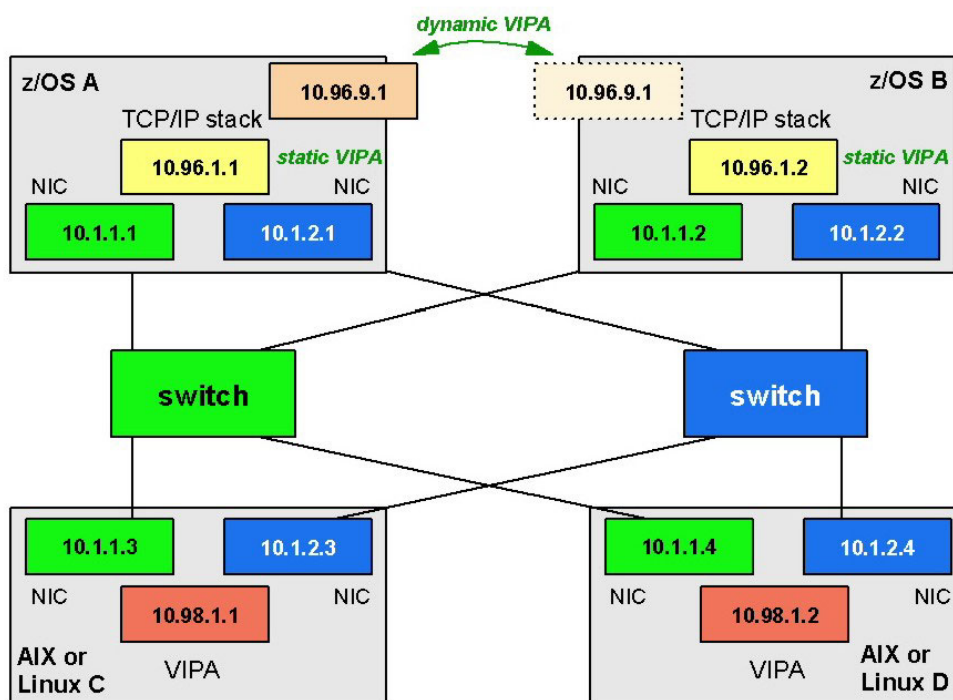


Figure 26. General concept of a fault-tolerant network

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 7, "Network considerations for high availability," on page 71 for details on how to set up a highly available network.

The following figures show how dynamic rerouting works. In Figure 27 on page 104 the virtual IP address virt_addr_1 on system A can be reached through IP addresses addr_1, addr_2 and addr_3. These real addresses are seen as gateways to the virtual IP address. ENQ and MSG indicate two applications running on that system. You can imagine that these are the SAP enqueue server and the message server.

HA SAP concepts

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

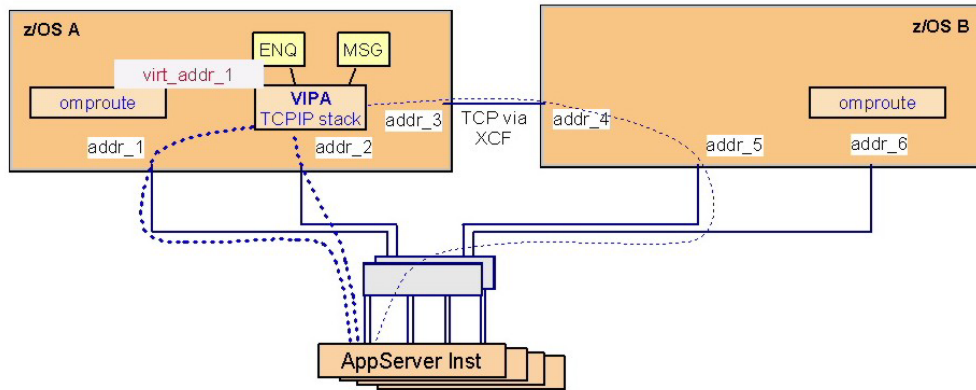


Figure 27. Alternative paths in a duplicated network

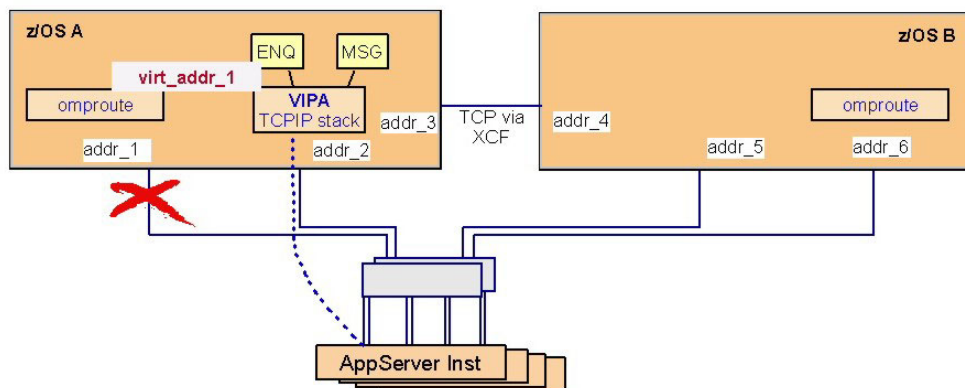


Figure 28. Rerouting if a network adapter card fails

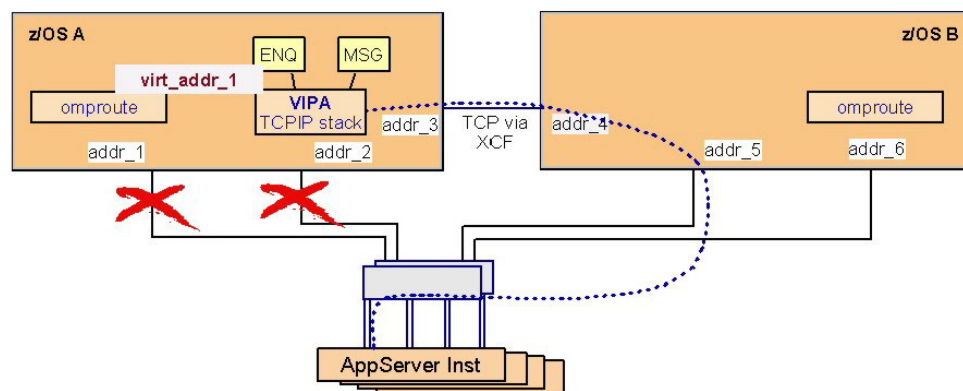


Figure 29. Rerouting in a sysplex even in case of two failing network cards

What happens if network adapter card `addr_1` fails? As shown in Figure 28 there is still a path from application server instances to system A. All TCP/IP traffic is now

routed through `addr_2`. The rerouting is absolutely transparent to the application. The router daemons on each system detect the missing links and propagate alternative routes. On z/OS, the router daemon is `omproute`.

What happens if network adapter card `addr_2` fails, too? As shown in Figure 29 on page 104, even then a path from application server instances to system A remains available. All TCP/IP traffic is now routed through system B via `addr_3`. Again, the rerouting is transparent to the applications.

Figure 29 on page 104 also shows that, as long as any system in the sysplex is reachable, all systems are reachable. However, what happens in case of a TCP/IP or LPAR failure? The automation software is able to detect such a failure, move `virt_addr_1` to system B, and restart the applications there. The takeover of the ENQ and MSG server together with the virtual IP address is shown in Figure 30. Now `addr_4`, `addr_5` and `addr_6` are propagated as gateways to `virt_addr_1`. The IP takeover to another system disrupts existing connections. Application server instances have to reconnect and resynchronize their communication.

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

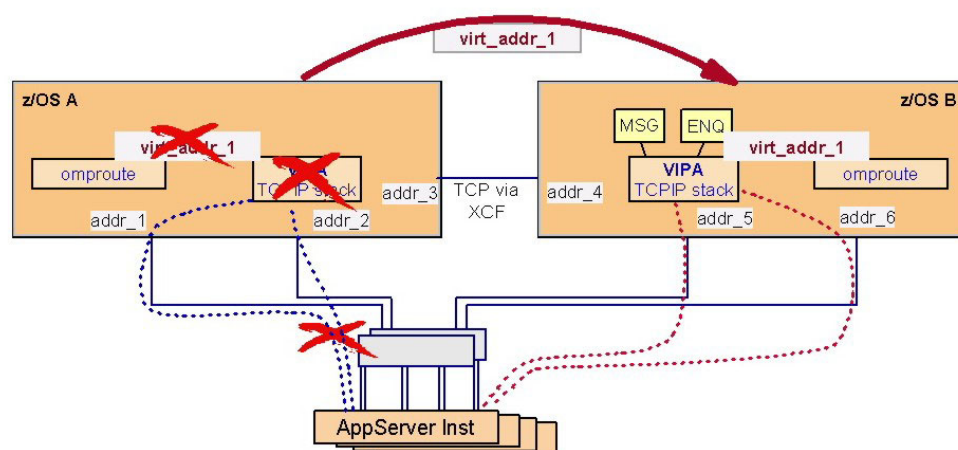


Figure 30. VIPA takeover and dynamic routing

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. In this configuration, a VIPA definition on the Linux system is not needed with respect to the database connection, though it could be useful for incoming connections from the LAN.

Static VIPAs are used 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:

HA SAP concepts

- NFS server
- ABAP and Java SCS
- SAP network interface router (SAProuter)

While the rerouting shown in Figure 27 on page 104 through Figure 29 on page 104 is applicable to both static and dynamic VIPAs, the takeover shown in Figure 30 on page 105 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 sharing is an option for other directories (for example, the directory containing the executables).

Shared directory access between z/OS systems is achieved with the Shared HFS feature. We highly recommend using zFS. zFS is the preferred file system and continued use of HFS is discouraged as HFS will become obsolete. New file systems should be created as zFS. See <http://www.redbooks.ibm.com/abstracts/TIPS0647.html> for further information.

1

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

In the case of UNIX 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.

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

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

Important

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

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

See also “Application design” on page 111.

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 physical file systems that are exported by the NFS server must also be accessible on the second system. This is another reason for using shared HFS.

The failover scenario is shown in Figure 31 and Figure 32 on page 108. Note that the NFS VIPA is different from the VIPA of SCS. So they can be handled independently of each other.

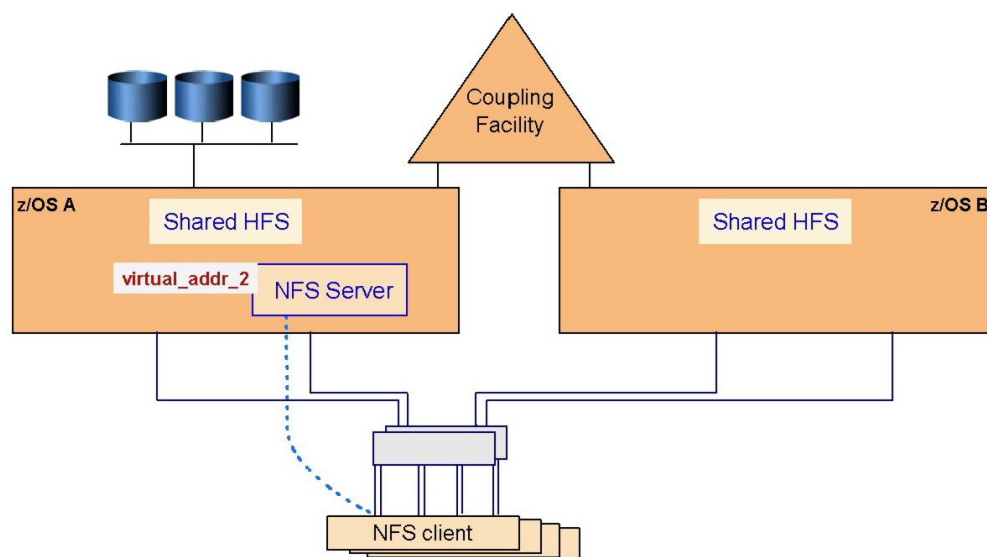


Figure 31. Initial NFS client/server configuration

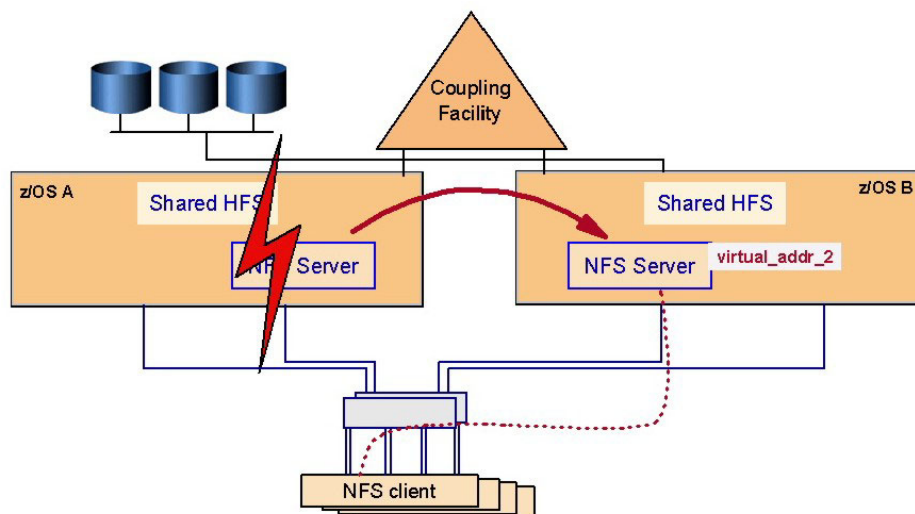


Figure 32. Failover of the NFS server

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 99.

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

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

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

Non-data-sharing

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

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

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

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 startups with less errors, thus providing improved overall system availability.

Data sharing

A data sharing configuration eliminates the database server as a single point of failure and provides for near continuous availability. In a data sharing configuration, planned outages can be avoided by using *DB2 connection failover* to move workload off the DB2 member needing the outage to an available DB2 member in the data sharing group. In the case of an unplanned outage, 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 stated above, automating the daily operations of starting, stopping, and monitoring all the components of the SAP system provides for improved SAP system availability by eliminating most human errors.

Remote application server and DB2 connection failover support

To give customers the ability to avoid planned and unplanned outages of the database server of SAP on DB2, SAP has always supported the use of DB2 Parallel Sysplex data sharing combined with DB2 connection failover. This removes the database server as a single point of failure.

General information

DB2 connection 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. The primary and one or more standby servers are configured by a profile that provides a list of database connections for each application server or group of application servers. Failover support for SAP application servers on z/OS enables the application server to switch over to a standby DBMS in the same LPAR.

When an SAP work process detects such a situation, it performs the redirection automatically after rolling back the current transaction. The SAP work process detecting this situation propagates this knowledge to all other work processes on the same SAP instance. If the standby server becomes inaccessible, its work processes are redirected to the next standby database server - which may well be the primary database server if it has become available again.

For a more detailed description of the SAP profile parameters that influence failover support, see the SAP online documentation *BC SAP High Availability* in the SAP Library or at

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

HA SAP concepts

In the navigation pane, open 'High Availability', then 'Media Library', and then 'HA Documentation'. See also the *SAP installation guides* and *SAP Database Administration Guide* for SAP basis release 6.40 (or higher). See also the section "Sysplex Failover and Connection Profile" in the respective SAP installation guide.

Redirection to a standby database server requires the use of DB2 data sharing. All primary and standby database servers must be members of the same data sharing group. SAP sysplex failover support is configured by a profile that provides a list of database connections for each application server or group of application servers.

All the standard recommendations for achieving high availability in DB2 data-sharing environments apply to the SAP system as well. For example, it is important to start a failed DB2 data sharing member on the same or another z/OS system as soon as possible in order to release the retained locks. Use Automatic Restart Management (ARM) (see *z/OS MVS Setting Up a Sysplex*) to restart a particular DB2 data sharing member quickly with minimal down time. As of DB2 V7, this restart for retained locks resolution can be accelerated by using the LIGHT option. When a DB2 data sharing member stops abnormally, the surviving z/OS systems determine if the corresponding z/OS system failed as well and restart the DB2 data sharing member appropriately on the same or a different system (see the DB2 manual *Data Sharing: Planning and Administration*).

For more information on high availability, see the SAP online documentation *BC SAP High Availability*, section "Replicated Database Servers".

Note:

In a data sharing environment, recovery from any failures of a database server, network, or gateways can be achieved with DB2 connection failover by switching over to a standby database server. For recovering from network and gateway failures, however, you have to provide the appropriate redundancies, such as duplicate LAN connections or ESCON® links.

Failover with multiple DB2 members in the same LPAR when using DB2 Connect

The following figure shows a configuration using DB2 Connect, with multiple DB2 members in one LPAR:

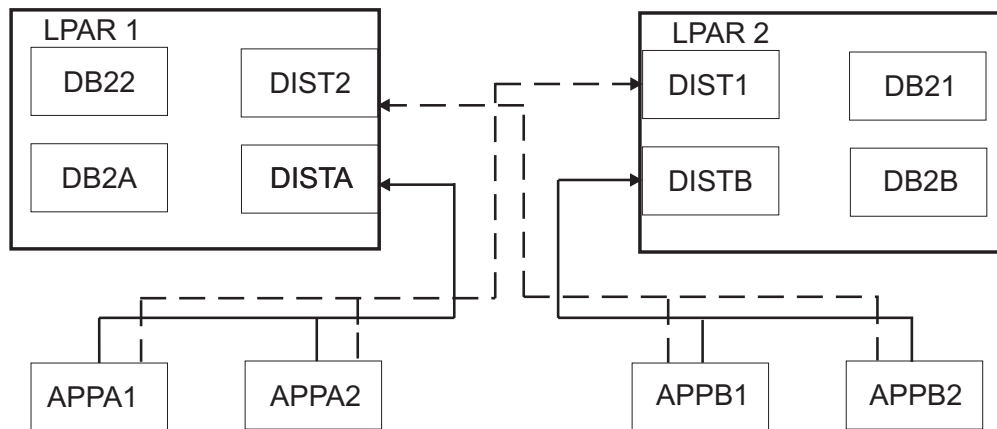


Figure 33. Failover setup using DB2 Connect, with multiple DB2 members in the same LPAR

DB2 requires that all members of a data sharing group use the same port number for their DDF address spaces. This means that the DB2 members of a data sharing group that reside in the same LPAR need to share the same port number. By using TCP server bind control, you can have two members from the same DB2 data sharing group in the same LPAR and still direct the connection of the application servers to whichever member you want. This is accomplished by using the 'BIND ipaddr' parameter on the PORT statement. When DB2 (ssidDIST) issues a bind to the port number in the PORT statement and to INADDR_ANY, the bind is restricted to the IP address specified on the PORT statement for that DB2 member. Then the application server selects the member via the IP address assigned to it.

You define for each member its own static VIPA, so there are 2 static VIPAs in each LPAR. Then you bind the member to its specific VIPA using the bind mechanism. The member-specific VIPA connects to the LPAR. This member specific VIPA must be used in the connect.ini file, as the SAPDBHOST in DEFAULT.PFL can only be set to one VIPA (normally the one which is given by SAPinst at DB installation).

Application design

The hardware, operating system, database, middleware, as well as the SAP components and applications, provide high availability features. Other applications or connectors to be used in a high availability environment should also be designed with high availability in mind.

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

- Make the applications restartable.

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

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

- Do not store vital data in files.

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

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

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

Failure scenarios and impact

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

When also running SA MP, it is possible to run the NFS server (file systems), SCS, application server instances, SAPOSCOL, and SAProuter under control of SA MP.

The scenarios discussed are those that are of most concern to customers. They are a subset of the scenarios discussed in “Verification procedures and failover scenarios” on page 249.

In the following tables, ‘SA’ indicates actions taken automatically and instantaneously by SA z/OS or SA MP, and ‘User’ indicates actions taken by the user. Also, for the action “User: Restart transactions” a customer could use workload scheduling software for this purpose (e.g., Tivoli Workload Scheduler).

The differences in impact between the configurations are marked in italics.

Old-style central instance without data sharing

In the scenario in Table 6, the SAP system is using the old style central instance and data sharing has not been implemented for the DB2 database server. Most customers without the need for high availability are currently using this configuration.

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

Table 6. Simple configuration

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

Table 6. Simple configuration (continued)

Failure	Impact	Actions
SAP gateway	<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
Syslog collector	<ul style="list-style-type: none"> For most transactions, no impact Global syslog file out of date 	SA. Restart syslog collector
SAProuter	<ul style="list-style-type: none"> User sessions lost Reconnect inhibited 	SA. Restart SAProuter User: Reconnect
NFS server	<ul style="list-style-type: none"> Some transactions stop, fail after timeout Batch transactions stop, fail after timeout Restart of application servers inhibited If data was written to file, last written data is in doubt 	SA. Restart NFS server User: Restart transactions
File system	<ul style="list-style-type: none"> Some transactions inhibited Batch transactions fail Restart of application servers inhibited If data was written to file, transaction is rolled back and last written data is in doubt 	User: Recover and remount the file system User: Restart transactions
Network (router, switch, adapter card)	<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
TCP/IP on central instance	Central instance fails (see failure of central instance)	SA. Restart TCP/IP SA. Restart central instance User: Restart transactions

Table 6. Simple configuration (continued)

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

Data sharing, DB2 connection failover, double network (single central instance)

The scenario in Table 7 builds on the previous scenario by adding DB2 data sharing, DB2 connection failover, shared HFS, and a double network with VIPA and OSPF. This scenario is still using 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.

Table 7. DB2 sysplex data sharing configuration with double network

Failure	Impact	Actions
DB2	<ul style="list-style-type: none"> Rollback of transactions Remote application servers fail over to other DB2 subsystems 	SA. Restart DB2 User: Restart transactions
DB2 Connect instance	<ul style="list-style-type: none"> Rollback of transactions 	The SAP application server automatically restarts the DB2 Connect instance.
Central instance	<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
Message server	<ul style="list-style-type: none"> Most transactions are inhibited because the enqueue work process is not reachable Application servers wait until message server is up again Group logon is inhibited 	SA. Restart message server User: Restart transactions

Table 7. DB2 sysplex data sharing configuration with double network (continued)

Failure	Impact	Actions
Application server instance	<ul style="list-style-type: none"> • Transactions on this instance are lost • Rollback of database updates • User sessions on this instance are lost 	User: Connect to another instance User: Restart transactions SA. Restart instance
SAP gateway	<ul style="list-style-type: none"> • For most transactions, no impact • Connections to registered RFC servers inhibited until they have reconnected to SAP gateway 	SA. Restart SAP gateway
Syslog collector	<ul style="list-style-type: none"> • For most transactions, no impact • Global syslog file out of date 	SA. Restart syslog collector
SAProuter	<ul style="list-style-type: none"> • User sessions lost • Reconnect inhibited 	SA. Restart SAProuter User: Reconnect
NFS server	<ul style="list-style-type: none"> • Some transactions stop, fail after timeout • Batch transactions stop, fail after timeout • Restart of application servers inhibited • If data was written to file, last written data is in doubt 	SA. Restart NFS server User: Restart transactions
File system	<ul style="list-style-type: none"> • For most transactions, no impact • If data was written to file, transaction is rolled back and last written data is in doubt 	User: Restart transaction
Network (router, switch, adapter card)	<i>None</i>	None
TCP/IP on central instance	Central instance fails (see failure of central instance)	SA. Restart TCP/IP SA. Restart central instance
TCP/IP on application server	Application server fails (see failure of application server)	SA. Restart TCP/IP SA. Restart application server instance User: Restart transactions
TCP/IP on database server	Connection to database server lost (see failure of DB2)	SA. Restart TCP/IP User: Restart transactions

Table 7. DB2 sysplex data sharing configuration with double network (continued)

Failure	Impact	Actions
z/OS LPAR	All components running in the LPAR fail (see failures of individual components)	User: Restart of LPAR SA. Restart DB2 SA. Restart other components

Enqueue replication and NFS failover: fully functional high availability

The scenario in Table 8 builds on the previous two scenarios by adding the SCS, the enqueue replication server, 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 9, “Preparing a high availability SAP solution,” on page 119.

Table 8. Fully implemented high availability solution for SAP

Failure	Impact	Actions
DB2	<ul style="list-style-type: none"> Rollback of transactions Remote application servers fail over to other DB2 subsystems 	SA. Restart DB2 User: Restart transactions
DB2 Connect instance	<ul style="list-style-type: none"> Rollback of transactions 	The SAP application server automatically restarts the DB2 Connect instance.
Enqueue server	None	SA. Failover enqueue server SA. Move enqueue replication server
Enqueue replication server	None	SA. Restart enqueue replication server
Message server	<ul style="list-style-type: none"> For most transactions, no impact Certain transactions inhibited (for example, SM66) Update/batch workload balancing inhibited Group logon inhibited 	SA. Restart message server
Application server instance	<ul style="list-style-type: none"> Transactions on this instance are lost Rollback of database updates User sessions on this instance are lost 	User: Connect to another instance User: Restart transactions SA. Restart instance

Table 8. Fully implemented high availability solution for SAP (continued)

Failure	Impact	Actions
SAP gateway	<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
Syslog collector	<ul style="list-style-type: none"> For most transactions, no impact Global syslog file out of date 	SA. Restart syslog collector
SAProuter	<ul style="list-style-type: none"> User sessions lost Reconnect inhibited 	SA. Restart SAProuter User: Reconnect
NFS server	<ul style="list-style-type: none"> None If data was written to file, last written data is in doubt 	SA. Restart NFS server
File system	<ul style="list-style-type: none"> For most transactions, no impact If data was written to file, transaction is rolled back and last written data is in doubt 	User: Restart transaction
Network (router, switch, adapter card)	None	None
TCP/IP on SCS	Enqueue server, message server, SAP gateway, syslog collector fail (see failures of individual components)	SA. Restart TCP/IP SA. Restart enqueue server, message server, SAP gateway, collector
TCP/IP on application server	Application server fails (see failure of application server)	SA. Restart TCP/IP SA. Restart application server instance User: Restart transactions
TCP/IP on database server	Connection to database server lost (see failure of DB2 server)	SA. Restart TCP/IP User: Restart transactions
z/OS LPAR	All components running in the LPAR fail (see failures of individual components)	User: Restart of LPAR SA. Restart DB2 SA. Restart other components

Chapter 9. Preparing a high availability SAP solution

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

The chapter includes the following sections:

- Software prerequisites
- Naming conventions
- DB2 setup
- File system setup
- Tivoli System Automation setup
- SAP installation aspects

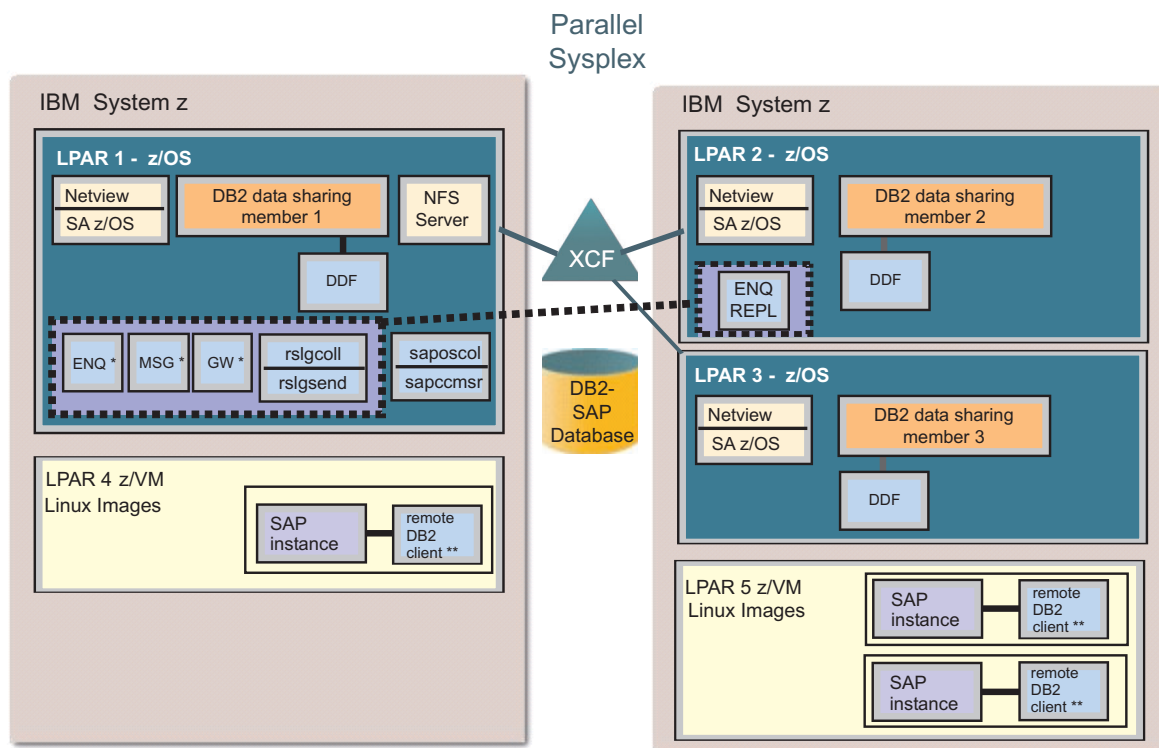
For networking considerations, see Part 3, “Network,” on page 69.

Sample high availability solution configuration for SAP

Figure 34 on page 120 shows a sample SYSPLEX configuration which demonstrates how SA z/OS and SA MP can be used to make all of the necessary SAP components highly available. The configuration includes:

- three LPARs running z/OS in a sysplex,
- a DB2 database with three data-sharing members,
- two LPARs with z/VM containing a total of three Linux guests.

Preparing a HA SAP solution



* optional IBM enhancements (for compatibility with older SAP HA policy versions)
 ** not modelled as SA resource

Figure 34. High availability solution configuration for SAP

Software prerequisites

Table 9 summarizes the software requirements. We provide 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 9. Software requirements for the HA solution

Product name	Minimum level requirement	Recommended level
z/OS	V1.8	V1.9 [1]
DB2 Universal Database for z/OS	Version 8	Version 9
Tivoli NetView for z/OS (required by Tivoli System Automation for z/OS)	V5.3	V5.3 [1]
System Automation for z/OS	V3.1 [1]	V3.2 [1]
SAP NetWeaver 2004s (7.00)	Minimum Support Package Stack 13 [1]	

Table 9. Software requirements for the HA solution (continued)

Product name	Minimum level requirement	Recommended level
[1] Or higher supported level		
For usage type Process Integration/Exchange Infrastructure (PI/XI) the high availability installation for clustered ASCS/SCS instance is only supported starting with NetWeaver 2004s, SPS8 (see SAP note 853510). SAP note 951910 describes configuration changes for SAP NetWeaver 2004s High Availability (HA) Usage Type PI.		

For SAP release previous to Netweaver 2004s (7.00), refer to *High Availability for SAP on IBM System z using Autonomic Computing Technologies*, SC-8206-01.

Application server instances were installed on zLinux on IBM System z. For details, see Table 10.

Table 10. SAP application server for Linux on System z

Product name	Minimum level requirement	Recommended level
Linux on System z	SUSE Linux Enterprise Server 10 for System z (64 bit), SP2	See SAP Note 81737.
Tivoli System Automation for Multiplatforms	3.1.0.0	3.1.0.0
Tivoli System Automation Application Manager	3.1.0.0	3.1.0.0
z/VM (optional)	V5.3	V5.3

Naming conventions

The following sections describe the conventions selected for naming of the policy components for SA z/OS and SA MP.

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 datasharing members.
- set of file systems.

Common questions that you might consider include:

- How do I monitor all SAP-related address spaces with SDSF?
- 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* will make it easier to monitor and maintain each SAP system. A SAP system setup for the high availability solution can also run on a server hosting

Preparing a HA SAP solution

other SAP systems. In this latter case, there are more components to consider when planning their names. You could, of course, define multiple HA SAP systems in one server or Parallel Sysplex.

When choosing 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 11) as a part of the component names belonging to that SAP system.

We recommend using **SAP** as a prefix for all SAP resources *not related* to a specific SAP system.

- In Table 11, we list the recommended names of all *z/OS-related components* of an SAP system, together with details of how or where they are defined.
- In Table 12 on page 123, we list the recommended names of all *components of an individual SAP system* that are defined within SA z/OS.

The names that are used in the “Sample” column correspond to a sample SAP system **HA1**, which is used in many examples throughout this book.

Table 11. Recommended names for all z/OS-related components of a SAP system

Component	Recommended name	Sample	How/where defined
DB2 address spaces	<SAPSID>xMSTR <SAPSID>xDBM1 <SAPSID>xIRLM <SAPSID>xDIST (where x defines the data sharing member)	HA1xMSTR HA1xDBM1 HA1xIRLM HA1xDIST (where x defines the data sharing member)	PROCLIB member names
High Level Qualifier for SAP VSAM objects	SAP<SAPSID>	SAPHA1	IDCAMS
High Level Qualifier for SAP zFS file systems: (A) SAPSID independent (B) SAPSID dependent	(A) SAPZFS.<basename(directory)> (B) SAPZFS.<SAPSID> <dirname(directory) tr "/" "."> (Also, see Notes below)	(A) /usr/sap/trans ==> DSN=SAPZFS.TRANS (B) /usr/sap/HA1 ==> DSN=SAPZFS.HA1.USR.SAP /sapmnt/HA1 ==> DSN=SAPZFS.HA1.SAPMNT (Also, see Notes below)	MOUNT FILESYSTEM command
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	ha1ascsv	
VIPA name for JAVA SCS	<sapsid>scsv	ha1scsv	TCP/IP DNS entry

Table 11. Recommended names for all z/OS-related components of a SAP system (continued)

Component	Recommended name	Sample	How/where defined
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

Notes:

1. This is to allow the z/OS Storage administrator to use the Dataset name prefix as the SMS selector for the SMS Group/Storage/Data CLASS(es).
2. The SAPZFS prefix could be used to point to an SMS pool and/or give a DATA CLASS that sets the correct zFS EXT attributes.
3. The second index level of <SAPSID> could allow a separate SMS POOL per SAP system (if required).

Table 12. Recommended names for all components of an individual SAP system

Component	Recommended name	Our name
Jobname for ABAP enqueue server	SAP<SAPSID>AE	SAPHA1AE
Jobname for ABAP enqueue replication server	SAP<SAPSID>ER	SAPHA1AER
Jobname for JAVA enqueue server	SAP<SAPSID>JE	SAPHA1JE
Jobname for JAVA enqueue replication server	SAP<SAPSID>JR	SAPHA1JR
Jobname for ABAP message server	SAP<SAPSID>AM	SAPHA1AM
Jobname for JAVA message server	SAP<SAPSID>JM	SAPHA1JM
Jobname for gateway	SAP<SAPSID>GW	SAPHA1GW
Jobname for syslog collector	SAP<SAPSID>CO	SAPHA1CO
Jobname for syslog sender	SAP<SAPSID>SE	SAPHA1SE
Jobname for sapccmsr (SAP system independent)	SAPSYSCR	SAPSYSCR
Jobname for saposcol (SAP system independent)	SAPOSCOL	SAPOSCOL
Jobname for saprouter (SAP system independent)	SAPROUTE	SAPROUTE
Jobnames for remote ABAP application server instances and their monitors:		
CI - ABAP instance containing SPOOL/BATCH/UPDATE	SAP<SAPSID>CI	SAPHA1CI
A<n> - ABAP dialog instances	SAP<SAPSID>A<n>	SAPHA1A1
Jobnames for remote JAVA application server instances and their monitors:		

Preparing a HA SAP solution

Table 12. Recommended names for all components of an individual SAP system (continued)

Component	Recommended name	Our name
JI - JAVA instance which includes the JAVA Software Deployment Manager (SDM)	SAP<SAPSID>JI	SAPHA1JI
J<n> - JAVA instances	SAP<SAPSID>J<n>	SAPHA1J1

Conventions used in the SA z/OS policy

The following table summarizes the naming conventions we used for the SA for z/OS policy described in “Adapting the SA z/OS best practices policy for SAP” on page 172:

Table 13. Naming conventions for SA z/OS resources

Type of resource	Naming convention
Resources related to SAP system <SAPSID>	SAP<SAPSID>*
Resources related to SAP in general which are independent of a specific SAP system	SAPSYS*
Groups with sysplex scope	*X
Jobnames for resources related to SAP system <SAPSID>	SAPHA1*
Jobnames for general SAP resources	SAP*

Tivoli System Automation for Multiplatforms

All resources (except those for application servers) that are generated by the mksap script (see Appendix E, “Sample Tivoli System Automation for Multiplatforms high availability policy for SAP,” on page 325) have the following naming conventions:

1. The first qualifier is defined by the value of the PREF variable of the configuration profile. Use a meaningful name such as 'SAP' or, for the resources of an SAP PI system, 'PI_ABAP' and 'PI_J2EE'.
2. The second qualifier is either SYS (for resources and groups existing only once in an SAP environment, such as the router) or the SAP system ID (SAPSID).
3. The third qualifier is a group name.
4. The fourth qualifier is the resource name.
5. All qualifiers are concatenated by an underscore ('_').

For example, the enqueue server of an SAP system with a PREF value of 'PI_ABAP' and an SAP system ID of HA1 is called 'PI_ABAP_HA1_ENQ_ES', and the IP address of the enqueue server is called 'PI_ABAP_HA1_ENQ_IP'.

The application server groups have the naming convention:

<PREF>_<SAPSID>_<nodename>_<instance_dir>

where:

- <SAPSID> is the SAP system ID
- <nodename> is the hostname of the node on which the AS is running
- <instance-dir> is the directory of the application server instance:
 - for ABAP and double-stack dialog instances, this is usually D<instance_number>.
 - for Java-only instances, this is usually J<instance_number>.

For example, the group for an application server of the SAP system with PREF value PI_ABAP and an SAP system ID of HA1 running on host p570sa04 with instance directory D00 is called

```
PI_ABAP_HA1_p570sa04_D00
```

The application server resources themselves are then constructed by adding _AS to the group name. Thus, the resource name for the above mentioned application server is

```
PI_ABAP_HA1_p570sa04_D00_AS
```

The name of a network equivalency for a service IP is the name of the group to which the service IP belongs, suffixed by '_NETIF'. For example, this results in <PREF>_<sapsid>_ENQ_NETIF

for the service IP of the enqueue group.

DB2

In a high availability environment, there is little sense in making SAP Central Services (SCS) highly available without making the database server highly available. We therefore strongly recommend the use of DB2 data sharing. Since 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 SA z/OS 'DB2 - Best Practise Policy' to perform a light restart

In case of an LPAR failure, the DB2 - Best Practise Policy is needed to allow recovery of the 'failed' DB2 subsystem on a different LPAR.

The DB2 - Best Practise Policy is set up according to the following requirements:

- "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 will:

1. Restart a DB2 data sharing member with a minimal storage footprint.
2. Terminate the DB2 member normally after freeing any retained locks that were held by this member.

Note: To use the DB2 - Best Practise Policy you must have installed and applied APAR OA26776.

File system setup

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

The Shared HFS feature allows you to define shared as well as system-specific file systems, by using special variables in the path name. If you have all your SAP systems within one sysplex, you can share all the files. If you, for example, have one production sysplex and one test sysplex, and still want to use the same file systems (for example, the Transport directory), you must use the NFS Server/Client feature. NFS Server must run on the system that owns the directory, and NFS Client must run on the other system.

File systems

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

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

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

```
/sapmnt/HA1/linux_s390x/exe
```

Figure 35 shows the SAP directory structure and file systems for the ABAP SCS. This is similar to the classic ABAP central instance except that the instance name is different.

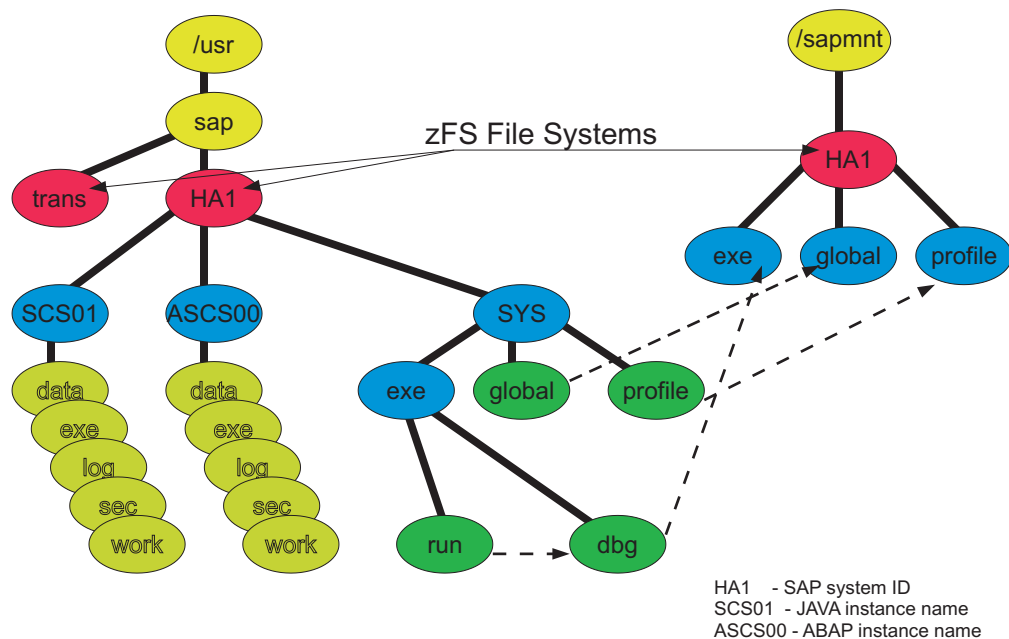


Figure 35. SAP directory structure and file systems

Setting up zFS filesystems

IBM recommends that you use the zFS *USS Filesystem* type rather than the older *HFS* type. Throughout this manual, this recommendation has been included. For High Availability, we also recommend that:

- zFS's be allocated from a DFS SMS DASD POOL.
- the SMS DATACLASS associated with all zFS's be 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. This can be verified using the following zFS command:

```
zfsadm configquery -aggrgrow
```

Providing sufficient volumes and space are available within the associated DFS SMS DASD POOL, the above recommendations allow the zFS's to expand "on-demand".

Once the SMS customization is complete, the creation of zFS Filesystems is relatively simple. 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 5GB, and secondary size of 1GB. **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 filesystem 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 mountpoint `/sapmnt`.

Notes:

1. Permanent mounting of any USS Filesystems should be done through `BPXPRMnn` members of `SYS1.PARMLIB`.
2. The `AUTOMOVE=YES MOUNT` option is required for High Availability in the event of an LPAR outage.
3. Correct SPACE requirements can be determined from the SAP installation manuals, and remember that you may require additional space for backups especially when adding new maintenance.
4. Further information on the usage of the zFS Filesystem type can be found in the IBM manual: *z/OS Distributed File Service zSeries File System Administration*.

SAP directory definitions

The following directories must be defined:

SAP global transport directory

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

SAP system-wide directories

The subdirectories of `/usr/sap/<SAPSID>/SYS` are usually defined at installation time as symbolic links to the corresponding subdirectories of `/sapmnt/<SAPSID>`, for example, `/usr/sap/HA1/SYS/profile` points to `/sapmnt/HA1/profile`. The directory `/sapmnt` is to be created in the root file system and thereby shared in the sysplex.

The directory `/sapmnt/HA1` is the mount point for the SAP system-wide file system. This file system needs to be exported by the NFS server such that it can be mounted by remote application server instances.

SAP local directories

On z/OS the directory `/usr` is a symbolic link to `$VERSION/usr`. This means that the contents of the `/usr` directory is different on every LPAR. This, however, is not

Preparing a HA SAP solution

practical for the /usr/sap directory. We propose to create the directory /sap in the root file system and to define symbolic links for /usr/sap to point to /sap. The symbolic links must be defined on each LPAR, i.e. in each \$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 points /usr/sap/<SAPSID> for the instance-specific file systems, such as ASCS00. These file systems do not need to be exported by NFS.

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

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

SAP administrator's home directory

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

SAPOSCOL/SAPCCMSR directory

The SAP executables that enable SAP monitoring are SAP system independent and should therefore be placed in a directory of its own, for example /usr/sap/saposcol. This directory needs to be shared across all members of the sysplex.

NFS server on z/OS

The z/OS NFS server used by SAP HA NFS clients must be moveable between z/OS LPARs within the same SYSPLEX (see "Mount handle databases and the remount site attribute" on page 130 for more information).

Note that the associated dynamic VIPA is only moveable within the same TCPIP subplex which typically consists of the whole SYSPLEX, but be aware that multiple TCPIP subplexes can exist within a single SYSPLEX. For the SAP z/OS UNIX filesystems to be accessible from multiple z/OS LPARs within a SYSPLEX the z/OS UNIX System Services (USS) PARMLIB member BPXPRMxx must contain the keyword SYSPLEX(YES).

How many NFS servers should I run?

Our recommendation is one server running with the NFS server security(exports) security model, as this will greatly simplify the setup and future diagnostics. However, each customer's existing environment may already have the complication of having or requiring multiple NFS servers.

If you run multiple SAP systems you may wish to run an NFS server per SAP system for reasons of NFS server security model (see next section) or Service Level Agreement (SLA). Providing that the NFS servers never run on the same z/OS LPAR then this should not be a problem. If however you do require to run multiple NFS servers on the same z/OS LPAR, then be aware that multiple TCPIP stacks must be utilized which is not recommended and is beyond the scope of this document.

If you wish to utilize an existing NFS sever for both existing applications and for SAP HA, then please consider that for SAP HA the z/OS NFS server must be movable between z/OS hosts within the same 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 and dynamic routing (usually OSPF). However, if you have other existing NFS clients using a fixed IP address for the NFS server's z/OS host, when the NFS server moves to another z/OS host those clients will "hang" until the NFS server is restarted on the same z/OS host. To avoid this situation, you must either:

- Reconfigure your existing NFS clients to also utilise the z/OS Dynamic VIPA and dynamic routing (on the client or on a gateway router), or
- Give preference to the movement of the NFS server back to the expected z/OS host.

Which NFS server security model (exports, safexp, or saf) should I use?

If you have an existing NFS Server that does not use the security(exports) NFS security model, then use of the security(safexp) is possible but more complex initially. The use of the security(saf) security model can not work in an HA environment because of the need for each NFS client to manually issue an mvslogin command per user.

Security(exports)

We recommend using security(exports) because this is the default UNIX NFS server mode of operation. It does not require any RACF definitions, or the use of the mvslogin command. Transparent access to NFS filesystems even through NFS server failures and restarts across different z/OS hosts is possible.

You may have concerns about the attribute security(exports). This attribute means that normal UNIX security applies. First of all, the export list of the movable NFS server can be limited to the mentioned global SAP directories, which do not contain sensitive data. Furthermore, the access can be restricted to specific client IP addresses. For further information on setting up NFS, see *Network File System Customization and Operation, SC26-7417-07*.

security(safexp)

As its name implies, this is a combination of both security(exports) and security(saf). When carefully configured dual NFS server security model usage is possible.

If you have an existing NFS Server that does not use the security(exports) NFS security model, then use of the security(safexp) is possible but more complex initially. To use security(safexp) requires that the SAP HA z/OS UNIX filesystems be added to a checklist file and that the site attribute checklist be added to the NFS server's attribute file.

However, note that the SAP installation program SAPinst must be run as the root user and must have root access to the SAP filesystems. For security(safexp) this can only be achieved by using the mvslogin command, so the adding the filesystems to the checklist must be done after SAP installation is complete.

security(saf)

This is not an option for SAP HA because of the need to use the mvslogin command. Automation of the mvslogin command on every client before access of the NFS filesystems at startup and recovery times is impractical.

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. These are the VSAM data sets specified as FHDBASE and FHDBASE2. The reason is that at mount time the NFS server stores the mount handles in these data sets to preserve them for restart or failover. If the NFS client loses a TCP/IP connection to an NFS server, it simply reconnects; the NFS protocol expects that the mount handles are still valid.

If the physical z/OS UNIX filesystem is remounted, the old mount handle becomes invalid. This is the default behaviour. However, the remount site attribute enables the NFS server to process NFS requests after the NFS server is restarted, even though the z/OS UNIX file system was remounted with a new z/OS UNIX file system number after its last usage. Use of the remount attribute causes the NFS server to automatically access a remounted z/OS UNIX file system. However, it cannot be assured that the file system has not been changed prior to remounting

The.nlm site attribute

You must ensure that the NFS Lock Manager (NLM) is started on the z/OS NFS Server. Many applications and tools will expect to use file locks and, unlike most other implementations, the z/OS NFS Server defaults to not running an NLM. To have the z/OS NLM start with the NFS Server simply add the following single word statement to the NFS Attributes file:

```
nlm
```

NFS client root access

This depends on the NFS server security model:

- security(saf) - this is unsuitable for SAP HA
- security(safexp) - use the mvslogin command and a RACF Userid that has uid=0
- security(exports) - in order to allow root (uid=0) access by NFS clients to the files exported by the z/OS NFS Server the files listed in the exports file must specify the NFS clients (by name or IP) and the new 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/write root access to the SAP profile subdirectory, you would enter:

```
/hfs/sapmnt/HA1/profile -rw=10.101.4.214<root>|\
                        10.101.4.215<root>|\
                        10.101.4.216<root>
```

Note: A normal SAP installation using SAPinst requires root access to any NFS mounted SAP filesystems.

NFS Server automation

If you plan to run the NFS server for your SAP system under Linux on System z, you need to make that NFS server highly available. This can be done via Tivoli System Automation for Multiplatforms. To get a highly available NFS server, you can set up and apply the NFS server HA policy, which is pre-configured by Tivoli System Automation for Multiplatforms. See “Making NFS highly available via SA MP” on page 208 for further information.

NFS Clients - General Information

For NFS Clients that connect to the z/OS NFS Server it is recommended storing all data in EBCDIC code page 1047. This will allow z/OS tools and applications that do not honor USS codepages to utilize any data created by the SAP Application Server.

Any mounts be they manual, via `/etc/fstab`, or an automount daemon should set the following server side attributes for the SAP profile, global, and trans directories:

```
TEXT,cln_ccsid(819),srv_ccsid(1047)
```

The above will cause text translation to occur using a codepage of ISO8859-1 for that Client and IBM-1047 for the Server.

All mounts should use the name or IP address of the Dynamic VIPA associated with the z/OS NFS Server. This will allow mounts and remounts to function correctly if the z/OS NFS Server is moved from one LPAR to another.

NFS Clients should ensure that they have a local Source VIPA set so that all IP packets sent to the z/OS NFS Server will 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).

All NFS Clients should also ensure that any user name and group name used to access NFS files are associated with exactly the same ones by uid and gid on the z/OS NFS Server.

NFS Client on Linux on System z

It is recommended to use the automount daemon to mount NFS filesystems on-demand.

Assuming that the z/OS Server exports the following filesystems:

```
/HFS/sapmnt/HA1/exe
/HFS/sapmnt/HA1/linux_s390x/exe
/HFS/sapmnt/HA1/profile
/HFS/sapmnt/HA1/global
/HFS/sapmnt/HA1/trans
```

These can be automounted by creating the following files in `/etc`:

```
auto.master
/sapmnt/HA1 auto.ha1.sapmnt
```

The `auto.ha1.sapmnt` contains the following:

```
exe          -rw,hard  sapnfsv:/hfs/sapmnt/HA1/linux_s390x/exe
profile      -rw,hard  sapnfsv:/hfs/sapmnt/HA1/profile,TEXT,cln_ccsid(819),srv_ccsid(1047)
global       -rw,hard  sapnfsv:/hfs/sapmnt/HA1/global,TEXT,cln_ccsid(819),srv_ccsid(1047)
trans        -rw,hard  sapnfsv:/hfs/sapmnt/HA1/trans,TEXT,cln_ccsid(819),srv_ccsid(1047)
```

The automount daemon will take total control of the directory specified in the `auto.master` file. In the examples above that means that the automount daemon will control: `/sapmnt/HA1`.

Only mountpoints should exist in this directory, and then only when the automount daemon is running. If the automount daemon is inactive then this directory should be empty.

To allow listing of the directory without causing mounts to occur it is recommended that you specify the `--ghost` automount daemon option. On SLES-10 this can be done by adding the following statement to the `/etc/sysconfig/autofs` file:

```
AUTOFS_OPTIONS="--ghost"
```

Tivoli System Automation

Setup of Tivoli NetView and Tivoli System Automation for z/OS

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

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)
- RRS
- SA Automation Manager
- TCP/IP
- (For further base z/OS resources, refer to the *SA z/OS Best Practise Base Policy*).

If you use the Automated Restart Manager (ARM), your configuration needs to be checked to ensure that it does not interfere with Tivoli System Automation.

The Status Display Facility (SDF) function of SA z/OS is useful for moving the SAP components between the LPARs. If you want to use SDF, define an SDF focal point and perhaps an SDF backup focal point on your systems. Of course, if you have the NetView Management Console (NMC) installed, you can use it instead of SDF.

Make sure that SA z/OS starts all applications and puts them into a “green” status.

Tivoli System Automation for Multiplatforms setup

Installation of SA MP is described in Chapter 12, “Customizing the Tivoli System Automation for Multiplatforms (Base),” on page 195 and Chapter 13, “Customizing the Tivoli System Automation Application Manager (E2E),” on page 227.

Control of remote ABAP application server instances

ABAP application server instances under AIX or Linux can be controlled in two ways:

1. IBM strategic and highly recommended: Use TSA MP together with end-to-end automation management TSA AM to control remote ABAP application servers running on platforms supported by SA MP. This requires the installation and setup of TSA MP and TSA AM.
2. Tactical solution using ssh/rsh: We created five shell scripts (see Appendix D, “Description of the z/OS high availability scripts,” on page 319) that allow you to start, stop and check local and remote ABAP application server instances.

SAP installation aspects

When installing an SAP instance, you will be prompted for the hostname. Specify the hostname associated with the static VIPA of the DB2 member of the z/OS LPAR.

SAP license

For normal SAP installations, you must obtain an SAP license for the LPAR where the message server runs. 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.

For further details, see SAP Note 538081 (“SAP Notes” on page 349) and “ABAP SAP Central Services (ASCS)” on page 157.

SAP logon groups

We recommend that you define LOGON groups. LOGON groups are used to automatically distribute user logons to individual instances (application servers) or to groups of SAP instances. They are also useful for reconnecting to another SAP instance in case the SAP GUI connection or the instance itself becomes unavailable.

Note: You must use the virtual hostname of the ABAP SCS as the message server hostname if you run ABAP-only or double-stack application servers. If you created a logon group before you switched from the CI to ASCS, you must adapt the message server hostname in the logon group definition. Also, if you are running a highly available SAProuter with its own virtual hostname, adapt the routing string in the logon group definition as well.

Preparing a HA SAP solution

Chapter 10. Customizing SAP for high availability

In this chapter, we describe what you need to do to setup SAP in order to run as a highly availability SAP solution.

The chapter covers the following:

- How to configure the ABAP and/or Java variants of SAP Central Services. Each SCS variant comprises the so-called standalone enqueue server
- How to configure the SAP environment for Tivoli System Automation

You can also find detailed SAP documentation for installing and using the ABAP standalone enqueue server (ABAP SAP Central Services) under:

http://help.sap.com/saphelp_nw04s/helpdata/en/36/67973c3f5aff39e1000000a114084/content.htm

Alternatively, you can go to

<http://help.sap.com>

and search in the documentation for 'standalone enqueue server'.

This chapter contains these main topics:

- "Prerequisites"
- "Setting up an ABAP SCS instance and/or a Java SCS instance" on page 136
- "Preparing SAP on z/OS for automation" on page 157

Prerequisites

ABAP SAP Central Services (ASCS) can run under z/OS UNIX System Services or Linux on System z.

- For high availability reasons, we recommend running the ASCS on z/OS. This recommendation holds for ABAP-only systems as well as for a double-stack system.
- For a double-stack system, we strongly recommend running both the ABAP SCS and Java SCS on z/OS.
- Preferably, you should also allocate the file systems needed by SAP on z/OS.

It is also possible to run ASCS on:

- Linux on System z,
- AIX,
- Linux on System x.

For such a heterogeneous environment, we strongly recommend running the SA MP end-to-end management automation component.

Java SAP Central Services can run under z/OS UNIX System Services or Linux on System z. For high availability reasons, we recommend running the Java SCS on z/OS.

It is also possible to run Java SCS on:

- Linux on System z,
- AIX,
- Linux on System x.

Customizing SAP for HA

For such a heterogeneous environment, we strongly recommend running the SAP end-to-end management automation component.

This chapter makes use of the `lsof` utility to list currently available ports. The `lsof` utility comes standard with Linux, but not for AIX. You can download the utility from <http://www-03.ibm.com/systems/p/os/aix/linux/toolbox/rpmsgroups.html>.

Setting up an ABAP SCS instance and/or a Java SCS instance

The following information applies to SAPinst for NetWeaver 04 SR1 and above.

Before you can start installing the SAP Central Services under z/OS UNIX system services (USS) with SAP installer tool SAPinst you must complete the following prerequisites:

1. Setup a highly available network.

Note: If this is not possible or you want to work in parallel on HA network and HA SAP then at least the hostnames of the VIPAs listed below must be defined. 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 and their corresponding virtual hostnames.

You need:

- 1 dynamic VIPA for the highly available z/OS NFS Server
- 1 dynamic VIPA for ASCS instance
- 1 dynamic VIPA for SCS instance
- n static VIPAs, where n is the number of LPARs which run a DB2 data sharing member for your SAP system

We suggest that you use the following names for a SAP system with a SAPSID of HA1:

Used for	Static or dynamic VIPA	Sample virtual hostname	IP address
Highly Available z/OS NFS Server	dynamic VIPA	sapnfs (or sapnfsv),	
ASCS instance	dynamic VIPA	ha1ascsv	
SCS instance	dynamic VIPA	ha1scsv	
(LPAR of) DataSharing Member 1	Static VIPA	coh1vipa (coh1 is LPAR name)	
(LPAR of) DataSharing Member 2	Static VIPA	coh2vipa (coh2 is LPAR name)	
(LPAR of) DataSharing Member n	Static VIPA	coh<n>vipa (coh<n> is LPAR name)	

This is a minimum number. If you finally decide to run SAP Web Dispatcher and/or SAP Router under USS you need another dynamic VIPA for each.

You can activate a dynamic VIPA via a command from UNIX System services. For example if 10.101.5.194 is the IP address of ha1ascsv, then the following command will activate it:

```
moddvipa -p tcpip -c 10.101.5.194
```

3. Setup a HA NFS server as described in “Making NFS highly available via SA MP” on page 208
4. Define group and userids and put <sapsid>adm into user alias table, according to the SAP Security Guide: IBM DB2 for z/OS, and the SAP Planning Guide.
5. Setup WLM definitions according to SAP Planning Guide

The dynamic VIPAs and their corresponding virtual hostnames are necessary in order to move applications (here NFS server and ASCS/SCS 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. You should install the ABAP Central Services and the Java Central Services directly with SAPinst using for each instance an own virtual hostname.

Rationale for enhancing the standard ASCS with additional SAP services

The standard SAP HA installation for ABAP SAP Central Service (ASCS) comprises the standalone enqueue server and the message server. There are two more important SAP services which are recommended to be made highly available by adding them to the ASCS instance:

- The central syslog collector service (CO)
- The gateway service (GW)

The CO service is normally running in the Central Instance. A GW service is normally running in the CI and in each Dialog Instance. The drawback of this design is that the CO service and the CI’s GW service are not available if the CI is down. To make them highly available, we recommend moving the CO service to the ASCS instance and to add a GW to it. That way, the CO and GW are always up as, in the case of a planned or an unplanned node outage, the ASCS fails over to a backup node. Together with the ASCS services, its virtual hostname is moved. Therefore after a failover the CO and GW are restarted and can be reached through the virtual ASCS hostname.

Additionally we recommend adding to the ASCS instance also a standard syslog sender which allows the ASCS instance to log entries into the central syslog.

Note: For these HA enhancements, you must make manual changes to the instance and start profiles of your SAP system as they are not included in the standard SAP HA installation setup. It does, however, mean adding complexity and introducing the potential for making errors. If you do not need these services highly available, then stick to the standard SAP HA installation.

In the sections below, the HA enhancements are optional.

Rationale for not running the Enqueue Replication server as an ERS instance

The standard SAP HA installation for ABAP SAP Central Service (ASCS) or Java SAP Central Service (SCS) does not support installing the Enqueue Replication Server (ERS) as its own instance. A description of how to setup such an ERS instance manually can be found on the SAP web site at http://help.sap.com/erp2005_ehp_03/helpdata/EN/47/8eb99350972b4a8c087b981c88430b/frameset.htm.

As long as SAP installer does not support such an ERS instance installation, we do not recommend setting up ERS as own instance.

Customizing SAP for HA

The following examples assume that the ERS is started by starting the enrepsvr executable using the ASCS or SCS instance profile.

In order to easily start the Enqueue Replication Server manually, we recommend putting a small script called StartABAP_EnqueueReplicationServer into the home directory of the <sapsid>adm.

In the example below, the SAPSID is HA1:

```
#-----  
# Start SAP ABAP enqueue replication service  
#-----  
/bin/rm -f er.sapHA1_ASCS00  
  
/bin/ln -s -f /usr/sap/HA1/ASCS00/exe/enrepsvr er.sapHA1_ASCS00  
  
er.sapHA1_ASCS00 pf=/usr/sap/HA1/SYS/profile/HA1_ASCS00_halascsv  
&exit
```

Setting up a standalone ERS instance requires having another additional virtual hostname. So, if you choose to run a double-stack SAP system, you need two more virtual hostnames which are used to make the ERS instance profile name independent from real hostnames. As the ERS is not really accepting 'client' requests under the virtual hostname, the IP is not used at all.

If you want to set up your ERS as a standalone instance, the steps are as follows:

1. You must add the following parameter to the ABAP ERS instance profile:
enque/encni/repl_port = xxxx
where xxxx is the port number defined in the (A)SCS instance profile for the same parameter.
This ensures that the ABAP Enq. replication server knows the port on which the standalone ABAP Enq. server is listening (valid according to SAP up to SAP kernel release 7.00).
2. Activate and replace the sample values with the values of your installation in the entries for ERS in the ABAP/J2EE_instances.conf files

General installation sequence

You install an AddIn-SAP system in following sequence:

1. Install ASCS (the virtual ascs host must be active)
2. Install SCS (the virtual scs host must be active)
3. Install DB
4. Install CI
5. Install at least one or more DI

This is also the start sequence of your SAP system. To start the SAP system manually you would:

1. Start ASCS (the virtual ASCS host must be active)
2. Start ASCS ERS on different system
3. Start SCS (the virtual SCS host must be active)
4. Start SCS ERS on different system
5. Start DB
6. Start CI
7. Start DI(s)

Installing and configuring ABAP SAP Central Services (ASCS)

Note: The information in this section does not apply to a Java only environment.

In order to move the ASCS from one cluster node to another within a cluster it requires its own virtual host name. Therefore, you must first define a virtual host name and IP address. Under z/OS use a dynamic VIPA.

SAP NetWeaver 04 SR1 includes ASCS installation support, as do later SAP releases. We strongly recommend installing the ASCS with SAPinst using the SAPinst option `SAPINST_USE_HOSTNAME=<virtual hostname>`.

After you have used SAPinst to install an ASCS with virtual hostname (as in “Option 1: Installing ASCS with a virtual hostname”), there is only one subsequent manual action that is required:

- change SAP profiles to switch on Enqueue table replication.

Optionally, change the SAP profiles for additional highly available services:

- SAP gateway
- syslog collector
- syslog sender.

The following section describes describe three options for creating a running ASCS instance using a virtual hostname.

- Option 1: Installing ASCS with a virtual hostname

This is the preferred installation option.

Use SAPINST option `SAPINST_USE_HOSTNAME`, for example:

```
./sapinst SAPINST_USE_HOSTNAME =<virtual ASCS host>
```

- Option 2: Installing ASCS with a physical hostname

You must do this before setting up the HA environment.

You must manually adapt the SAP profiles and environment to switch to a virtual hostname.

- Option 3: Installing SAP with classic CI

You must do this before setting up the HA environment. You must manually create a ASCS instance from the existing CI and manually adapt the SAP profiles and environment to use a virtual hostname.

If you choose option 1, only one manual profile change is necessary to switch on Enqueue table replication.

Option 1: Installing ASCS with a virtual hostname

Before starting the installation under USS, ensure that:

1. Java is installed.
2. Add or check the settings of the following variables in `.profile/.logon` of your installation user (UID 0). For example:

```
export SAPINST_JRE_HOME="/usr/lpp/java/a40/J1.4"
export JAVA_HOME="/usr/lpp/java/a40/J1.4"
export PATH="$SAPINST_JRE_HOME"/bin:"$PATH"
export _BPXK_AUTOCVT=ON**
export _TAG_REDIR_IN=TXT
export _TAG_REDIR_OUT=TXT
export _TAG_REDIR_ERR=TXT
export DISPLAY=127.0.0.1:2
export TZ=MEZ-1MES,M3.5.0,M10.5.0
export _BPX_SHAREAS=NO
```

Customizing SAP for HA

** Only required, if you do not have 'Enable Enhanced ASCII (Auto Conversion)' enabled at the LPAR level. See SAP Note 858969 and '5.2 Post Installation for HA system on USS' for details.

Note: If you have problems with SAPinst, you can set the environment variable `_EDC_ADD_ERRN02` to 1 in order to generate `errno2` which displays the reason code.

Install the ABAP Central Services directly with SAPinst using an own virtual hostname. For example, for a SAP system with `<SAPSID> HA1` and to install the ASCS with virtual host `ha1ascsv`, run from the install directory of the SAP DVD:

```
./sapinst SAPINST_USE_HOSTNAME=ha1ascsv
```

Use the High-Availability System menu option within SAPinst:

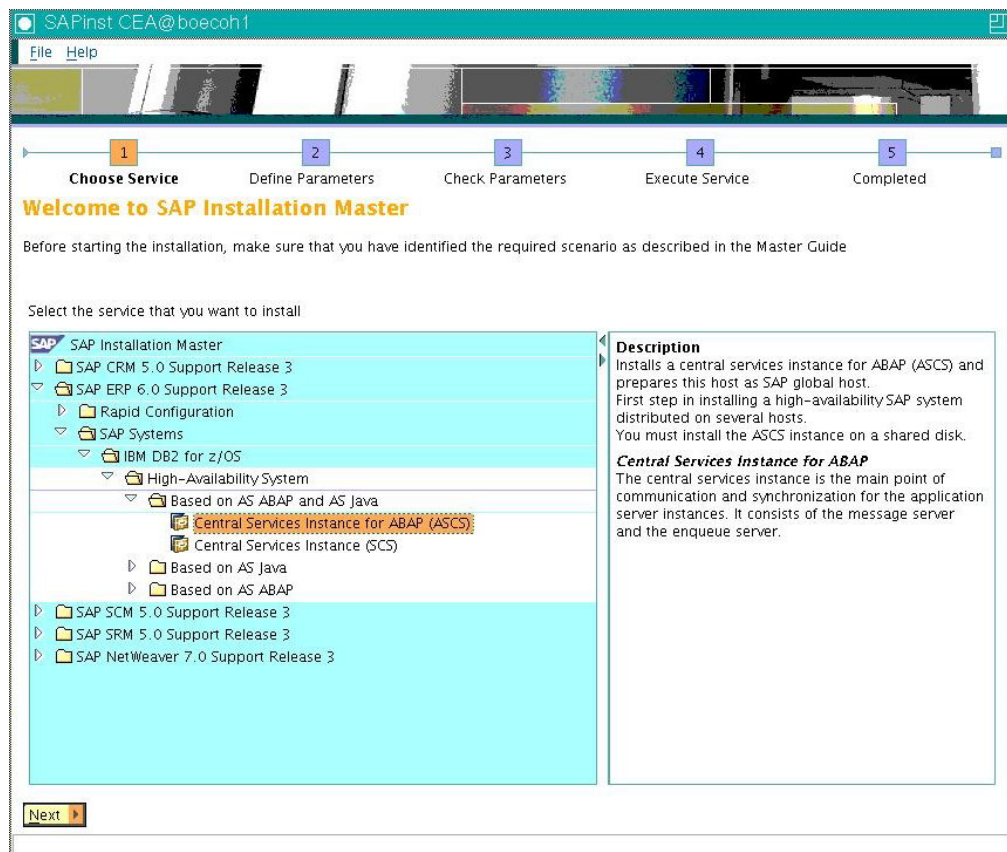


Figure 36. High-Availability System option within SAPinst

Above is an example for an ERP 2006 SR3 installation on SAP on System z running with DB2 z/OS and installing the ABAP Central Services Instance with the virtual hostname `ha1ascsv`. Select the **High-Availability System** option according to your DB and OS platform.

Stop SAPinst after ASCS installation and start it again for SCS installation with:

```
./sapinst SAPINST_USE_HOSTNAME=ha1ascsv
```

That way, the Java SCS is installed with its own virtual host `ha1ascsv`.

In addition, you should:

- Switch on **Enqueue table replication**
- Optionally run a highly available gateway, the syslog collector and the syslog sender service.

We strongly recommend saving a copy of the original profiles before changing them.

Setting up Enqueue table replication

The examples in the section below show an SAP system with a <SAPSID> of AX0 and instance number 00.

Perform the following steps to enable enqueue table replication:

1. Add the following to the ASCS instance profile:

```
enqueue/server/replication = true
enqueue/encni/repl_port = 6500
```

We recommend using 6500 plus 100 * ASCS instance number. Ensure that the chosen port number for replication is not used by another application. To check whether the port number is in use, enter the command `lsof -i -P` under Linux or AIX to list currently used ports.

If you add `-n` to the `lsof` command, it will run much faster because DNS lookups will not be done for *every* hostname. Piping into a `grep` will find the port. For example, to find port 5539 you would enter:

```
lsof -i -n -P | grep ':5539'
```

If replication is set to true but the `repl_port` is not defined, the port of the gateway service will be used (3300 plus 100 * ASCS instance number). In this case, either the gateway service or the replication will fail, depending on which service can bind to the port.

2. Add the following two lines to each Application server instance profile:

```
#-----
# VS 15thApr08: to adapt enqueue replication settings
#-----
enqueue/con_retries = 120
enqueue/deque_wait_answer = TRUE
```

SAP installer creates the instance directories on the installation node. You must create the directories where the ASCS instance and the Enqueue Replication server can run locally on each of the other nodes. That is the easiest way to have them available on all cluster nodes where the ASCS instance can run.

3. If your `/etc/services` is different for each LPAR/host, then 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, then the following entries are needed:

```
sapmsAX0 3600/tcp # SAP System Message Server Port
sapdp00 3200/tcp
sapdp00s 4700/tcp # SAP System Dispatcher Security Port
```

The following steps are required for Linux and AIX but are not necessary for USS

4. As `<sapsid>adm`, on the installation node enter the following commands:

```
cd /usr/sap/AX0
tar -cvf ASCS00.tar ASCS00
scp ASCS00.tar ax0adm@<other cluster node>:/usr/sap/AX0
```

Customizing SAP for HA

5. As <sapsid>adm, on each other node where the ASCS can potentially run, enter the following commands: :

```
cd /usr/sap/AX0
tar -xvf ASCS00.tar ASCS00
rm ASCS00.tar
```

In order to easily start the Enqueue Replication Server manually, we recommend putting a small script called StartABAP_EnqueueReplicationServer into the home directory of the <sapsid>adm. An example of the script is given below:

```
#-----
# Start SAP ABAP enqueue replication service
#-----
/bin/rm -f er.sapAX0_ASCS00
/bin/ln -s -f /usr/sap/AX0/ASCS00/exe/enrepserver er.sapAX0_ASCS00
er.sapAX0_ASCS00 pf=/usr/sap/AX0/SYS/profile/AX0_ASCS00_AIXASCSServ &
exit
```

Adapt instance number and <SAPSID> to suit your installation. You must also grant the file permission to make the script executable.

Adding additional SAP services

Optionally perform the following steps to run a highly available gateway (GW), syslog collector (CO) and syslog sender (SE) service within ASCS:

1. Add to the ASCS start profile the entries to start the three additional services:

```
#-----
# Start syslog collector daemon
# VS 18thMar08: collector start is moved to ASCS instance...
#-----
_CO = co.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_05 = local rm -f $_CO
Execute_06 = local ln -s -f $(DIR_EXECUTABLE)/rslgcoll $_CO
Start_Program_02 = local $_CO pf=$(DIR_PROFILE)/AX0_ASCS00_AIXASCSServ -F
#-----
# Start syslog send daemon
#-----
_SE = se.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_07 = local rm -f $_SE
Execute_08 = local ln -s -f $(DIR_EXECUTABLE)/rslgsend $_SE
Start_Program_03 = local $_SE pf=$(DIR_PROFILE)/AX0_ASCS00_AIXASCSServ -F
#-----
# Start SAP gateway service
#-----
_GW = gw.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_09 = local rm -f $_GW
Execute_10 = local ln -s -f $(DIR_EXECUTABLE)/grwd $_GW
Start_Program_04 = local $_GW -a pf=$(DIR_PROFILE)/AX0_ASCS00_AIXASCSServ
```

where the Execute_0x numbers are unique and in sequence in the file.

2. As <sapsid>adm, add the executables rslgsend, rslgcoll and grwd to the instance executable directory on each cluster node where ASCS can run. For example, enter:

```
cd /usr/sap/AX0/ASCS00/exe
cp -p /sapmnt/AX0/exe/grwd .
cp -p /sapmnt/AX0/exe/rslgcoll .
cp -p /sapmnt/AX0/exe/rslgsend .
```

3. Comment out the collector start entry in the CI start profile:

```
#-----
# Start syslog collector daemon
#-----
# VS 18thMar08: collector start is moved to ASCS instance...
#_CO = co.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
```

```
#Execute_04 = local rm -f $( _CO)
#Execute_05 = local ln -s -f $(DIR_EXECUTABLE)/rslgcoll $( _CO)
#Start_Program_01 = local $( _CO) pf=$( _PF) -F
```

4. Add the following to the default profile:

```
#-----
# VS 18thMar08: The collector is now started in ASCS instance, therefore
#               the host and the ports must be adapted to the ASCS
#               instance number, here 00.
#-----
rslg/collect_daemon/host = AIXASCSserv
rslg/collect_daemon/listen_port = 1400
rslg/collect_daemon/talk_port = 1500
rslg/send_daemon/listen_port = 1200
rslg/send_daemon/talk_port = 1300
```

5. Ensure that the port names used for ASCS GW are defined in `/etc/services` on all nodes in the cluster where the ASCS instance can run. Otherwise, the application server instances will not be able to connect to the gateway server. If your instance number for ASCS is 00, then the following entries are needed:

```
sapgw00 3300/tcp
sapgw00s 4800/tcp # SAP System Gateway Security Por
```

Now you can directly go to “Verification of ABAP SCS with Enqueue Replication” on page 148.

Option 2: Installing ASCS with a physical hostname

If you have installed your ASCS without using `SAPINST_USE_HOSTNAME`, that is using the physical hostname, then you must adapt the ASCS instance to run with a virtual hostname.

1. Change the ASCS instance by performing the following steps:
 - a. Rename the ASCS profile from `<SID>_ASCS<xx>_<real_hostname>` to `<SID>_ASCS<xx>_<virtual_hostname>`.
 - b. Inside the profile you must comment out the following `rdisp/enqname` entry:

```
#-- Must be inactivated for SAP HA with SA for Multiplatforms --
#rdisp/enqname = $(rdisp/myname)
```

Note: If you have NetWeaver 04s SR2 or later it is not necessary to comment out this line.

- c. Add the following entries, if they do not exist already:

```
enque/process_location = LOCAL
enque/server/threadcount = 3
enque/server/replication = true
enque/encni/repl_port = 6502
```

```
ipc/shm_psize_34 = 0
```

We recommend using 6500 plus 100 * ASCS instance number. Ensure that the chosen port number for replication is not used by another application. To check whether the port number is in use, enter the command `lsof -i -P` under Linux or AIX to list currently used ports.

If replication is set to true but the `repl_port` is not defined, the port of the gateway service is used - 3300 plus ASCS instance number in the example below. In this case, either the gateway service or the replication will fail, depending on which service can bind to the port.

Customizing SAP for HA

Note: The parameter `enqueue/backup_file = $(DIR_GLOBAL)/ENQBCK` is not required with enqueue replication and should not be used.

The following steps are required for Linux and AIX but not for USS

2. Copy the ASCS instance directory. SAP installer creates the directories on the installation node. You must create the directories locally on each node where the ASCS instance and the Enqueue Replication server can run. That is the easiest way to have them available on all cluster nodes where the ASCS instance runs.
 - a. As `<sapsid>`, on the installation node enter the following commands:

```
cd /usr/sap/LOP
tar -cvf ASCS02.tar ASCS02
scp ASCS02.tar lopadm@<other cluster node>:/usr/sap/LOP
```
 - b. As `<sapsid>adm`, issue the following commands on each of the other nodes:

```
cd /usr/sap/LOP
tar -xvf ASCS02.tar ASCS02
rm ASCS02.tar
```
 - c. Adapt the entries for starting the `sapstartsrv` services in the file `/usr/sap/sapservices`.
3. Modify the `DEFAULT.PFL` profile as follows:
 - a. If you have not installed the ASCS instance with a virtual hostname, you must change in the `DEFAULT.PFL` file as follows:

```
rdisp/mshost = <virtual ASCS hostname>
enqueue/serverhost = <virtual ASCS hostname>
```
 - b. If you have decided to run the central syslog collector within the ASCS instance by adding the entry in 5 to the ASCS start profile below, then you must also update the `DEFAULT.PFL` with the modified port numbers. In the examples below the virtual hostname is `siccps29`:

```
# the central system log collector runs in the ASCS instance
#-----
# VS 18thMar08: The collector is now started in ASCS instance, therefore
#               the host and the ports must be adapted to the ASCS
#               instance number, here 02.
#-----
rslg/collect_daemon/host = siccps29
rslg/collect_daemon/listen_port = 1402
rslg/collect_daemon/talk_port = 1502
rslg/send_daemon/listen_port = 1202
rslg/send_daemon/talk_port = 1302
```
4. Add the following parameter to all dialog instance profiles:

```
enqueue/con_retries = 120
enqueue/deque_wait_answer = TRUE
```
5. Adapt the start profile of the ASCS instance:
 - a. Rename the `START` profile to include the virtual hostname.
 - b. Change all occurrences of the physical hostname in the profile to the virtual hostname.
 - c. Optionally include the start commands of the gateway, syslog collector daemon and syslog sender daemon by adding the following lines to the profile:

```
#-----
# Start syslog collector daemon
# VS 18thMar08: collector start is moved to ASCS instance...
#-----
_CO = co.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_05 = local rm -f $_CO
Execute_06 = local ln -s -f $(DIR_EXECUTABLE)/rslgcoll $_CO
```

```

Start_Program_02 = local $_CO pf=$(DIR_PROFILE)/LOP_ASCS02_siccps29 -F
#-----
# Start syslog send daemon
#-----
_SE = se.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_07 = local rm -f $_SE
Execute_08 = local ln -s -f $(DIR_EXECUTABLE)/rslgsend $_SE
Start_Program_03 = local $_SE pf=$(DIR_PROFILE)/LOP_ASCS02_siccps29 -F
#-----
# Start SAP gateway service
#-----
_GW = gw.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_09 = local rm -f $_GW
Execute_10 = local ln -s -f $(DIR_EXECUTABLE)/gwrld $_GW
Start_Program_04 = local $_GW -a pf=$(DIR_PROFILE)/LOP_ASCS02_siccps29

```

where the Execute_0x numbers are unique and in sequence in the file.

- d. As <sapsid>adm, add the executables rslgsend, rslgcoll and gwrld to the instance executable directory on each cluster node where ASCS can run. For example, enter:

```

cd /usr/sap/LOP/ASCS02/exe
cp -p /sapmnt/LOP/exe/gwrld .
cp -p /sapmnt/LOP/exe/rslgcoll .
cp -p /sapmnt/LOP/exe/rslgsend .

```

6. Remove from the Central Instance Start profile the entries starting the syslog collector daemon.

Note: Do this only if you have decided to run the syslog collector within the ASCS instance by adding the entry described in 5 on page 144 to the ASCS start profile.

```

#-----
# Start syslog collector daemon
#-----
# VS 18thMar08: collector start is moved to ASCS instance...
#_CO = co.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
#Execute_04 = local rm -f $_CO
#Execute_05 = local ln -s -f $(DIR_EXECUTABLE)/rslgcoll $_CO
#Start_Program_01 = local $_CO pf=$_PF -F

```

7. Define the service name in /etc/services. If your /etc/services is different for each LPAR/host, then 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 02, then the following entries are needed:

```

sapmsLOP 3602/tcp # SAP System Message Server Port
sapdp02 3202/tcp
sapgw02 3302/tcp
sapdp02s 4702/tcp # SAP System Dispatcher Security Port
sapgw02s 4802/tcp # SAP System Gateway Security Port

```

The gateway entries are only necessary if you decide to run the optional gateway within the ASCS instance.

8. Update the /usr/sap/sapservices file. Change the start profile entries for sapstartsrv in /usr/sap/sapservices to the new hostname. In the examples used in this section this is:

```
pf=/usr/sap/LOP/SYS/profile/START_ASCS02_siccps29
```

Now you can directly go to “Verification of ABAP SCS with Enqueue Replication” on page 148 paragraph below.

Installing SAP with classic CI

If you have installed SAP with classic CI before setting up the HA environment, you must:

- manually create a ASCS instance from the existing CI
- manually adapt the SAP profiles and environment to use a virtual hostname.

Prior to SAP NetWeaver 2004s, the SAP installation routine (SAPinst) did not support the installation of an ABAP SCS instance. Therefore, you must create the ASCS instance manually. In order to do this, you must have a running SAP system based on SAP kernel version 6.40.

Preferably, you should have installed the ABAP central instance on z/OS. If you did not install the central instance on z/OS and want to run the ASCS on z/OS as recommended above, then you need to make preparations on the ASCS host similar to those for installing an SAP application server. You can proceed as listed in SAP Note 821904, or you can, as a minimum, perform the installation steps for one dialog instance. This ensures that the parameters for UNIX System Services are appropriate for SAP; the sap<sapsid> user environment is defined; and the standard SAP directory structure is created.

We recommend creating and setting up the ASCS as described in SAP note 821904 *Separating SCS instances for ABAP and Java*. The SAP note describes the procedure, including some tools support.

If you have to do all steps manually, then make the necessary changes listed in the following section. The standalone enqueue architecture represented by the ASCS instance is activated by changing a few parameter in the SAP profiles. For a detailed SAP documentation of installing and using the ABAP standalone enqueue server (ABAP SAP Central Services) see the SAP web site at http://help.sap.com/saphelp_nw04s/helpdata/en/36/67973c3f5aff39e10000000a114084/content.htm. Alternatively you can go to <http://help.sap.com> and search in documentation for *standalone enqueue server*.

You must select an instance number for the new ASCS instance. Please select a unique number, different from all other existing instance numbers. This is necessary, because the ASCS may run on application servers which already run a dialog instance and SAP does not support running several instances with the same instance number on the same host. The new ASCS requires a virtual host name. Therefore, you must define such a virtual host name and IP address. In the examples used here `siccps29` is defined as the hostname for IP address 9.153.165.88. You must also download the latest version of the ASCS executables and `rfcping` utility and put them into the executable directory.

Start with creating an instance profile that is used by all services that belong to ASCS and which uses the virtual hostname associated with the ASCS. In the examples that follow the instance name is `ASCS02` and the virtual host name is `siccps29`.

1. Create the ASCS instance profile, which should have its own instance number and instance name. For example, with a <SAPSID> of `LOP`, an instance of `ASCS02` and a virtual hostname of `siccps29`, you create the profile `LOP_ASCS02_siccps29`, which contains the following:

```
# Profile for enqueue server/message server/gateway/syslog collector...
SAPSYSTEMNAME = LOP
INSTANCE_NAME = ASCS02
SAPSYSTEM = 02
```



```

enqueue/process_location = LOCAL
enqueue/server/replication = true
enqueue/server/threadcount = 3
enqueue/encni/repl_port = 6502
ipc/shm_psize_34 = 0

```

We recommend using 6500 plus 100 * ASCS instance number. Ensure that the chosen port number for replication is not used by another application. To check whether the port number is in use, enter the command `lsof -i -P` under Linux or AIX to list currently used ports.

If replication is set to true but the `repl_port` is not defined, the port of the gateway service will be used - 3300 in the example below. In this case, either the gateway service or the replication will fail, depending on which service can bind to the port.

Note: The parameter `enqueue/backup_file = $(DIR_GLOBAL)/ENQBCK` is not required with enqueue replication and should not be used.

2. This step is required for Linux and AIX, but not for USS. As `<sapsid>adm`, create the instance directory `/usr/sap/LOP/ASCS02` and its subdirectories `data`, `log`, `sec` and `work`.

Create the directories locally on every node where the ASCS instance and the Enqueue Replication server are to run. That is the easiest way to have them available on all cluster nodes where the ASCS instance is to run.

Following the examples given above, enter the following commands:

```

mkdir /usr/sap/LOP/ASCS02
mkdir /usr/sap/LOP/ASCS02/data
mkdir /usr/sap/LOP/ASCS02/log
mkdir /usr/sap/LOP/ASCS02/sec
mkdir /usr/sap/LOP/ASCS02/work

```

```

cd /usr/sap/LOP
tar -cvf ASCS02.tar ASCS02
scp ASCS02.tar lopadm@<other cluster node>:/usr/sap/LOP

```

3. This step is required for Linux and AIX, but not for USS. As `<sapsid>adm`, on each of the other nodes enter the following commands:

```

cd /usr/sap/LOP
tar -xvf ASCS02.tar ASCS02
rm ASCS02.tar

```

4. Save a copy of the `DEFAULT.PFL` under another name, for example `DEFAULT.CentralInstance`. This allows you to fall back to the old architecture if required.

5. Modify the `DEFAULT.PFL` profile. The following example shows the changed entries in **bold**:

```

# message server parameter for ABAP SCS
rdisp/msserv = 3602rdisp/mshost= siccps29
ms/server_port_0= PROT=HTTP, PORT=8102

rdisp/sna_gateway= siccps29
rdisp/sna_gw_service= sapgw02

# enqueue server parameter for ABAP SCS
# rdisp/enqname = $(rdisp/myname) must be inactive! enqueue/serverhost = siccps29
enqueue/serverinst = 02
enqueue/process_location = REMOTESA

```

6. If you have decided to run the central syslog collector within the ASCS instance, then you must also update the DEFAULT.PFL with the modified ports:

```
# the central system log collector runs in the ASCS instance
#-----
# VS 18thMar08: The collector is now started in ASCS instance, therefore
#               the host and the ports must be adapted to the ASCS
#               instance number, here 02.
#-----

rslg/collect_daemon/host = siccps29
rslg/collect_daemon/listen_port = 1402
rslg/collect_daemon/talk_port = 1502
rslg/send_daemon/listen_port = 1202
rslg/send_daemon/talk_port = 1302
```

As noted earlier, siccps29 is the example hostname of the virtual IP address assigned to the ASCS.

7. Add the following parameter to all dialog instance profiles:
enqueue/con_retries = 120
enqueue/dequeue_wait_answer = TRUE
8. Set the number of enqueue processes in the old CI instance profile to 0:
rdisp/wp_no_enq = 0
9. Optionally create or adapt the start profile START_ASCS02_siccps29.
We recommend creating this start profile, because then you can use startsap and stopsap (see SAP note 809477) to start and stop the ASCS instance in case you need to start the instance manually.
You can copy the start profile of the existing CI and then modify it. Please see 1 on page 142 for examples of the modified entries to start the three additional and optional services.
10. Define service name in /etc/services. If your /etc/services is different for each LPAR/host, then 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 02, then the following entries are needed:

```
sapdp02 3202/tcp
sapgw02 3302/tcp
sapdp02s 4702/tcp # SAP System Dispatcher Security Port
sapgw02s 4802/tcp # SAP System Gateway Security Port
```

The gateway entries are only necessary, if you decided to run the optional gateway within the ASCS instance.

The ABAP message server sapms<sapid> must also have an entry similar to:
sapmsLOP 3602/tcp # ASCS Message Server Port

11. Remove from the Central Instance Start profile START_DVEBMGS00_<CI host name> the entries starting the message server and the syslog collector daemon.

Do this only if you have decided to run the syslog collector within the ASCS instance.

Verification of ABAP SCS with Enqueue Replication

When you have completed the above steps, you must verify that your double-stack system runs without problems. You can start and stop the ASCS services manually.

1. First you have to activate the VIPA under z/OS USS or the IP alias under Linux. Under *Unix System Services* you should use this `moddvipa` command as root user:

```
moddvipa -p tcpip -c 10.101.5.194 (if this is the IP address of halascsv)
```

Under *Linux* you should use `ifconfig` command as root user:

```
ifconfig eth0:1 <virt. host IP> broadcast < broadcast ip> netmask <netmask> mtu 1500 up
```

Note: Check if the alias `eth0:1` is already in use by entering the command `ifconfig -a`.

2. Manually start the ASCS00 instance as `<sapsid>adm`:


```
startsap r3 ASCS00 halascsv
```
3. Either: (1) Manually start the enqueue replication server on another machine:


```
enrepserver pf=/usr/sap/HA1/SYS/profile/HA1_ASCS00_halascsv &
```

or (2) Use the adapted `StartABAP_EnqueueReplicationServer` script.

Afterwards verify manually that the SAP system is running correctly with ASCS by using SAP transaction SM12 to generate test entries in the enqueue table. See Chapter 16, “Verifying your implementation on Linux/AIX,” on page 289 for further information.

Generating entries and displaying them is only possible when the enqueue server of ASCS is running, and not when it is stopped.

To check if the replication server has successfully connected to the ABAP standalone enqueue server, use `ensmon pf= <profile> -H <hostname>`. For example, enter:

```
ensmon pf=/usr/sap/HA1/SYS/profile/HA1_ASCS00_halascsv
```

From the main menu choose **1: Dummy request**. The dummy request should execute successfully.

Then choose **2: Get replication information**. This should return “Replication is enabled in server, repl. server is connected Replication is active” and should also show statistics.

To display the generated test entries, run:

```
enqt pf=/usr/sap/HA1/SYS/profile/HA1_ASCS00_halascsv 20
```

Kill the enqueue server and message server manually. If you have running the optional services (GW, CO and SE) kill them also. Then move the VIPA to the system, where the replication server is started. Start the ASCS instance there. Then the replication server must stop itself after the enqueue table has been rebuild.

Again run:

```
enqt pf=/usr/sap/HA1/SYS/profile/HA1_ASCS00_halascsv 20
```

If it shows the same output as before you killed the enqueue server, then you have *verified that replication works*.

Installing and configuring Java SAP Central Services (SCS)

You can install a Java stack (either as Java only stack or as part of a double-stack installation) only with a separate SCS instance. You have two installation options for installing the Java (double-stack) SCS instance:

- **Option 1: installing with virtual hostname**

This is the recommended option. Use SAPINST option `SAPINST_USE_HOSTNAME`, for example:

```
./sapinst SAPINST_USE_HOSTNAME =<virtual SCS host>
```

Post-Installation tasks should also refer to <virtual SCS host>.

- **Option 2: install with physical hostname** (before setting up the HA environment) Please refer to SAP note 757692 *Changing the hostname for J2EE Engine 6.40 installation* for a description of the necessary changes when switching the J2EE Engine from physical to virtual hostname.

SAPinst installation of the SCS with its own virtual hostname means that only one manual profile change is necessary to run under the pre-canned SA MP SAP policy. See Appendix E, “Sample Tivoli System Automation for Multiplatforms high availability policy for SAP,” on page 325 for further information about the SA MP policy. The change is that you must switch on **Enqueue table replication**. “3. Manually make the changes to enable enqueue table replication” on page 153 explains how to do this.

If you have not installed the SCS with its own virtual hostname, you have to make additional manual profile changes and changes in the Java configuration database with the SAP configtool, as described below.

1. Changes necessary if your system was installed with Option 2

If your system is installed with Option 2 you must switch the SCS to the new virtual host name. In the examples used below and throughout this chapter, the SCS instance number is 01 and the virtual hostname is `siccps40`. In order to do this, make the changes described in SAP note 757692 (Version 10 from 25.01.2005) and 821904 (Version 4 from 06.10.2005). The steps listed below are an excerpt from 757692/821904. All that is required is to change the host name of the Java enqueue and message server of the SCS.

Complete the following steps as <sapsid>adm:

1. Before changing the SCS enqueue/message server hostname stop all J2EE engine processes cleanly.
2. Change the hostname occurrences in the SCS instance profile located in `./usr/sap/<sid>/SYS/profile`, as well as the names of the profiles (including the START profile) to the new virtual SCS hostname.
3. Comment out the following `rdisp/enqname` entry in the SCS instance profile as follows:

```
#-- Must be inactivated for SAP HA with SA for Multiplatforms  
--#rdisp/enqname = $(rdisp/myname)
```

Note: If you have NetWeaver 04s SR2, do not comment out the entry as SAP is now able to handle it correctly.

Check these entries and add them if required:

```
enque/serverhost = siccps40  
enque/process_location = LOCAL  
enque/server/replication = true  
enque/server/threadcount = 3  
enque/encni/repl_port = 6501
```

```
ipc/shm_psize_34 = 0
```

4. Change in `DEFAULT.PFL`:

```
j2ee/scs/host = <virtual SCS hostname>
```

For example, to change <virtual SCS hostname> to siccps4, enter:

```
j2ee/scs/host = siccps4
```

5. Modify the entries for starting the sapstartsrv services in the file /usr/sap/sapservices.
6. Modify the node host settings according to SAP Note 821904 in the J2EE configuration database as follows:
 - a. Change to the configtool directory and run the **configtool**:


```
cd /usr/sap/<SID>/<CI_INSTANCE_NAME>/j2ee/configtool
./configtool.sh
```
 - b. Connect to the database and expand in the left tree the following items: cluster-data> Global Dispatcher Configuration> managers> Cluster Manager
 - c. Change the following attributes:


```
mshost = <scshost> (for the examples above this would be siccps40)
ms.port = <j2ee_ms_port> (for the examples above this would be 3601)
```
 - d. In the left hand tree choose: cluster-data> Global Dispatcher Configuration> managers> Locking Manager
 - e. Change the following attributes:


```
enqu.host = <scshost> (in our case siccps40)
enqu.port = <j2ee_es_port> (in our case 3201)
enqu.profile.filename = /usr/sap/<SID>/SYS/profile/
<SID>_SCS<INO>_<scshost> (in our case /usr/sap/LOP/SYS/profile/
LOP_SCS01_siccps40)
```
 - f. In the left hand tree choose: cluster-data > Global Server Configuration> managers> Cluster Manager.
 - g. Change the following attributes:


```
mshost = <scshost> (in our case siccps40)
ms.port = <j2ee_ms_port> (in our case 3601)
```
 - h. In the left hand tree choose: cluster-data > Global Server Configuration> managers> Locking Manager.
 - i. Change the following attributes:


```
enqu.host = <scshost> (in our case siccps40)
enqu.port = <j2ee_es_port> (in our case 3201)
enqu.profile.filename = /usr/sap/<SID>/SYS/profile/
<SID>_SCS<INO>_<scshost> (in our case /usr/sap/LOP/SYS/profile/
LOP_SCS01_siccps40).
```
7. For each INSTANCE_IDxxxxx complete the following steps:
 - a. In the left hand tree choose: cluster-data>Instance_IDxxxxx.
 - b. Change the following attributes:


```
Message server setting: <scshost> (in our case siccps40)
Message server port: <j2ee_ms_port> (in our case 3601)
```
 - c. In the tree on the left-hand side go to Instance_IDXXXX -> Dispatcher_IDXXX -> managers -> Cluster Manager:


```
Change ms.host property so that it points to the new hostname.
Perform the same for the server node.
```
 - d. Go to instance -> dispatcher -> Locking manager:


```
Change enqu.host to point to the new hostname.
If you have changed the profile names modify also the enq.profile.filename accordingly.
```

Perform the same steps for the server node.

- e. In the left hand tree choose: cluster-data> Instance_IDxxxxx> dispatcherIDxxxxx >managers>ConfigurationManager: secstore*:
- f. Check for the correct path:(/sapmnt/<SID>/global...)
- g. In the left hand tree choose: cluster-data> Instance_IDxxxxx> serverIDxxxxx >managers>ConfigurationManager: secstore*:
- h. Check for the correct path:(/sapmnt/<SID>/global...)
- i. Check in file /usr/sap/<SID>/<InstanceDIR>/j2ee/cluster/bootstrap/bootstrap.properties, that the path for the secstore* entries is correct.
- j. In the left hand tree choose: cluster-data>Global Server Configuration>services>com.sap.security.core.ume.service.
- k. Change the following attributes:
ume.r3.connection.master.msgghost = <ascshost> (in our case siccps29)
ume.r3.connection.master.msserv = <abap_ms_port> (in our case 3602)

Note: This is only necessary if a login was configured for the Java RFC user SAPJSF using logon groups - in other words, using the SAP message server.

- l. Apply all the changes by choosing File > Apply.
- m. Exit from **configtool**.

The above steps update the J2EE configuration database. The bootstrap process detects those changes and updates the .properties files which also contain the Java enqueue and message server hostname, such as ../j2ee/cluster/dispatcher/cfg/kernel/LockingManager.properties or ../j2ee/cluster/server0/cfg/kernel/LockingManager.properties.

For the *J2EE engine*, you are required to perform another step with the *Offline Config Editor*. In this step, you must ensure that you have shut down all of the Java instances (nodes):

1. cd /usr/sap/<SID>/<CI_INSTANCE_NAME>/j2ee/configtool
./offlinecfgeditor.sh
2. Switch to editing mode. Then navigate to the entry:
- cluster_data --> Property sheet instance.properties.IDxxxxx
3. Double-click to open the property sheet. Check whether there are entries for instance.en.host und instance.en.port. If these entries do not exist, do not change anything and exit the editor. If these entries *do exist*, set them to the following values:
+ instance.en.host: <scshost>
+ instance.en.port: <j2ee_es_port>
4. Exit the *Offline Config Editor*.

2. Manually create the SCS instance directory on switch-over nodes

Note: This section does not apply to USS installations.

SAP installer creates the SCS instance directories on the installation node only. You must create the directory structure also on all other nodes locally where the SCS instance and the SCS Enq. Replication server runs. That is the easiest way to have them available on all cluster nodes where the SCS instance runs.

Log in as <sapsid>adm and complete the following steps, adapting the instance number where appropriate:

1. Tar the instance directory on the installation node and copy it to the switch-over node:

```
cd /usr/sap/<SID>
tar -cvf SCS01.tar SCS01
scp SCS01.tar <sid>adm@<switch-over node>:/usr/sap/<SID>
```

2. On the switch-over node extract the tar file created in the step above:

```
cd /usr/sap/<SID>
tar -xvf SCS01.tar SCS01
rm SCS01.tar
```

3. Manually make the changes to enable enqueue table replication

In the Java SCS instance profile:

Add the following parameter to the instance profile of Java SCS, which is in the example LOP_SCS01_siccps40. These additions are always necessary, even if you have installed your Java SCS using 149 and also for a Java only installation such as. Enterprise Portal 6.0 (EP 6.0). The reason is that the standard HA installation using 149 does not enable replication.

Add the following to enable enqueue table replication:

```
enqueue/serverhost = halscsv

enqueue/server/replication = true
enqueue/encni/repl_port = 6501
enqueue/server/threadcount = 3
```

Note: We recommend using 6500 plus 100 * ASCS instance number. Ensure that the chosen port number for replication is not used by another application. To do so, at the Linux or AIX command prompt enter `lsof -i -P` to list currently used ports.

In the Java ERS instance profile:

If you have setup your SAP system with an own Java ERS instance, then you must add the following parameter to the instance profile of the Java ERS instance(s):

```
enqueue/encni/repl_port = 6501
```

Note: We recommend using 6500 plus 100 * ASCS instance number. Ensure that the chosen port number for replication is not used by another application. To do so, at the Linux or AIX command prompt enter `lsof -i -P` to list currently used ports.

This ensures that the Java Enqueue Replication server knows the port on which the standalone Java Enqueue server is listening. This is valid according to SAP up to SAP kernel 7.00 releases where a manual ERS setup is required..

We recommend adding to the home directory of the <sapsid>adm a small script in order to start ERS manually, for example:

```
#cat StartJ2EE_EnqueueReplicationServer
#-----
# Start SAP J2EE enqueue replication service
#-----
```

```
/bin/rm -f er.sapLOP_SCS01
/bin/ln -s -f /usr/sap/LOP/SCS01/exe/enrepsrver er.sapLOP_SCS01
er.sapLOP_SCS01 pf=/usr/sap/LOP/SYS/profile/LOP_SCS01_siccps40 &
exit
```

Verification of Java SCS with Enqueue Replication

You must verify that your double-stack system runs without problems with the new virtual host name and the replication server. You can start and stop the SCS services manually.

1. First you have to activate the VIPA under z/OS USS or the IP alias under Linux. Under *Unix System Services* you should use this `moddvipa` command as root user:

```
moddvipa -p tcpip -c 10.101.5.195 (if this is the IP address of halscsv)
```

Under *Linux* you should use `ifconfig` command as root user:

```
ifconfig eth0:2 <virt. host IP> broadcast < broadcast ip> netmask <netmask> mtu 1500 up
```

Note: Check if the alias `eth0:2` is already in use by entering the command `ifconfig -a`.

2. Manually start the SCS01 instance as `<sapsid>adm`:

```
startsap r3 SCS01 halscsv
```
3. Either: (1) Manually start the enqueue replication server on another machine:

```
enrepsrver pf=/usr/sap/HA1/SYS/profile/HA1_SCS01_halscsv &
```


or (2) Use the adapted `StartJ2EE_EnqueueReplicationServer` script.

Verify manually afterwards that the SAP SCS is running correctly. Use the utility that SAP provides **enqt**. Run as `<sapsid>adm`:

```
enqt pf=/usr/sap/HA1/SYS/profile/HA1_SCS01_halscsv 97
```

The output is similar to:

```
---REQ-----
EnqId:          EnqTabCreaTime/RandomNumber   = 06.09.2005 00:06:19 1125957979 / 8563
ReqOrd at Srv:  TimeInSecs/ReqNumberThisSec   = 09.09.2005 13:45:43 1126266343 / 1
-----
```

`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_halscsv 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 `ensmon pf= <profile> -H <hostname>` to check if the replication server has successfully connected to the Java standalone enqueue server. For example, run :

```
ensmon pf=/usr/sap/HA1/SYS/profile/HA1_SCS01_halscsv
```

From the main menu choose

- 1: **Dummy request** . The dummy request should be executed successfully.

Then choose

2: Get replication information. This should return the information that *Replication is enabled in server, repl. server is connected Replication is active* and should also show statistics lists.

Kill the enqueue server and message server manually. Then 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 shows the same output as before you killed the enqueue server, then you have verified that replication works.

Stop the manually started processes and then start your SAP system normally to ensure it starts.

Your double-stack setup for high availability is complete and you can start creating the TSA for MP resources for your double-stack system. The first step in creating the SA MO resources is to adapt the parameter values contained in the sample ABAP and J2EE configuration files to your actual environment.

SAP profile parameters

The following table lists and describes the profile parameters that are related to ASCS/SCS.

Figure 37. SAP profile parameters relevant for the high availability solution

Parameter	Description	Default value	Recommended value
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/server/replication	Enables replication.	false	true
enqueue/encni/repl_port	Port number which the enqueue server opens for replication. Note: The default value is in conflict with the gateway port. Therefore, you MUST choose a different port if the gateway is part of ASCS. This is the case in our sample policy. Additionally, ensure that the chosen port number is not used by another application. For example, you can use 'netstat' under USS or 'lsof -i -P' under Linux to list currently used ports.	3300 + <instance number>	<port number> (for example 6000) + <instance number>
enqueue/server/threadcount	Number of I/O threads in the enqueue server.	1	2 or 3

Customizing SAP for HA

Parameter	Description	Default value	Recommended value
enque/ dequeue_wait_answer	Indicates whether a dequeue request waits for the acknowledgement. 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
enque/backup_file	Specifies where the enqueue server saves the locks on shutdown. If a shared file system is used, the default value is satisfactory. Note: You are strongly recommended not to set this parameter for an HA setup.	/usr/sap/<SID>/ D<instance no.>/log/ENQBCK	
enque/con_retries	Number of seconds the application server tries to reconnect to the enqueue server before an error is indicated to the application. If you do not set the value to 0, it should be larger than the typical time for an (A)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.	60	120
rdisp/mshost	Host of the message server.		<virtual hostname>
rdisp/sna_gateway	Host where the SNA gateway is running.		<virtual hostname>
rdisp/ sna_gw_service	TCP service under which the SNA gateway can be reached.		sapgw<instance number>
rslg/ collect_daemon/ host	Host of the syslog collector.		<virtual hostname>
rdisp/vbname	Application server that runs update work processes. Note: This is now obsolete as update requests are dispatched among appropriate instances automatically.		\$(rdisp/myname)
ipc/shm_psize_34	Shared memory segments used by enqueue server and replication server. When size is set to 0 the segments are allocated directly, not as pools. Set this parameter only in the instance profile of the enqueue server.		0

Preparing SAP on z/OS for automation

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

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

For a comprehensive list of scripts and other key files, see Appendix D, “Description of the z/OS high availability scripts,” on page 319.

C-shell and logon profiles

Tivoli System Automation needs to execute UNIX commands on behalf of the SAP administrator to be able to manage SAP. It invokes these commands by starting the user’s default shell and naming the shell script that is to be executed (for example: `/bin/tcsh -c '<command>'`). The C-shell is usually defined as the default shell for the SAP administrator ID.

The C-shell knows four profiles:

- `/etc/csh.cshrc`
- `/etc/csh.login`
- `$HOME/.cshrc`
- `$HOME/.login`

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

ABAP SAP Central Services (ASCS)

ASCS is a collection of unique SAP resources that share the same instance profile and instance directory:

- ABAP enqueue server
- ABAP message server

Optionally:

- Gateway server
- Syslog collector
- Syslog sender

In order to allow transparent failover of the ASCS to another system, the enqueue server must restart on the system that keeps a copy of the actual enqueue table.

To allow detailed monitoring and faster recovery, all resources are started, stopped and monitored individually. For this purpose, we created the `sapctrl_em` shell script. See Appendix D, “Description of the z/OS high availability scripts,” on page 319 for a detailed description.

The shell script must be adapted to your environment.

Customizing SAP for HA

The individual components are started as follows:

Table 14. Using `sapctrl_em` to manage ABAP SCS and its resources

Command	Result	Example
<code>sapctrl_em ES</code>	Starts the ABAP enqueue server	<code>z/OS USS: sapctrl_em ERP HA1 ASCS00 halascsv 0 ES START</code>
<code>sapctrl_em ERS</code>	Starts the ABAP enqueue replication server	<code>z/OS USS: sapctrl_em ERP HA1 ASCS00 halascsv 0 ERS START</code>
<code>sapctrl_em MS</code>	Starts the ABAP message server	<code>z/OS USS: sapctrl_em ERP HA1 ASCS00 halascsv 0 MS START</code>
<code>sapctrl_em GW</code>	Starts the gateway	<code>z/OS USS: sapctrl_em ERP HA1 ASCS00 halascsv 0 GW START</code>
<code>sapctrl_em CO</code>	Starts the syslog collector	<code>z/OS USS: sapctrl_em ERP HA1 ASCS00 halascsv 0 CO START</code>
<code>sapctrl_em SE</code>	Starts the syslog sender	<code>z/OS USS: sapctrl_em ERP HA1 ASCS00 halascsv 0 SE START</code>

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 will run:

```
saplicense -get
```

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

Different licenses are required if SCS runs on zLinux, AIX or xLinux cluster.

Java Central Services

Java SCS is a collection of unique SAP resources that share the same instance profile and instance directory:

- Java enqueue server
- Java message server

In order to allow transparent failover of the Java SCS to another system, the enqueue server must run on the system that keeps a copy of the current enqueue table.

To allow detailed monitoring and faster recovery, all resources are started, stopped and monitored individually. For this purpose, we created the `sapctrl_em` shell script. See Appendix D, "Description of the z/OS high availability scripts," on page 319 for a detailed description.

The shell script must be adapted to your environment.

The individual components are started as follows:

Table 15. Using `sapctrl_em` to manage JAVA SCS and its resources

Command	Result	Example
<code>sapctrl_em ES</code>	Starts the Java enqueue server	<code>z/OS USS: sapctrl_em ERP HA1 SCS01 halscsv 0 ES START</code>
<code>sapctrl_em ERS</code>	Starts the Java enqueue replication server	<code>z/OS USS: sapctrl_em ERP HA1 SCS01 halscsv 0 ERS START</code>
<code>sapctrl_em MS</code>	Starts the Java message server	<code>z/OS USS: sapctrl_em ERP HA1 SCS01 halscsv 0 MS START</code>

ABAP application server instances

We created three shell scripts (see Appendix D, “Description of the z/OS high availability scripts,” on page 319) that allow you to start, stop and check remote ABAP application server instances. With the availability of SA MP, together with the end-to-end Application Manager, we strongly recommend using SA MP and SA AM to control remote ABAP application servers running on Linux/AIX platforms.

startappsrv_v5 <hostname> <instnr> <instancedir> [<via>]
Starts an ABAP application server instance.

stopappsrv_v5 <hostname> <instnr> <instancedir> [<via>]
Stops an ABAP application server instance.

checkappsrv_v4 <hostname> <instnr>
Starts an ABAP application server monitor

These shell scripts are listed in Appendix D, “Description of the z/OS high availability scripts,” on page 319. The host name (<hostname>), instance number (<instnr>), and instance directory (<instancedir>) identify the instance to be managed.

The parameter <via> is optional. It identifies the remote execution type (REXEC or SSH) used to send commands to remote application servers (running under AIX, Linux on System z, or Windows). If a remote application server is started or stopped, the default is REXEC. It can also be set to SSH if the remote application server is controlled via SSH.

You are recommended to use SSH, since it has two advantages compared to REXEC:

- There is no password-expiry.
- A “plain text” file is not required under USS.

What the shell scripts do

In the following section, we describe what tasks the shell scripts perform.

startappsrv_v5:

- For a remote ABAP application server, it first checks if the database can be reached at all via R3trans. In the case of a remote ABAP application server, it first checks if the remote host can be reached via ‘ping’ and, if so, it then checks if 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. It then checks whether the instance is already running by using

Customizing SAP for HA

the SAP utility rfcping (see “rfcping”). 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, checkappsrv_v4 terminates and Tivoli System Automation simply issues the startappsrv_v5 command again. Based on the notification of the active state, Tivoli System Automation now starts checkappsrv_v4 again.

Using this approach, Tivoli System Automation only has to monitor a single process, namely the one started by checkappsrv_v4.

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

```
cleanipc <instnr> remove
stopsap r3 <instancedir>
startsap r3 <instancedir>
```

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

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

stopappsrv_v5:

- The ABAP application server is stopped by invoking the following scripts:
stopsap r3 <instancedir>

checkappsrv_v4:

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

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

The script creates a link to the rfcping executable such as

```
./rfcping_<hostname>_<instance number>
```

in the current directory, which must be the home directory of the <sapsid>adm. In the SA policy, this must be added to the USS Path as ./rfcping_<hostname>_<instance number> for correct monitoring via SA.

rfcping: This utility establishes an RFC connection to an ABAP application server and retrieves the SAP system information. The command line parameters allow you to choose between different modes.

- The default option is that rfcping closes the RFC connection and ends after it gets a response from the ABAP application server. This is used in the startappsrv_v5 script to check whether an application server instance is up and running.
- Another option specifies that rfcping stays connected and sends a dummy request every few seconds. It only ends if a problem occurs. This mode is used in the checkappsrv_v4 script to monitor an ABAP application server instance.

Remote execution

With the availability of SA MP, together with its end-to-end management automation component, we strongly recommend using this environment to control remote ABAP application servers running on platforms supported by SA MP.

Remote execution using the `rexec` command with `userid` and `password` supplied in a `.netrc` file is not recommended for these platforms. It would imply that the user ID and password of the remote system are stored in plain text on z/OS. Furthermore, if the password is changed on the remote system, the `.netrc` file must be changed as well.

A better alternative is to use OpenSSH. This is a secure shell which allows different methods of authentication. It is available as a Program Product for z/OS (IBM Ported Tools for z/OS) and as an Open Source product on most other platforms including Linux on System z, AIX, and Windows.

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

http://www.ibm.com/servers/eserver/zseries/zos/unix/port_tools.html
<http://www.openssh.org/>

As you can see in the `startappsrv_v5` script, the remote execution command is executed in background. The reason for this is that `rexec` waits until all started processes have ended or have detached as demons redirecting the standard file descriptors (`stdin`, `stdout`, `stderr`). However, the `startsap` script invokes `saposcol` as a normal child process, which implies that the remote execution command waits for `saposcol` to finish.

Remote control of Windows application servers

Application servers on Windows can be remotely controlled by Tivoli System Automation using `rexec` or `ssh`. Because Windows itself does not support remote execution functionality, you need a separate product (e.g. Ataman TCP Remote Logon Services) or an OpenSource package. See <http://www.openssh.org/windows.html> for further information.

With Ataman, the caller does not have a user-specific environment; rather, the system-wide definitions apply. Therefore, it is required to define a common directory and add it to the system-wide `PATH` environment variable, for example `c:\sap\control`. This directory contains batch files to control the SAP instance(s). Furthermore, Ataman does not support concatenation of commands (separated by `;`) in a single `rexec` call. This is another reason for using batch files. See Appendix D, “Description of the z/OS high availability scripts,” on page 319 for details.

Java and double-stack application server instances

Java-only and double-stack application server instances only run on *non-z/OS systems*. Using SA MP, together with the end-to-end Application Manager, we strongly recommend that you use SA MP and SA AM to control remote Java application servers running on Linux/AIX platforms.

To start and check remote Java application server instances from USS, we have created two *additional* shell scripts:

- `startjappsrv` (described in “`startjappsrv`” on page 162)
- `checkjappsrv` (described in “`checkjappsrv`” on page 163)

The SA z/OS resources are “proxy” resources. The start/stop of the real resources is done either via SSH or REXEC. Monitoring is done with the TCPIP-based Java utility `GetWebPage`.

“Proxy” resource also means that a normal stop of the Appl. Server resource under SA z/OS *only stops the monitoring utility*. If you want to really stop the Appl. Server on the *remote machine*, you must use the *force* stop option.

Customizing SAP for HA

It is also important to understand that a “double 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 SA, one double-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 java program *GetWebPage*, which does a primitive health check of the Java application server.

The syntax and processing of the scripts are now described.

Note: The related *stopappsrv* script is described in Appendix D, “Description of the z/OS high availability scripts,” on page 319. It has the syntax `stopappsrv_v5 <hostname> <instnr> <instancedir> [<via>]`. It is used in the same way as when stopping ABAP instances.

startjappsrv

This is the syntax of the *startjappsrv* script, which starts a Java application server instance:

```
startjappsrv <hostname> <instnr> <instancedir> <InstType> [ <via> ]
```

- The host name (<hostname>), instance number (<instnr>), and instance directory (<instancedir>) identify the instance to be managed.
- The installation type (InstType) identifies if the instance is an Java only instance, or if it is part of a double stack instance. It must be '1' for double stack AS, or '2' for a Java-only application server instance.
- <via> is an *optional* parameter. It identifies the remote execution type (REXEC or SSH) used to send commands to remote application servers (running under AIX, Linux on System z):
 - If a remote application server is started or stopped, the default type is REXEC.
 - However, if the remote application is controlled via SSH we highly recommend that you set type to SSH.
- Two lines need to be adapted to your environment:
 - `cd /u/ha1adm` : Adapt the first line to your home directory of <sapsid>adm.
 - `Max_HC_Retries=24` : The number of retries is defined for the health checker (HC) call `java GetWebPage`.
- Because the script sleeps 10 seconds between each HC call, it gives up after 240 seconds and then returns a STARTUP FAILED message. If starting the Java AppServer takes longer in your environment, you should adapt this value.

The *startjappsrv* script performs this processing:

1. Checks that the java path is set.
2. Checks that the Java AppServer is already up and working. This step protects a running application server instance from unnecessary restarts. For example, in case of an intermittent communication error, checkjappsrv terminates and System Automation simply issues the startjappsrv command again. Depending upon the notification of the active state, System Automation now starts checkjappsrv again. Using this approach, SA only has to monitor a single process, namely the one started by checkjappsrv.
3. If the AppServer is not active and if the installation type is 'Java only', it checks if the remote host can be reached via 'ping'. If it can be reached, it send the 'cleanipc, stopsap and startsap' command sequence for the Java-only application server to the remote host (as for the ABAP case).
4. Waits up to 240 seconds (Max_HC_Retries * 10) for the Java instance to start (by default). As mentioned above, you can adjust this default.

checkjappsrv

This is the syntax of the *checkjappsrv* script, which starts a Java application server monitor (GetWebPage):

```
checkjappsrv <hostname> <instnr>
```

- The host name (<hostname>) and instance number (<instnr>) identify the instance to be managed.
- Adapt line `cd /u/ha1adm` to your home directory of <sapsid>adm.

The checkjappsrv script performs this processing:

1. Checks that the java path is set.
2. Performs a health check by calling the GetWebPage java utility (described in "GetWebPage" on page 330). The script creates a link to the java executable, such as `./javaexe_<hostname>_<instance number>`, in the current directory. This must be the home directory of the <sapsid>adm. In the SA policy, this must be added to the USS Path as `./javaexe_<hostname>_<instance number>` to ensure correct monitoring via SA.

saposcol

Usually you would have one SAP OS collector saposcol running on each server on which SAP applications are running.

In the z/OS SYSPLEX environment we have the special situation that we need only *one* saposcol and *one* SAP monitoring agent sapccmsr (described under "sapccmsr") running per SYSPLEX. The SA z/OS SAP "best practice policy" ensures that this is the case.

sapccmsr

This section describes how you setup up and register a *SAP monitoring agent* sapccmsr, which should only run once per z/OS sysplex. This agent should be registered in a way that it is easily identifiable as *belonging to the z/OS sysplex* in SAP transaction RZ21 .

1. Set up a directory which is shared by all LPARs in the z/OS sysplex. This directory will be used to hold the profiles and the working files used by sapccmsr. This directory is later referred to as:

```
<sap-ccms-work-and-profile-directory>
```

Customizing SAP for HA

which is the sample value implemented in the SAP best practice policy
/usr/sap/ccms.

2. Choose a name which makes the sapccmsr easily identifiable in SAP transaction RZ21. A good choice is the z/OS *sysplex name* (sample value: **COHPLEX**). The agent must have this name defined as the value for the SAPLOCALHOST variable in its profile. This name will be referred to below as:

<sysplex-name>

The registration process of a SYSPLEX-wide sapccmsr consists of the following steps:

1. Make sure a directory <sap-ccms-work-and-profile-directory> is shared among all z/OS systems of the sysplex.
2. Create a start profile cmsconf. For details of how to do so, refer to chapter "Creating the CSMCONF Start File for CCMS Agents" in the *SAP Monitoring Setup Guide*, for the corresponding SAP product.
3. Create a subdirectory called sapccmsr in <sap-ccms-work-and-profile-directory>.
4. Copy cmsconf into directory <sap-ccms-work-and-profile-directory>/sapccmsr.
5. Create a profile <ccms-profile> in <sap-ccms-work-and-profile-directory>, which is the sample filename implemented in the SAP "best practice policy" ccms.pfl. This policy *must* contain containing the following entries:

```
DIR_PERF=<sap-ccms-work-and-profile-directory>  
SAPSYSTEMNAME=<SAP system-ID>  
SAPLOCALHOST=<sysplex-name>
```

6. Register sapccmsr by issuing this command:

```
sapccmsr -R pf=<sap-ccms-work-and-profile-directory>/sapccmsr/<ccms-profile>
```

This generates an RFC Destination for CCMS agent sapccmsr called:

```
SAPCCMSR.<sysplex-name>.99
```

An example is now provided which illustrates how to register sapccmsr for a sysplex. In this example, the values for the place holders (<...>) discussed previously are:

```
<sap-ccms-work-and-profile-directory> = usr/sap/ccms  
<ccms-profile> = ccms.pfl  
<SAP system-ID> = HA1  
<sysplex-name> = COHPLEX
```

In this example, we have a shared directory /usr/sap/ccms containing a subdirectory sapccmsr.

Contained in /usr/sap/ccms/sapccmsr are the:

- Start profile: cmsconf
- Profile: ccms.pfl

Profile ccms.pfl contains these entries:

```
DIR_PERF=/usr/sap/ccms  
SAPSYSTEMNAME=HA1  
SAPLOCALHOST=COHPLEX
```

To register sapccmsr we issue this command:

```
sapccmsr -R pf=/usr/sap/ccms/sapccmsr/ccms.pfl
```

which generates an RFC Destination for CCMS agent sapccmsr called SAPCCMSR.COHPLEX.99.

SAP transaction RZ21 can then be used to check the status of the CCMS agent on the sysplex COHPLEX.

Additional SAP setup for RFC connections

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

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

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

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



Figure 38. Defining the gateway host for rfcoscol with SAP transaction SM59

SAProuter

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

Summary of start, stop and monitoring commands

Table 16 summarizes the start, stop and monitoring commands that are needed when you set up the Tivoli System Automation policies for SAP.

Table 16. Summary of start/stop monitoring commands

Actions	Value or command
VIPA for ASCS	
- start command (started task)	S TCPVIPA,VIPA=172.20.10.1

Table 16. Summary of start/stop monitoring commands (continued)

Actions	Value or command
Application server instances: - start command - poststart (monitor) command - stop command - process name to be monitored	<pre> /u/<sapsid>adm/startappsrv_v5 <hostname> <instnr> <instancedir> [<via>] /u/<sapsid>adm/checkappsrv_v4 <hostname> <instnr> /u/<sapsid>adm/stopappsrv_v5 <hostname> <instnr> <instancedir> [<via>] /usr/sap/<SAPSID>/rfc/rfcping_ <hostname>_<instnr> </pre>
saposcol: - start command - stop command - process name to be monitored	<pre> /usr/sap/<SAPSID>/SYS/exe/run/ saposcol -l /bin/kill -2 %PID% /usr/sap/<SAPSID>/SYS/exe/run/ saposcol </pre>
sapccmsr: - start command - stop command - process name to be monitored	<pre> /usr/sap/<SAPSID>/<instance>/exe/ sapccmsr -DCCMS pf=/usr/sap/ccms/ sapccmsr/ccms.pfl /bin/kill -2 %PID% /usr/sap/<SAPSID>/<instance>/exe/ sapccmsr </pre>
VIPA for saprouter: - start command (started task)	<pre> S TCPVIPA,VIPA=172.20.10.3 </pre>
saprouter: - start command - stop command - process name to be monitored	<pre> /usr/sap/<SAPSID>/SYS/exe/run/ saprouter -r /usr/sap/<SAPSID>/SYS/exe/run/ saprouter -s /usr/sap/<SAPSID>/SYS/exe/run/ saprouter </pre>

Note: ABAP and Java SCS are managed by the *sapctrl_em* script. It is used to start, stop, and check the resources of the ABAP and Java SCS instances MS, ES, ERS, CO, SE, GW. Please see Appendix E Automation Scripts for details and invocation syntax.

Other installation issues and recommendations

SAP license:

You need an SAP license for each z/OS system on which the SCS can run and which is a physically separate machine. The same is true in a heterogeneous

Customizing SAP for HA

environment if you run the SCS on another application server platform that is supported by the IBM SAP solution on System z and also by SA MP.

SAP logon groups:

Setup logon groups covering at least two application servers. This can be done by using transaction 'smlg'.

Define Primary and Secondary Datasharing members for DB2 connection failover. This can be done by using transaction 'dbacockpit'.

Enable all non CI application servers to run Batch, Update and Spool services:

Add Batch, Update and Spool services to all application servers.

Part 5. System Automation

Chapter 11. Customizing Tivoli System

Automation for z/OS	171
Preparing SA z/OS for SAP high availability	171
Before you start	171
Setting initialization defaults for SA z/OS (AOFEXDEF)	171
Setting the region size for NetView to 2 GB	172
Sending UNIX messages to the syslog	172
Adapting the SA z/OS best practices policy for SAP	172
Overview of the resources	173
Description of the group structure	174
SAP system dependent groups	174
SAP infrastructure group	175
ABAP central services and enqueue replication server	177
Dependencies between the ABAP enqueue server and the enqueue replication server	178
Optional components of the ABAP central services	180
JAVA central services and enqueue replication server	180
DB2 policy	181
DB2 database server group (SAPHA1_DBX)	182
Classes	183
C_SAP_USS	183
SAP application servers	183
APAP-only application server	183
Double-stack (ABAP plus Java) application server	184
Java-only application server	185
SAP Application server groups	187
SAPHA1RAS	187
SAPHA1RASX	187
Overview of groups/applications.	189
Adding entries to the Automation Table	190
Adding the definitions for extension DFS/SMB	191
Additions to the SA z/OS policy	191
Application	191
DFS_SMB	191
Application group.	192
SMB_PLEX	192
Additions to the Automation Table for DFS/SMB	192

Chapter 12. Customizing the Tivoli System

Automation for Multiplatforms (Base)	195
Introduction	195
Overview of Tivoli System Automation for Multiplatforms	196
Installing SA MP	197
Setting up SA MP cluster to manage SAP resources	199
Installing the high availability policy for SAP	199
Implementing Option 1A (Variant A)	201
Customizing the HA policy for a double-stack or Java-only SAP system	203

Step 1: Adapt the sample ABAP and Java configuration files	203
Step 2: Run the mksap script to create SA MP resources	205
Step 3: Perform a quick test of the SAP policy	206
Step 4: Save the policy	207
Step 5: Verify your SAP installation running under SA MP control	207
Implementing Option 1B (Variant B).	207
Making NFS highly available via SA MP	208
Run the NFS server in its own cluster	208
Special considerations for AIX.	209
Customize the sample SA MP high availability policy for SAP	209
Creating the SA MP resources for a SAP system	213
Customizing the high availability policy for a double-stack or Java-only SAP system	215
Step 1: Adapt the sample ABAP and Java configuration files	215
Step 2: Run the mksap script to create SA MP resources for components A, J and I	217
Step 3: Adapt the dependency file to create dependencies to the NFS server	219
Step 4: Run the commands in the dependency file	219
Step 5: Perform a quick test of the SAP policy	221
Step 6: Save the policy	221
Verify your highly available double-stack installation	221
Starting the SAP system.	222
Verifying your high availability implementation with SA MP	223
Removing the HA policy	223
Using a tie breaker with SA MP	224
Using SA MP Quorums with Linux on System z under z/VM	225

Chapter 13. Customizing the Tivoli System

Automation Application Manager (E2E)	227
Overview of end-to-end automation management	227
Sample high availability environment of the SAP on System z solution	228
Setting up the end-to-end product	230
Defining and installing the end-to-end high availability policy for SAP	237

Chapter 14. Change management

Updating the SAP kernel	241
Updating the SAP kernel (release 6.40 or later)	242
Updating the enqueue server or replication server, or changing the size of the enqueue table	242

Applying SAP/other maintenance when SAP is controlled by SA MP	243
Rolling kernel upgrade	243
Rolling update of DB2 Connect	244
Updating DB2 or z/OS	245

Chapter 11. Customizing Tivoli System Automation for z/OS

This chapter shows you how to set up Tivoli System Automation for z/OS (referred to here as SA z/OS) for the high availability solution for SAP.

Notes:

1. A detailed knowledge of SA z/OS is required to make SAP high availability work.
2. Throughout this chapter, the sample SAP system-ID (SAPSID) of **HA1** is used. When setting up your SA policy, ensure you replace HA1 with the SAPSID of *your own SAP system*.

This chapter contains these main topics:

- “Preparing SA z/OS for SAP high availability”
- “Adapting the SA z/OS best practices policy for SAP” on page 172
- “Overview of groups/applications” on page 189
- “Adding entries to the Automation Table” on page 190
- “Adding the definitions for extension DFS/SMB” on page 191

Preparing SA z/OS for SAP high availability

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

Before you start

If you have not already done so, refer to “Setup of Tivoli NetView and Tivoli System Automation for z/OS” on page 132. Verify the following:

- NetView is customized and running.
- SA z/OS is customized and running.
- Automated Restart Manager (ARM) does not interfere with SA z/OS.
- Either the NetView Management Console (NMC) or the Status Display Facility (SDF) is customized and working.
- You can stop and start all “base” z/OS subsystems (such as JES, VTAM, or TCP) using SA z/OS.

Setting initialization defaults for SA z/OS (AOFEXDEF)

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

- AOFRESTARTALWAYS = 0

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

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

- AOFUSSWAIT = 30

AOFUSSWAIT is the time SA z/OS waits for the completion of a user-specified z/OS UNIX monitoring routine (defined in the z/OS UNIX Control Specification

Customizing Tivoli System Automation for z/OS

panel) until it gets a timeout. When the timeout occurs, SA z/OS no longer waits for the response from the monitoring routine and sends a SIGKILL to that routine.

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

For details, refer to the sections “How to Automate USS Resources” (‘USS’ is an abbreviation for UNIX System Services) and “Global Variables” in the SA z/OS publication *Customizing and Programming*.

Setting the region size for NetView to 2 GB

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

```
//HSAAPPL PROC PROG=DSIMNT, ** PGM USED TO START NETVIEW  
// REG=0, ** REGION SIZE(IN M) FOR NETVIEW
```

Sending UNIX messages to the syslog

To send UNIX messages to the syslog, you must:

1. Ensure that you have a running USS 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 send UNIX syslogd messages to the z/OS syslog, add the following entry to your syslog configuration file `/etc/syslog.conf` (or other specified on the start of the syslogd with the `-f` option):

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

UNIX messages will then appear 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 for SAP.

Adapting the SA z/OS best practices policy for SAP

To use this manual, you must have installed and activated the z/OS SA APAR OA26776. This APAR includes the SAP “Best Practice” policy version, which we assume throughout this chapter. The SAP policy comes as a predefined add-on policy which must be customized and imported thru the SA z/OS customization dialog. The start-, stop-, and monitor-scripts used with this “Best Practice” policy are based on and contained in the file `SAP_v8.zip`. It can be downloaded from:

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

The “Best Practice” policy (and all the diagrams and samples used in this book) refer to a sample SAP system with a SAP system ID (**SAPSID**) of **HA1**. When you adapt the policy to your SAP system, we strongly recommend that you follow the naming conventions that are described in “Conventions used in the SA z/OS policy” on page 124.

Adapting the “Best Practice” SAP policy means:

- During import of the selected and customized SAP policy parts rename the resources and resource groups according to your SAPSID and the above mentioned naming conventions.

- If you do not have an OSPF based (dynamic routing based) highly available network between Appl. server and DB server, then you must replace the HASPARENT relationship of the SAP and NFS Server resources to the OMROUTE application by a HASPARENT to TCPIP application.
- For each SAP resource you have to check and/or adapt:
 - Under Application Information:
 - the Subsystem Name
 - the Job Name
 - JCL Procedure Name for the VIPA definition
 - Startup Parameters: VIPA='10.101.5.194' for the VIPA definition
 - Under Startup:
 - the Startup commands
 - the Poststart commands
 - Under Shutdown:
 - the Shutforce command for 'remote Application Server'?
 - Under USS control:
 - the User-ID
 - the Command/Path

ABAP and Java SCS are managed by the sapctrl_em script. Sapctrl_em is used to start, stop, and check the resources of the ABAP and Java SCS instances MS, ES, ERS, CO, SE, GW. Please see Appendix E Automation Scripts for details and invocation syntax.

Remote ABAP Application servers are managed by the scripts: startappsrv_v5, checkappsrv_v4, stopappsrv_v5.

Remote Java Application servers are managed by the scripts: startjappsrv, checkjappsrv, stopappsrv_v5.

For details and invocation syntax, see Chapter 10, "Customizing SAP for high availability," on page 135.

Overview of the resources

The following SAP-related components are defined in the SA z/OS policy. Not all of the components may be needed for your specific environment. For example, for an ABAP-only SAP system you will *not* need the JAVA components. So the first step in adapting the policy for a specific SAP system will consist of determining which parts of the policy you actually want to exploit.

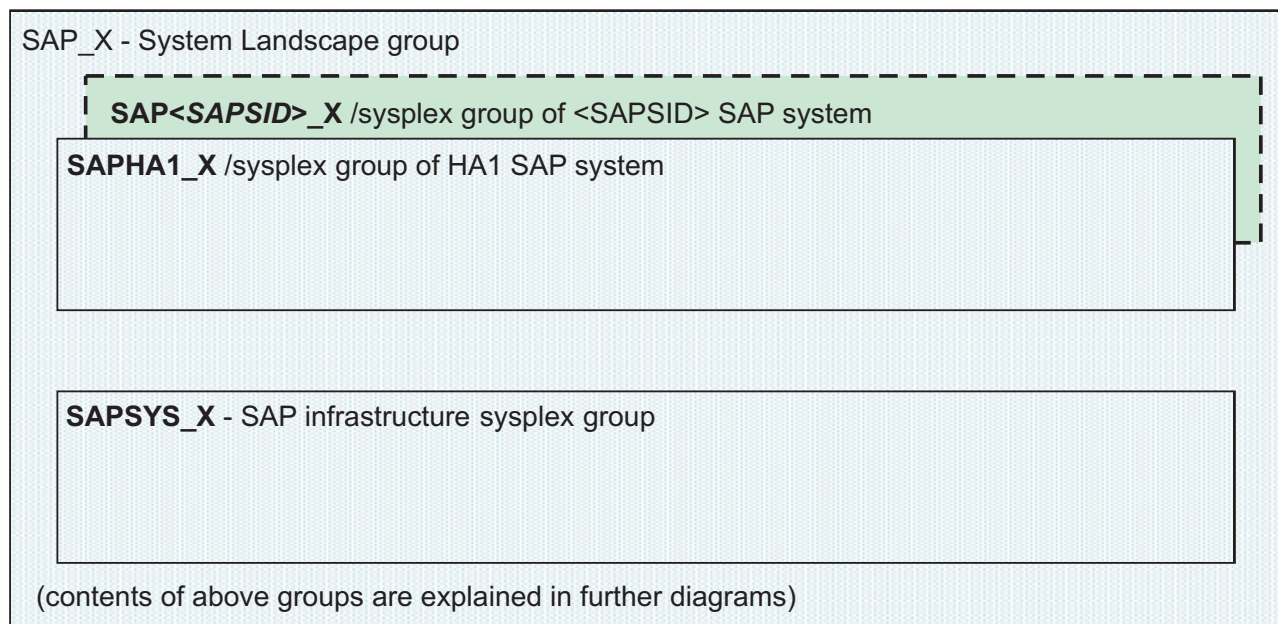
- Resources that are related to a specific SAP system (in our case, HA1):
 - DB2 z/OS resources
 - ABAP Central services - including enqueue server and message server, and the associated VIPA.
 - the optional components SAP gateway, syslog collector, and syslog sender
 - ABAP Enqueue replication server
 - JAVA Central services - including enqueue server and message server, and the associated VIPA
 - Remote SAP application servers (*not* shipped with the SAP best practice policy!)
- Resources that are common to all the SAP systems:
 - NFS server
 - SapRouter and/or SAP Web Dispatcher

- saposcol and sapccmsr

Description of the group structure

The top level group in the best practice policy that contains all the components listed in “Overview of the resources” on page 173 is **SAP_X**. Figure 39 shows the SAP_X group, which contains:

- SAPHA1_X** The group containing specific groups and resources for SAP system HA1.
- SAPSYS_X** The group containing all SAP-system-independent groups and resources.




- indicates optional resources / optional groups
-  Sysplex/Basic active group

Figure 39. SA z/OS best practices policy for SAP

Optionally, SAP_X can also contain one or more **SAP<SAPSID>_X** groups. These contain the resources needed for other SAP systems.

Note: You can use the SAP_X group for easy monitoring of the availability of your *complete SAP landscape*. Providing this group is available, your SAP systems will also be available.

SAP system dependent groups

Each SAP system dependent group contains all the resources needed for the operation of a certain SAP system. The general naming convention that we recommend is **SAP<SAPSID>_X**. This allows you to easily filter on the NMC or define SDF panels for monitoring the resources required by a specific SAP system.

Figure 40 shows the details of the **SAPHA1_X** group. This SAPSID-specific group comprises the:

- ABAP enqueue and message server group (SAPHA1AENQX).
- ABAP enqueue replication server group (SAPHA1AER_X).
- JAVA enqueue and message server group (SAPHA1JENQX).
- JAVA enqueue replication server group (SAPHA1JER_X).
- DB2 database server group (SAPHA1_DBX).

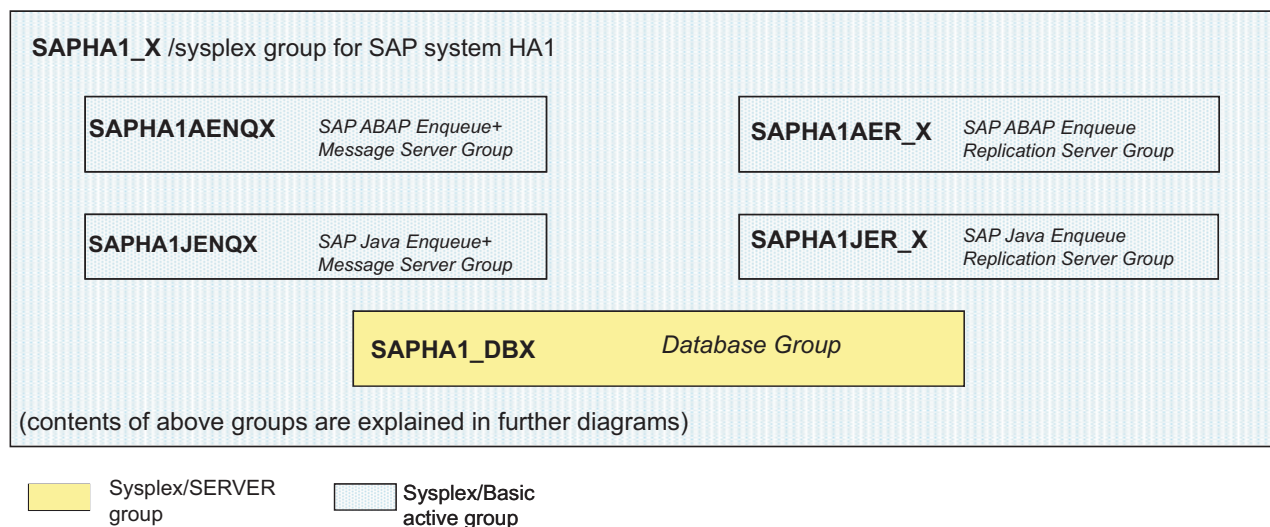


Figure 40. Group **SAPHA1_X** belonging to SAP system HA1

SAP infrastructure group

The SAP infrastructure group **SAPSYS_X** contains all groups which are needed for the operation of the SAP resources contained in all SAP<SAPSID>_X groups. If you use our recommended setup of running the NFS server on z/OS, the availability of this NFS server is of high importance to the operation of *all SAP systems*.

Figure 41 on page 176 shows the details of the SAP-system independent groups and resources. It contains these groups:

- The NFS server group (NFS_SERV_X).
- The group required to perform SAP monitoring (SAPSYSOSC_X).
- An optional group for the SAP router (SAPSYSRTE_X).

Customizing Tivoli System Automation for z/OS

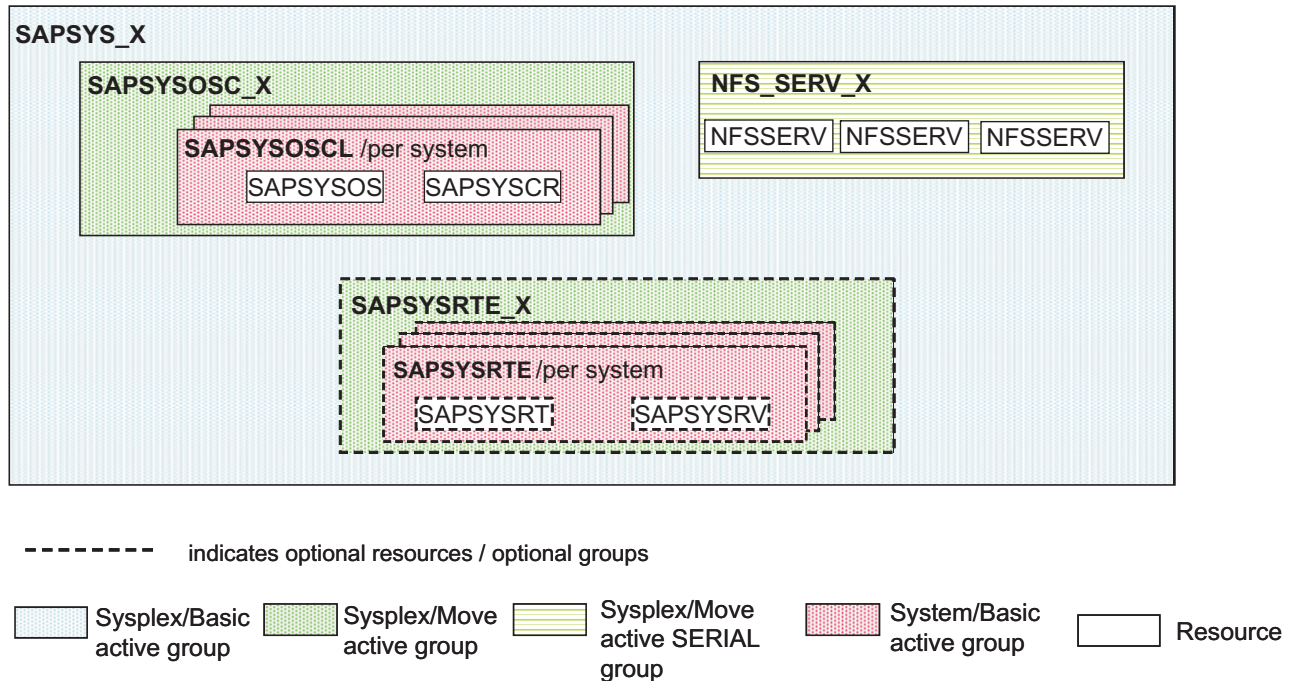


Figure 41. SAP-system-independent groups and resources

Further details of Figure 41:

- The NFS server that is needed for the SAP infrastructure can only be active 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.
- To enable SAP monitoring on z/OS, we recommend that you have only *one* SAP operating system collector *saposcol* and *one* SAP CCMS Agent *sapccmsr* running per SYSPLEX.
- Both *saposcol* and *sapccmsr* must be running on the *same LPAR*. This is accomplished by defining two SA applications SAPSYSOS and SAPSYSOCR, which correspond to the SAP's *saposcol* and *sapccmsr* agents. Both are grouped together in a SYSTEM group SAPSYSOSCL.
- Since there must be only one pair of *saposcol* and *sapccmsr* running in the SYSPLEX at a time, we define the SYSPLEX MOVE group SAPSYSOSC_X containing the SAPSYSOSCL group as shown in Figure 41.
- For further information on SAP Monitoring and how to set up *saposcol* and *sapccmsr*, refer to the chapter "Monitoring a Standalone Database" in the SAP document *SAP NetWeaver Guides: Monitoring Setup Guide*.
- For the required setup of *saposcol* and *sapccmsr*, see Chapter 10, "Customizing SAP for high availability," on page 135.
- If you decide to run your SAP router on z/OS, you also have to define a VIPA to be used in accessing the SAP router. The SAP router and its associated VIPA must run together on the same LPAR. Therefore, the policy defines two SA applications SAPSYSRTE and SAPSYSRVR which correspond to the SAP router and its associated VIPA. Both are grouped together in a SYSTEM group SAPSYSRTE_G. Since there must only be one active SAP router and its associated VIPA in the SYSPLEX at a time, we define the SYSPLEX/MOVE group SAPSYSRTE_X containing SAPSYSRTE (as shown in Figure 41).
- If you decide to run your SAP Web Dispatcher on z/OS, the same prerequisites and definitions apply as for the SAP Router: own VIPA and two SA applications

SAPSYSWD and SAPSYSWV which correspond to the SAP Web Dispatcher and its associated VIPA. Both are grouped together in a SYSTEM group SAPSYSWD_G. You should use the SAP Router resources of the “best practices” policy as a template.

ABAP central services and enqueue replication server

In this section, we describe 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 180) are very similar to this one.

Figure 42 shows the lowest level in the group structure of the ABAP central services and enqueue replication server.

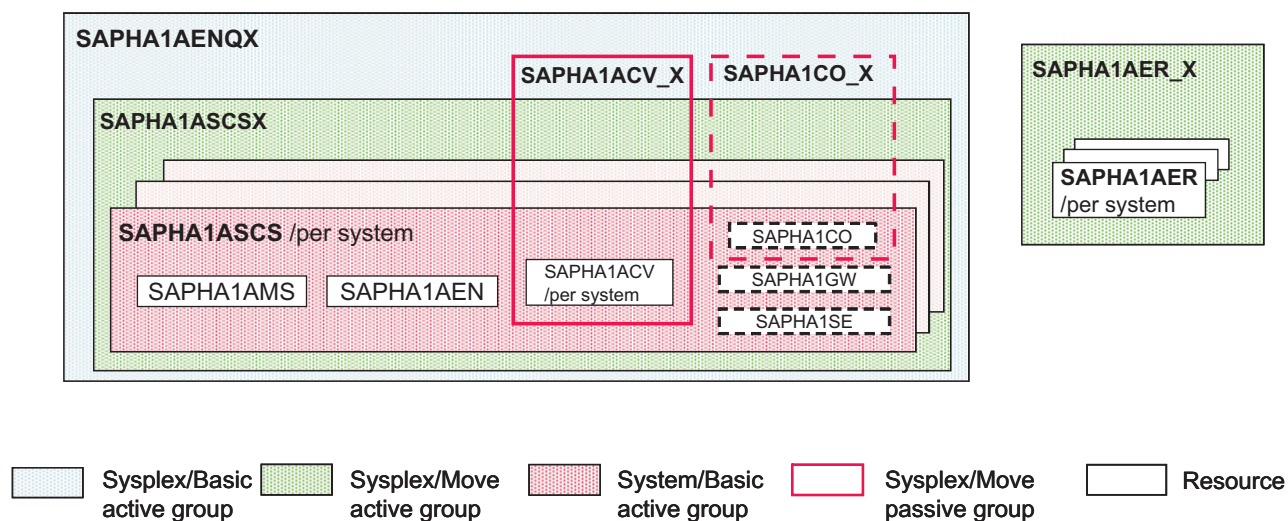


Figure 42. Lowest level in group structure of ABAP central services and enqueue replication server

SAPHA1AENQX of Figure 42 contains these resources:

- ABAP enqueue server (SAPHA1AEM).
- ABAP message server (SAPHA1AMS).
- The VIPA associated with the ABAP enqueue server (SAPHA1ACV).
- The *optional* resources that might be placed in this group:
 - The SAP SYSLOG collector (SAPHA1CO).
 - The SAP SYSLOG sender (SAPHA1SE).
 - The SAP gateway (SAPHA1GW).

SAPHA1AER_X of Figure 42 contains the ABAP enqueue replication server resource SAPHA1AER.

Further details of Figure 42:

- The *threshold definition* for the SAPHA1EM application is as follows:

Minor Resource Name	
Critical Number	1
Critical Interval	01:00
Frequent Number	1
Frequent Interval	02:00
Infrequent Number	1
Infrequent Interval	12:00

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

- The VIPA resource SAPHA1ACV (in addition to being member of the SAPHA1AENQX) is also member of the SYSPLEX MOVE PASSIVE group SAPHA1ACV_X. Its purpose is to define a relationship between the enqueue server its VIPA and the enqueue replication server. The relationships of SAPHA1AER_X are as follows:

Relationship Type	MAKEAVAILABLE
Supporting Resource	SAPHA1ACV_X/APG
Description	Place replicator after central services
Sequence Number	
Automation	PASSIVE
Chaining	WEAK
Condition	WhenAvailable

This ensures that the INGGROUP command (described in Table 17 on page 179) in the application automation definitions of the SAPHA1ACV resource is processed by SA z/OS prior to the decision where to place the enqueue replication server. Since SAPHA1ACV_X is a MOVE group, only *one* VIPA is started or is active in the SYSPLEX at one time.

- The SAP enqueue replication server resource SAPHA1AER is a member of the SYSPLEX MOVE group SAPHA1AER_X. This ensures that only one enqueue replication server is started or is active in the SYSPLEX at one time. For a 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.”
- The SAP Syslog collector resource (in addition to being member of the SAPHA1AENQX) is also a member of the SYSPLEX MOVE PASSIVE group SAPHA1CO_X. The purpose of this additional group is to ensure that only one SAP syslog collector is started or active in the SYSPLEX at one time.

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 may run:

- During normal operations, the enqueue server and the enqueue replication server 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 is the enqueue server then able to 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 above restrictions are implemented in the *SAP policy definition*. Via definitions within this policy, we 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.

Implementation of Dependency 1: The INGGROUP commands in the “Messages and User data” section of the policy definition for the SAPHA1ACV 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 (SAPHA1AER) and the LPAR where the VIPA for the enqueue server (SAPHA1ACV) is running.

Table 17. Messages and User Data section from the SAPHA1ACV policy definition

Message ID	Command Text
ACORESTART	INGGROUP SAPHA1AER_X/APG,ACTION=ADJUST,MEMBERS=(SAPHA1AER/APL/&SYSNAME.),PREF=(1)
RUNNING	INGGROUP SAPHA1AER_X/APG,ACTION=RESET INGGROUP SAPHA1AER_X/APG,ACTION=ADJUST,MEMBERS=(SAPHA1AER/APL/&SYSNAME.),PREF=(1)

Implementation of Dependency 2: The INGGROUP commands in the policy startup POSTSTART definitions of the SAPHA1AER 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 the SAPHA1ASCS and the LPAR where the enqueue replication server (SAPHA1AER) is running.

Table 18. Startup section from the SAPHA1AER policy definition

Message ID	Command Text
POSTSTART	INGGROUP SAPHA1ASCS_X/APG,ACTION=RESET INGGROUP SAPHA1ASCS_X/APG,ACTION=ADJUST,MEMBERS=(SAPHA1ASCS/APG/&SYSNAME.),PREF=(700)

Implementation of Dependency 3: The MAKEAVAILABLE WhenObservedSoftDown relationship against SAPHA1ASCS/APG/= will prevent the start of the enqueue replication server (SAPHA1AER) whenever the ABAP central services group SAPHA1ASCS on the same system is in HARDDOWN status.

Table 19. Relationships section from the SAPHA1AER policy definition

Relationship Type	Supporting Resource	Automation	Chaining	Condition
MAKEAVAILABLE	SAPHA1ASCS/APG/=	PASSIVE	WEAK	WhenObservedSoftDown

Consequences of this implementation: An SA operator has to *manually* change the status of the failed resource(s) in the group to AUTODOWN (after he/she has investigated/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 (not included in the best practices policy): In a two-LPAR environment, this may prevent the enqueue replication server from restarting at all. You may want to set a BROKEN enqueue server to AUTODOWN as soon as it is restarted on the other system, in order to allow the enqueue replication server to restart. This can be done by following changes to the SAPHA1AER definition:

1. Remove the 'MAKEAVAILABLE/WhenObservedSoftDown' relationship to SAPHA1ASCS/APG/=.
2. Add to the list of POSTSTART commands: SETSTATE SAPHA1AEN,AUTODOWN.

Note that one possible consequence of using SA to automatically reset the enqueue server (SAPHA1AEN) status instead of letting an SA Operator do it manually is that the enqueue server may 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 moves it to LPAR2. There the enqueue server fails again.
3. SA 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 SAPHA1AER resource accordingly.

Optional components of the ABAP central services

One of the optional resources that might be placed in the ABAP central services group (SAPHA1AENQX) is the *SAP syslog collector* (SAPHA1CO).

SAPHA1CO_X is another SYSPLEX/MOVE PASSIVE group with the only member being the SAP syslog collector (SAPHA1CO). Its SERIAL attribute ensures that *only one* SAP syslog collector daemon is started or is active in the SYSPLEX at one time.

JAVA central services and enqueue replication server

This section describes the policy definition for the:

- JAVA central services, and
- (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 178). Therefore, for detailed explanations you should refer to the ABAP central services description (simply replacing the resource names in the explanation).

Figure 43 on page 181 shows the lowest level in the group structure of the JAVA central services and enqueue replication groups:

- SAPHA1JENQX contains the resources:
 - JAVA enqueue server (SAPHA1JEM).
 - JAVA message server (SAPHA1JMS).
 - VIPA (SAPHA1JCV).
- SAPHA1JER_X contains the JAVA enqueue replication server resource SAPHA1JER.

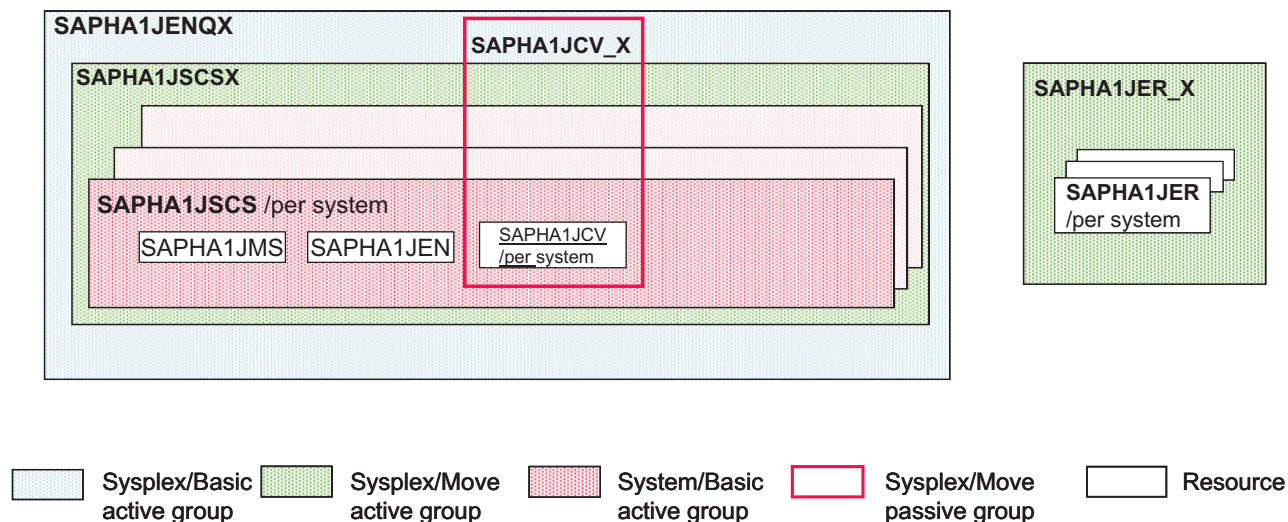


Figure 43. Lowest level in group structure of JAVA central services and enqueue replication groups

DB2 policy

To use this manual, you must have installed and activated the z/OS SA APAR OA26776.

APAR OA26776 includes the DB2 Best Practice Policy version in which a *DB2 LIGHT restart* is performed by SA z/OS.

Figure 44 on page 182 shows how IBM exploits SA's DB2 "Best Practise" policy within the test environment.

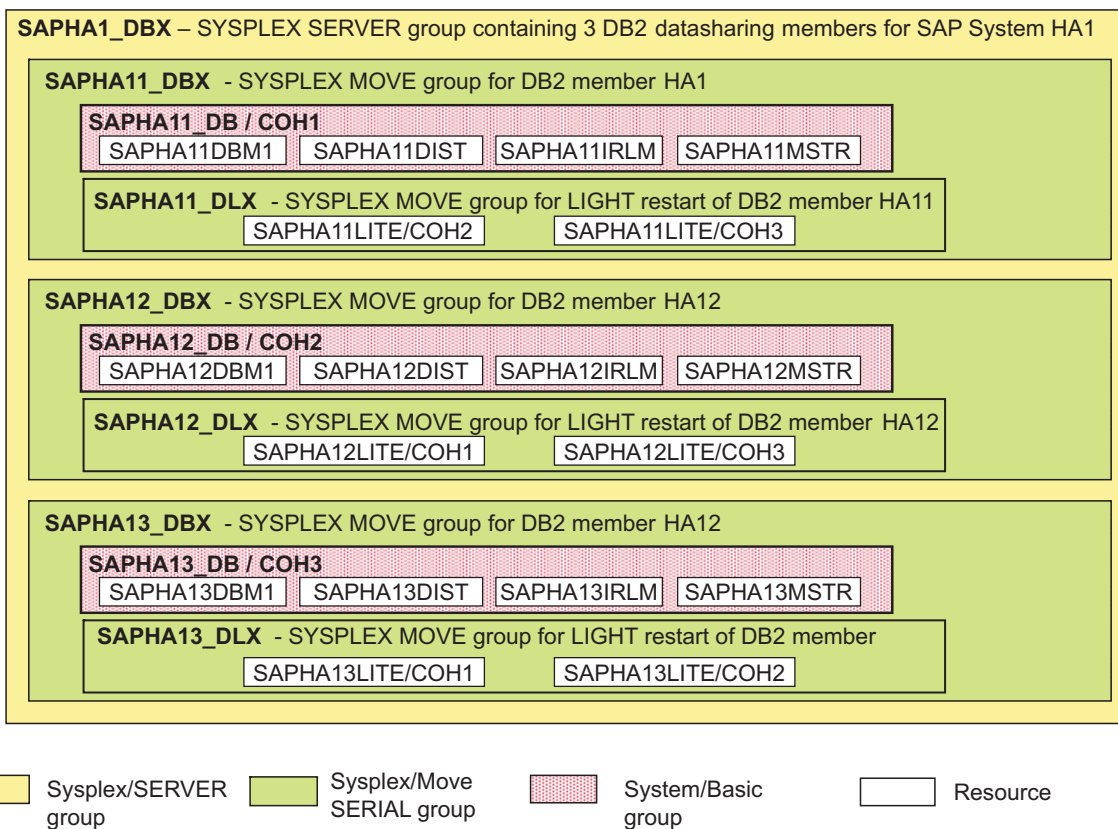


Figure 44. DB2 Best Practice Policy – adapted for SAP system HA1

DB2 database server group (SAPHA1_DBX)

In the sample policy, the following three DB2 members are defined for the sample SAP system **HA1**:

- HA11 running on COH1.
- HA12 running on COH2.
- HA13 running on COH3.

Each of these members is represented by a SYSPLEX/MOVE named SAPHA1<n>_DBX. 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 also implemented in the DB2 policy in form of the SYSPLEX/MOVE groups SAPHA1<n>_DLX. For a more detailed explanation of the relationships mechanisms inside the DB2 policy, refer to the relevant SA z/OS documentation.

For your environment you need to 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 121.

Classes

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

For SAP, we use one class only, the default UNIX System Services class (C_SAP_USS).

C_SAP_USS

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

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

SAP application servers

In this section, we provide the definitions of the *remote* application servers. With SAP NetWeaver 04s, there are *three* types:

- an ABAP-only SAP application server
- a double-stack (ABAP plus Java) SAP application server
- a Java-only SAP application server

These three types are now described.

ABAP-only application server

We start with the definitions of an ABAP-only application server (AS). The SA Application is called **SAPHA1ACI**. It is the rest of the remote ABAP Central Instance under Linux on System z. This is a USS 'proxy' resource.

Because the application server runs on a remote Linux system, it can not be "seen" by SA z/OS. When started/stopped via TCP/IP based tools, the only indication for an "up and running" status is the response of the monitor routine.

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

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

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

- One SHUTFORCE command, which forces the SAP application server to stop running. The <InstType> parameter of startjappsrv must be '2' for a Java-only AS.

Definition:

Entry Name: SAPHA1CI
Link to Class: C_SAP_USS

```
Application Information
Application Type. . . USS
Subsystem Name. . . . SAPHA1CI
Job Type. . . . . NONMVS
Job Name. . . . . SAPHA1CI
Start Timeout . . . . 00:08:00
Shutdown Pass Interval 00:05:00
```

Customizing Tivoli System Automation for z/OS

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

Startup STARTUP
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/ha1adm/startappsrv_v5 ih1scoh1
02 DVEBMGS02 SSH >& /u/ha1adm/startappsrv_02.ih1scoh1.log'

Startup POSTSTART
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/ha1adm/checkappsrv_v4 ih1scoh1
02 >& /u/ha1adm/checkappsrv_02.ih1scoh1.log'

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

Shutdown SHUTIMMED
1
INGUSS /bin/kill -2 %PID%

Shutdown FORCE
1
INGUSS /bin/tcsh -c '/u/ha1adm/stopappsrv_v5 ih1scoh1 02 DVEBMGS02 SSH
>& /u/ha1adm/stopappsrv_02.ih1scoh1.log'

USS Control
User ID. . . . . HA1ADM
Command/Path . . . . ./rfcping_ih1scoh1_02
```

Important Notes:

- a) The startappsrv_v5 and checkappsrv_v4 scripts must be adapted.
- b) The startappsrv_v5 ends by default after 120 seconds if the 'rfcping' monitor was not up and running. If starting the Application server takes longer in your environment, you must adapt the script accordingly.

Double-stack (ABAP plus Java) application server

A *double-stack* application server is physically one SAP instance which runs both the ABAP and the Java stack.

Although it is physically one instance, and both stacks are started by default when the (ABAP) instance starts, the SA z/OS policy separates it into its *two* logical parts:

- an ABAP AS
- a Java AS

This means that within SA, one “double-stack” (or Add-In) 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 those two logical application servers:

- The Java instance must only be *started* after the ABAP instance is active. So there is a HasParent relationship between both. This HasParent relationship ensures that starting the Java instance *automatically* triggers the previous start of the ABAP instance. Then “starting” simply means just waiting until Java is up.
- On the other hand, *stopping* the Java application server does *not* stop any of the Java server processes. It only stops the monitoring java program *GetWebPage*, which does a primitive health check of the Java application server.

The definitions for the *logical ABAP AS part* of a ‘double-stack’ instance are similar to those for an ABAP-only AS (listed in “APAP-only application server” on page 183).

Listed below are the definitions for the *logical Java AS part* of a ‘double-stack’ instance. The SA Application is called **SAPHA1JCI**. It is the remote Java Central Instance under *Linux on System z* (belonging to the ‘rest’ remote ABAP CI). This is also a USS ‘proxy’ resource.

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

Note: The <InstType> parameter of startjappsrv must be ‘1’ for this double-stack AS.

Entry Name: SAPHA1JCI
Link to Class: C_SAP_USS

Application Information

Application Type. . . USS
Subsystem Name. . . . SAPHA1JI
Job Type. NONMVS
Job Name. SAPHA1JI
Start Timeout 00:08:00
Shutdown Pass Interval 00:05:00

Relationships

Relationship Type. . HASPARENT
Supporting Resource. SAPHA1CI/APL/=

Startup STARTUP

INGUSS JOBNAME=&SUBSJOB,/bin/sh -L -c '/u/haladm/startjappsrv ihlscoh1
02 DVEBMGS02 1 SSH > /u/haladm/startjappsrv_02.ihlscoh1.log 2>&1'

Startup POSTSTART

INGUSS JOBNAME=&SUBSJOB,/bin/sh -L -c '/u/haladm/checkjappsrv ihlscoh1
02 > /u/haladm/checkjappsrv_02.ihlscoh1.log 2>&1'

Shutdown NORM

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

USS Control

User ID. HA1ADM
Command/Path/javaexe_ihlscoh1_02

Important Notes:

- a) The startjappsrv and checkjappsrv scripts must be adapted.
- b) The startjappsrv ends by default after 240 seconds, if the 'GetWebPage' monitor was not up and running. This time starts after the ABAP AS is up. If starting the Java Application server takes longer in your environment, you must adapt the script accordingly.

Java-only application server

Here are the definitions for a *Java-only AS*. The SA Application is called **SAPHA1J1**. This is a USS ‘proxy’ resource.

Because the application server runs on a *remote Linux system*, it can not be “seen” by SA z/OS.

Customizing Tivoli System Automation for z/OS

When started/stopped via TCP/IP based tools, the only indication for an “up and running” status is the response of the monitor routine. The same mechanism applies to a Java-only AS as for an ABAP-only AS.

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

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

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

- One SHUTFORCE command, which forces the SAP application server to stop running.

Entry Name: SAPHA1J1
Link to Class: C_SAP_USS

Application Information
Application Type. . . USS
Subsystem Name. SAPHA1J1
Job Type. NONMVS
Job Name. SAPHA1J1
Start Timeout 00:08:00
Shutdown Pass Interval 00:05:00

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

Startup STARTUP
INGUSS JOBNAME=&SUBSJOB,/bin/sh -L -c '/u/ha1adm/startjappsrv ihlscoh2
94 JC94 2 SSH > /u/ha1adm/startjappsrv_94.ihlscoh2.log 2>&1'

Startup POSTSTART
INGUSS JOBNAME=&SUBSJOB,/bin/sh -L -c '/u/ha1adm/checkjappsrv ihlscoh2
94 > /u/ha1adm/checkjappsrv_94.ihlscoh2.log 2>&1'

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

Shutdown SHUTIMMED
1
INGUSS /bin/kill -2 %PID%

Shutdown FORCE
1
INGUSS /bin/tcsh -c '/u/ha1adm/stopappsrv_v5 ihlscoh2 94 JC94 SSH >&
> /u/ha1adm/stopappsrv_94.ihlscoh2.log'

USS Control
User ID. HA1ADM
Command/Path/javaexe_ihlscoh2_94

Important Notes:

- a) The startjappsrv and checkjappsrv scripts must be adapted.
- b) The startjappsrv ends by default after 240 seconds, if the 'GetWebPage' monitor was not up and running. If starting the Java Application server takes longer in your environment, you must adapt the script accordingly.

SAP Application server groups

Having defined the SA 'proxy' Application Server resources, you have to create an application group to combine the remote SAP application servers.

Here we have only one remote double-stack AS in our configuration, but you can add further remote applications servers as required. Then create a superior group at the sysplex level for the proxy AS.

SAPHA1RAS: This application group is created to combine the remote SAP application servers, although we have only one remote application server.

Definition:

```
Entry Type: ApplicationGroup
Entry Name: SAPHA1RAS
Application Group Type . SYSTEM
Nature . . . . . BASIC
```

Select applications:

```
SAPHA1CI
SAPHA1JCI
```

Our sample policy does not contain an explicit MAKEAVAILABLE/WhenAvailable relationship between the remote SAP application servers and DB2. There are two reasons for this:

1. The first step during startup of an application server is to test the database connection. So this 'function' is already part of the SAP AS start procedure.
2. The possibility exists to create a sysplex server group containing all DB2 sysplex members (of the SAP system) with an availability goal of '1'. You can then add a *MAKEAVAILABE/WhenAvailable* relationship between the SAPHA1RAS group and this DB2 group. This means, an AS can start as soon as one DB2 member is up. However, there is no check that the definitions for DB2 connection failover (the connect.ini entries for ABAP for example) are consistent with these SA definitions. It is nearly impossible to detect and identify problems caused by such inconsistent definitions. Therefore, we *we do not recommend* such an SA policy extension.

SAPHA1RASX: This application group is a SYSPLEX/MOVE group defined for the proxy SAP application servers. These application servers are running on remote systems like Linux, AIX or Windows. They are monitored by SA z/OS on one LPAR only, as shown in Figure 45 on page 188 (active SA applications are represented as shaded boxes). If the LPAR has to be stopped, only the monitoring of the servers is moved via the MOVE group. The application servers themselves will not be stopped.

The application group SAPHA1RASX needs a HASPARENT relationship to the NFS sysplex group, because the application server executables reside on the NFS. If the NFS server is moved, the application servers are not stopped. If NFS is stopped, the application servers must also be stopped.

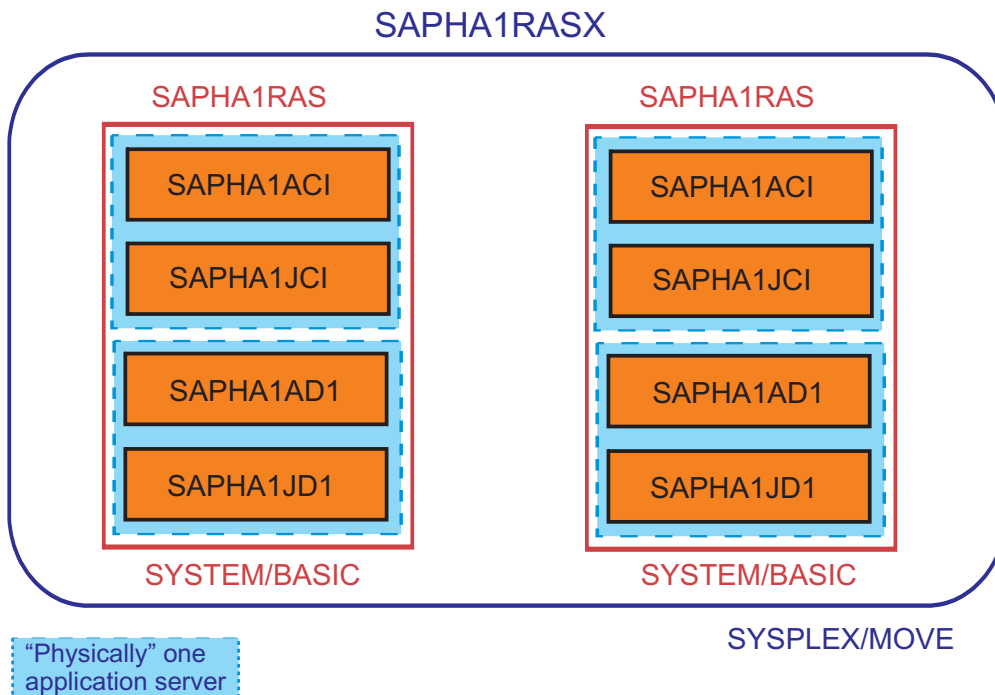


Figure 45. SAPHA1RASX application group

Figure 45 describes a sample that consists of *two* double-stack application servers. The ABAP “remaining CI” is called SAPHA1ACI, and the corresponding java CI is called SAPHA1JCI. The second application server consists of SAPHA1AD1 and SAPHA1JD1.

Definition:

```

Entry Type: ApplicationGroup
Entry Name: SAPHA1RASX
Application Group Type . SYSPLEX
Nature . . . . . MOVE

Select applications:
SAPHA1RAS/APG/COH1
SAPHA1RAS/APG/COH2
SAPHA1RAS/APG/COH2

Relationships
Relationship Type. . HASPARENT
Supporting Resource. NFS_SERV_X/APG
    
```

Overview of groups/applications

Figure 46 gives you the overall picture of all the SA z/OS groups that are contained in the SAP “best practise” policy. To improve readability, SYSTEM/Basic groups are only shown as *one box*, although there exists *one such group per LPAR* in the SYSPLEX.

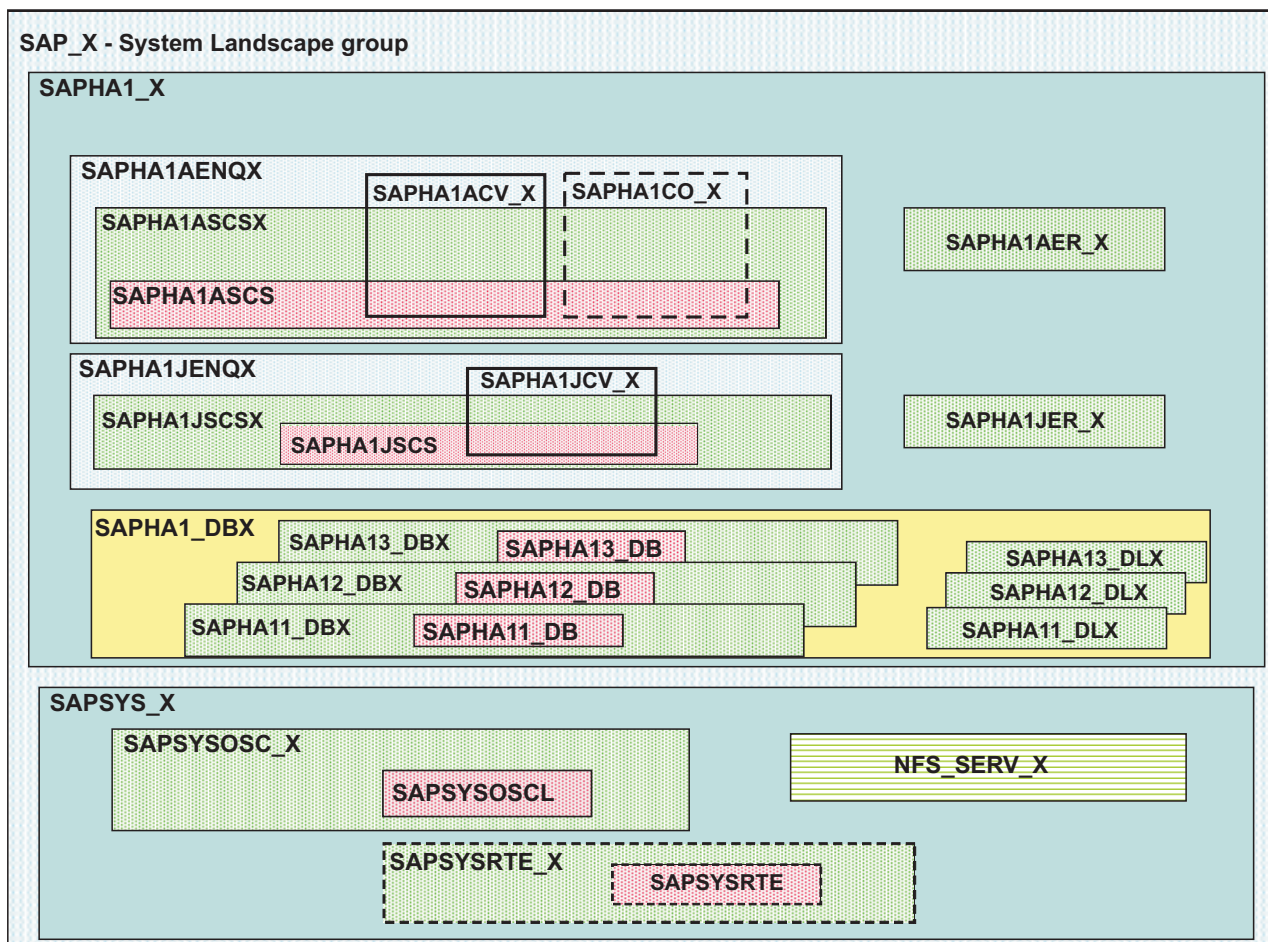


Figure 46. Overview of the resources

Figure 47 on page 190 provides an overview of the policy that is provided for SAP Version 7.

Customizing Tivoli System Automation for z/OS

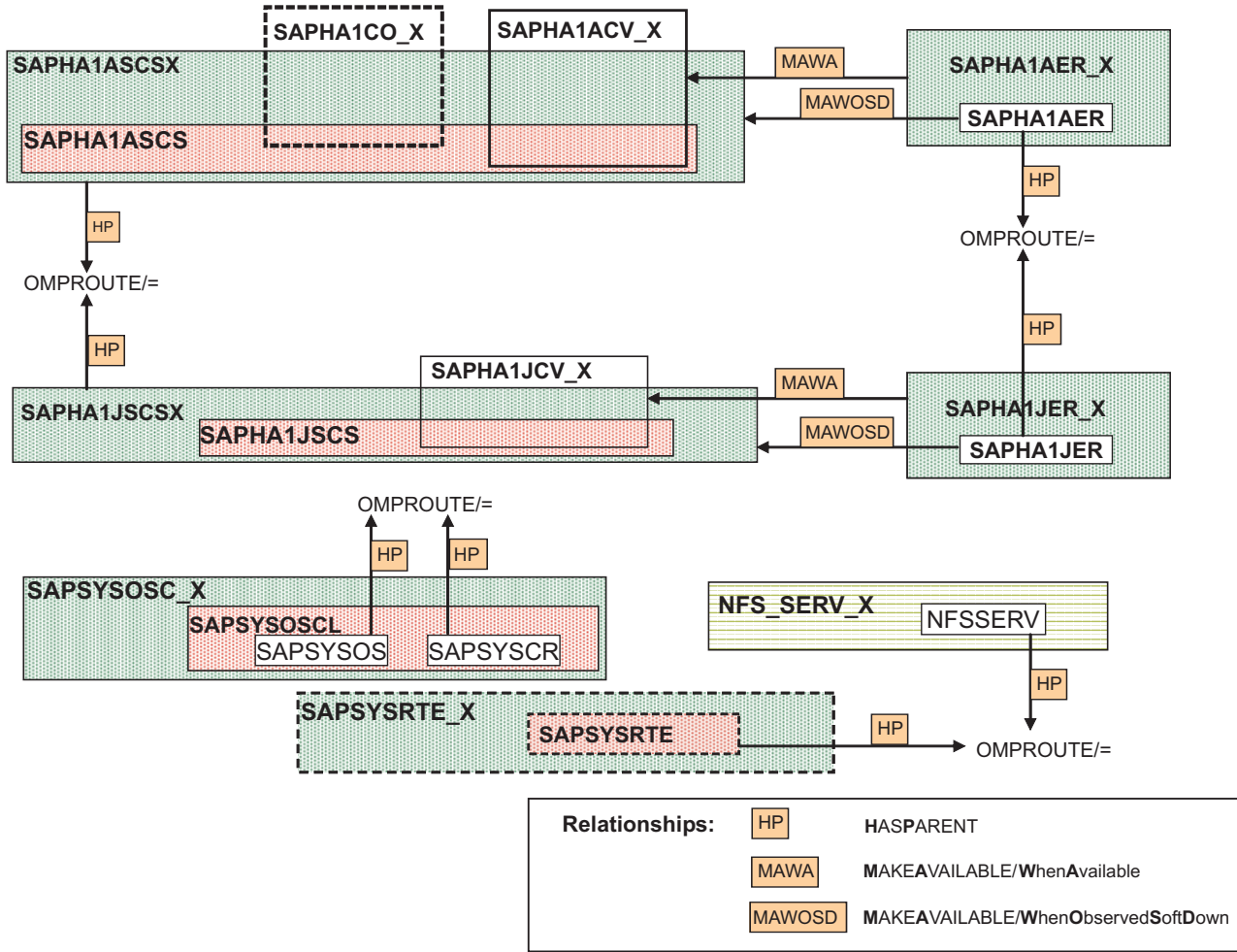


Figure 47. Overview of the relationships between elements of the SAP policy (excluding DB2 elements)

Notes to Figure 47:

- 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 178.

Adding entries to the Automation Table

If you are using “proxy” SA z/OS resources (as shown in Figure 6 on page 18) for “remote” SAP application server, you must enhance the Automation Table to enable NetView to trap a special message and route this message to SA z/OS.

These “remote” SAP application servers are started using either:

- script startappsrv_v5, or
- script startjappsrv

The entry for message BPXF024I traps error messages during startup of these “remote” SAP application server. For further details, see “Sending UNIX messages to the syslog” on page 172.

These are the statements you must insert in any additional automation table. These lines are contained in the file SA_AutomationTableAddition of SAP_v8.zip.

```
*****
*
* SPECIAL SHELL-SCRIPT MESSAGE TO TRAP SAP APPL. SERVER STARTUP ERRORS*
*
*****
IF MSGID = 'BPXF024I' & DOMAINID = '&DOMAIN.' THEN BEGIN;
*
  IF TOKEN(4)='STARTUP' & TOKEN(5)='FAILED'. & TOKEN(3) = JOBN
    THEN EXEC(CMD('TERMMSG JOBNAME=' JOBN ',BREAK=YES,FINAL=YES')
      ROUTE(ONE %AOFOPGSSOPER%));
*
  ALWAYS;
*
END;
*
```

Adding the definitions for extension DFS/SMB

This is an extension to “Adapting the SA z/OS best practices policy for SAP” on page 172. We describe here how to add the definitions for DFS/SMB to the SA z/OS policy to the Automation Table.

Additions to the SA z/OS policy

In this section, we provide the additions to the SA z/OS policy.

Application

We define one application for DFS/SMB.

DFS_SMB: This application corresponds to DFS_SMB.

Definition:

Entry Name:
DFS_SMB

Application Information
Application Type. . . STANDARD
Job Name. DFS_SMB
JCL Procedure Name. . DFS

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

Relationship Type . . MAKEAVAILABLE
Supporting Resource . OMPROUTE/APL/=
Automation. ACTIVE
Chaining. WEAK
Condition WhenAvailable

PRESTART
MVS SETOMVS FILESYS,FILESYSTEM='SAPHA1.SHFS.SAPMNT',SYSNAME=&SYSNAME.

Customizing Tivoli System Automation for z/OS

```
MVS SETOMVS FILESYS,FILESYSTEM='SAPHA1.SHFS.TRANS',SYSNAME=&SYSNAME.  
  
Shutdown NORM  
1  
MVS P &SUBSJOB  
4  
MVS C &SUBSJOB
```

Application group

We define one application group for DFS/SMB.

SMB_PLEX: DFS/SMB should run on one of the two systems at a time. Therefore, we define a SYSPLEX/MOVE group with DFS/SMB, as shown in Figure 48 (active SA applications are represented as shaded boxes).

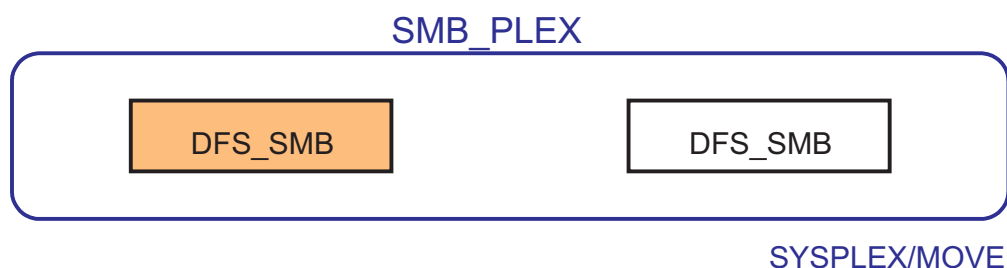


Figure 48. SMB_PLEX application group

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

```
Select applications:  
DFS_SMB
```

We want to have both subsystems MVSNFSSA and DFS_SMB always running on the same LPAR, and want to always move them together; this is why we insert the following STARTUP POSTSTART commands:

For MVSNFSSA:

```
INGGROUP SMB_PLEX/APG,ACTION=RESET  
INGGROUP SMB_PLEX/APG,ACTION=ADJUST,MEMBERS=(DFS_SMB/APL/&SYSNAME.), PREF=(999)
```

For DFS_SMB:

```
INGGROUP NFS_HAPLEX/APG,ACTION=RESET  
INGGROUP NFS_HAPLEX/APG,ACTION=ADJUST,MEMBERS=(MVSNFSSA/APL/&SYSNAME.), PREF=(999)
```

If DFS_SMB moves to a different LPAR, the *POSTSTART* command of DFS_SMB first resets the preference value of the NFS_HAPLEX group to default. Then, it sets the preference value for MVSNFSSA to 999.

This will cause MVSNFSSA to move also to the LPAR on which DFS_SMB is restarted, since the running MVSNFSSA application has a preference value of only 950.

Additions to the Automation Table for DFS/SMB

We define IOEPO1103I as the UP message and IOEPO1100I as the DOWN message for the DFS subsystem:

```

*****
*
* DFS
*
*-----*
*
* IOEP01103I DFS KERNEL INITIALIZATION COMPLETE. ==> UP MESSAGE
*
* IOEP01100I DFS DAEMON DFSKERN HAS STOPPED. ==> FINAL END MESSAGE
*
*****
*
IF MSGID = 'IOEP' . & DOMAINID = '&DOMAIN.' THEN BEGIN;
*
  IF MSGID = 'IOEP01103I' .
    THEN EXEC(CMD('ACTIVMSG UP=YES')
              ROUTE(ONE %AOFOPGSSOPER%));
*
  IF MSGID = 'IOEP01100I' .
    THEN EXEC(CMD('TERMMMSG FINAL=YES')
              ROUTE(ONE %AOFOPGSSOPER%));
*
  ALWAYS;

```

Chapter 12. Customizing the Tivoli System Automation for Multiplatforms (Base)

This chapter describes the implementation and design of the automated and highly available SAP system driven by the base component of IBM Tivoli System Automation for Multiplatforms (SA MP). We provide guidance and recommendations for our high availability strategy with respect to SAP environments. We also discuss practical considerations regarding design and implementation.

This chapter contains these main topics:

- “Introduction”
- “Overview of Tivoli System Automation for Multiplatforms” on page 196
- “Installing SA MP” on page 197
- “Setting up SA MP cluster to manage SAP resources” on page 199
- “Installing the high availability policy for SAP” on page 199
- “Implementing Option 1A (Variant A)” on page 201
- “Implementing Option 1B (Variant B)” on page 207
- “Starting the SAP system” on page 222
- “Verifying your high availability implementation with SA MP” on page 223
- “Removing the HA policy” on page 223
- “Using a tie breaker with SA MP” on page 224
- “Using SA MP Quorums with Linux on System z under z/VM” on page 225

For information on the SA MP end-to-end product, see Chapter 13, “Customizing the Tivoli System Automation Application Manager (E2E),” on page 227.

Introduction

SAP on System z is built around IBM DB2 for z/OS, which is used as the SAP database server. The application logic, written in ABAP/4 or Java, is supported on several platforms. This chapter covers the 64-bit operating systems such as Linux on System z and AIX. Check the SAP Product Availability Matrix (PAM) to determine which platforms are supported. The PAM can be found at <http://service.sap.com/platforms>.

The SAP resources you need to automate and make highly available with SA MP depend on your selected option for your SAP HA environment, which are described in Chapter 2, “Planning overview for high availability for SAP,” on page 13.

- Option 1A: Automation via *SA for z/OS*, *SA for MP*, and *SA End-to-End: Variant A*
- Option 1B: Automation via *SA for z/OS*, *SA for MP*, and *SA End-to-End: Variant B*

Using Option 1A you automate Application server (AS) instances and make SAP utilities with an affinity to those Application servers like SAProuter or SAP Web Dispatcher highly available. Please remember, that we do not recommend moving AS instances from one node to another in the cluster as the SAP architecture allows to run more than one AS, rather run at least 2 AS on different hardware to get the

Customizing the Tivoli System Automation for Multiplatforms (Base)

necessary AS redundancy. AS resources are fixed resources, whereas utilities like SAProuter are floating resources needing their own virtual hostname which will move with those resources. This assumes, that the NFS server and the SAP Central Services are run on z/OS and USS.

Using Option 1B you automate in addition to “Option 1A resources” the SAP Central Services (for ABAP and/or for Java) and the NFS server, if you decided to run the NFS server within the SAP cluster for just this SAP system. You may also run NFS server in its own SA MP cluster serving several SAP systems in the ‘enhanced NFS Server policy’ variant. SAP Central Services as well as the NFS Server are floating resources needing their own virtual hostname which will move with those resources.

In the following sections, first some general information is given. Next, the detailed steps are listed to setup your Option 1A or 1B resources. The necessary steps to set up SA MP are discussed in Installing SA MP.

Extensive testing is required to verify a proper configuration. You can find the verification procedure in “Verification procedure and failover scenarios” on page 289. Appendix E, “Sample Tivoli System Automation for Multiplatforms high availability policy for SAP,” on page 325 describes in detail a sample SA MP policy that defines one SAP system in a three-node cluster. It also describes the SAP processes and how to manage them using the scripts furnished with the policy.

Overview of Tivoli System Automation for Multiplatforms

IBM Tivoli System Automation for Multiplatforms (SA MP) is a product that provides high availability (HA) by automating the control of IT resources such as processes, file systems, IP addresses, and other arbitrary resources in Linux/AIX-based clusters. It facilitates the automatic switching of users, applications, and data from one system to another in the cluster after a hardware or software failure. A complete high availability setup includes many parts, one of which is the HA software. In addition to tangible items such as hardware and software, a good HA solution includes planning, design, customization, and change control. An HA solution reduces the amount of time that an application is unavailable by removing single points of failure.

SA MP delivers high availability to applications and middleware spanning any combination of Linux, AIX, or z/OS platforms by offering a product structure with two orderable components:

- the *SA MP (Base)* provides high availability and disaster recovery capabilities for Linux, including Linux on System z and AIX server clusters
- the *SA AM (E2E)* provides high availability for multi-tiered or composite business applications that can span z/OS, Linux, and AIX

This chapter discusses the customization of the SA MP (Base), while Chapter 13, “Customizing the Tivoli System Automation Application Manager (E2E),” on page 227 describes tailoring of the end-to-end product.

Tivoli System Automation for Multiplatforms 2.3 or later supports Linux on all IBM Systems platforms, in addition to AIX. For more information, visit:

<http://www.ibm.com/software/tivoli/products/sys-auto-linux>

Installing SA MP

This section provides a detailed description of the installation of SA MP. We use the following scenario, which reflects the 'Option 1A' mentioned above. It is also applicable to the 'Option 1B' scenario. The SA MP installation and cluster setup is exactly the same, only the number of SAP components (and SA MP resources) is larger.

We use the following scenario:

- A three node cluster is implemented on System z hardware.
- The SAP system ID is EP0.
- The <sapid>adm user ID is ep0adm, with the home directory /home/ep0adm.
- ABAP and/or Java SCS runs under z/OS USS (outside the cluster).
- SAP application servers are installed and run on each node of the cluster.
- a SAProuter is installed and must be highly available. It therefore runs anywhere in the cluster.

From the SAP perspective you need to install the application servers (ABAP only, Java only or double-stack one) on the different nodes of the cluster. For ABAP application server monitoring, SA MP needs the SAP supplied program rfcping. For Java application server monitoring, SA MP needs the GetWebPage.class utility, which is part of the preconfigured SAP policy as of version 5.3. If you plan to use the SAProuter, you need to set up a routing table (saproustab).

Figure 49 shows what a recommended Linux on System z cluster looks like. A light background indicates the floating resources that are currently active. In this example, the SAProuter group is active on Inxsaph.

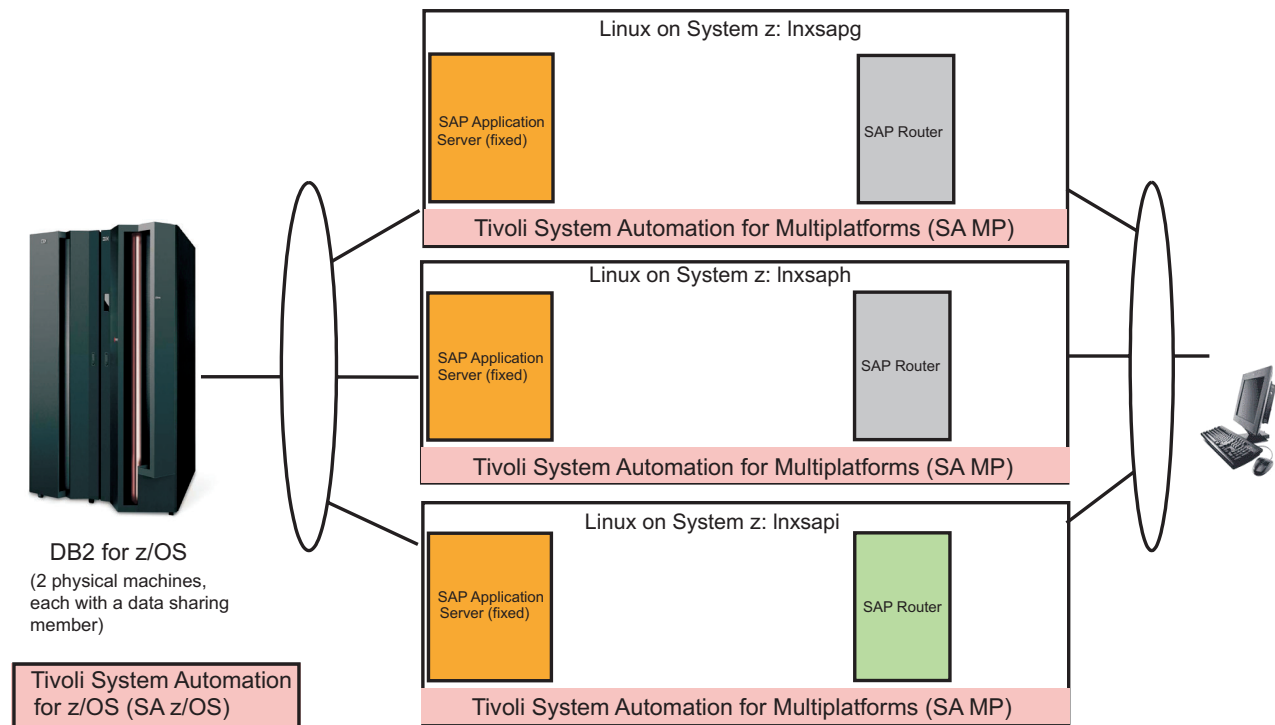


Figure 49. Initial Setup: Linux on System z cluster

Customizing the Tivoli System Automation for Multiplatforms (Base)

SA MP must be installed on all systems locally. To install SA MP, follow the installation procedure described in the respective publication for the release (see Table 40 on page 347). Briefly, to do so requires the following sequence of steps:

1. Log in as a user with root authority.
2. Be sure to run the required RSCT level on the system where required. The RSCT level check is only necessary under AIX and releases up to and including SA MP 2.3. Under Linux, RSCT was always part of SA MP and that is the case for AIX starting with 3.1 .
3. Change to the installation directory (for example SAM3100MPLinux) on the CD or in the structure unpacked from the tar archive.
4. Verify that your system has the necessary prerequisites for SA MP by running the following command:

```
./prereqSAM
```

This command writes a log to /tmp/prereqSAM.1.log, which should be checked before proceeding with the installation.

5. Install the SA MP software by entering:

```
# ./installSAM -nonls
```

The command writes a log to /tmp/installSAM.1.log, which should be checked to verify the installation. The -nonls option installs only the default English language.

6. The environment variable CT_MANAGEMENT_SCOPE must be set to "2" for all users of SA MP.

On SLES-10 this can be done by creating 2 new files in the /etc/profile.d/ directory. For example:

```
sa_mp.sh
export CT_MANAGEMENT_SCOPE=2
sap_mp.csh
setenv CT_MANAGEMENT_SCOPE 2
```

Any new logins automatically get the variable set. To verify after a new login, enter:

```
echo $CT_MANAGEMENT_SCOPE
```

This should print "2".

In addition, you have to grant read/write access for the IBM.Application class to the <sapsid>adm user-ID. You do this by adapting the ACL on each node in the following way:

1. Check for the existence of the /var/ct/cfg/ctrmc.acls file.
2. If it does not exist, copy it from /usr/sbin/rsct/cfg/ctrmc.acls.
3. With <sapsid>adm user-ID ep0adm as an example, add the lines:

```
IBM.Application
ep0adm@LOCALHOST * rw
UNAUTHENT * r
```

Activate the changes with the command:

```
# refresh -s ctrmc
```

Note: The above procedure assumes that you already know the SAPSID of your SAP system. If this is not yet determined, you must do this later on.

Setting up SA MP cluster to manage SAP resources

First, all nodes that will form the cluster need to be prepared. This includes a security setup, without which the following commands will not work.

To set up SA MP to manage SAP resources:

1. Execute the following command *on each node* (in this example lnxsapg, lnxsap, and lnxsap):

```
# preprnode lnxsapg lnxsap lnxsap
```

2. Create the SAP for SA MP cluster domain:

```
# mkcrpdomain sap lnxsapg lnxsap lnxsap
```

3. Start (online) the domain:

```
# startcrpdomain sap
```

4. Ensure the domain is online:

```
# lsrdomain
```

You should see output similar to

Name	OpState	RSCTActiveVersion	MixedVersions	TSPort	GSPort
sap	Online	2.4.3.2	No	12347	12348

5. Ensure that all nodes in the domain are online:

```
# lsrdnode
```

You should see output similar to

Name	OpState	RSCTVersion
lnxsapg	Online	2.4.3.2
lnxsaph	Online	2.4.3.2
lnxsapi	Online	2.4.3.2

Although this is a three-node cluster, a tie breaker is required as the nodes are not separate physical machines. For details of how to use a tie breaker and define quorum nodes, see “Using a tie breaker with SA MP” on page 224 and “Using SA MP Quorums with Linux on System z under z/VM” on page 225.

Installing the high availability policy for SAP

You require the sample SAP HA policy *version 5.3 or later* as a minimum version. This is because *all samples in this version of the book* are based on it.

Note: It is not within the scope of the current version of the sample SAP HA policy to support virtual IP addresses (VIPAs). IP aliasing is used instead.

The sample SAP HA policy allows you to run more than one SAP system within the same SA MP cluster. For simplicity and manageability reasons we highly recommend to run one SAP system in one cluster.

The high availability policy for SAP consists of a set of scripts necessary to control the various SAP resources and utilities that simplify management of the cluster. Installation of SA MP Policy Package installs the scripts in the `/usr/sbin/rsct/sapolicies/sap`

Customizing the Tivoli System Automation for Multiplatforms (Base)

directory. We recommend downloading the latest version. The pre-canned policies can be found on OPAL (select Tivoli System Automation under 'Tivoli products'): <http://catalog.lotus.com/wps/portal/topal>

Procedure:

1. Obtain the latest Policy Package and copy it to /tmp, for example.
2. Install the package with rpm under Linux, for example, or smitty under AIX.
3. Make sure your SAP <sapsid>adm user ID has read and execute access to the files and directories by entering:


```
# chown -R <sapsid>adm:sapsys /usr/sbin/rsct/sapolicies/sap
```
4. The following scripts and other files are installed:

Table 20. Components of the SA MP high availability policy for SAP

File	Description
readme.txt	Latest information
mksap	Script to create the SAP resources
rmsap	Script to remove SAP resources
lssap	Script to list the status of SAP resources
sapctrl_as	Script to control ABAP-only, double-stack, or Java-only application server instances
sapctrl_em	Script to control SAP Central Services processes (i.e., enqueue server, enqueue replication server, message server)
sapctrl_sys	Script to control SAPSID-independent processes
sapctrl_pid	Script to control a UNIX (or Linux) process and handle stop escalation (used by the above components)
ABAP_instances.conf	Sample configuration file for ABAP instances
J2EE_instances.conf	Sample configuration file for Java instances
GetWebPage.class-linux	Java program to do a primitive health check of the Java stack on <i>Linux systems</i> . Note: If you extracted the files via the .tar archive, you must rename this file to 'GetWebPage.class' on Linux systems. The GetWebPage.class file contained in the preconfigured policy package delivered with SA MP is the correct version for the underlying operating system.
GetWebPage.class-aix	Java program to do a primitive health check of the Java stack on <i>AIX systems</i> . Note: If you extracted the files via the .tar archive, you must rename this file to 'GetWebPage.class' on AIX systems. The GetWebPage.class file contained in the preconfigured policy package delivered with SA MP is the correct version for the underlying operating system.
GetWebPage.class-uss	Java program to do a primitive health check of the Java stack on <i>z/OS Unix System Services (USS) systems</i> . Note: If you extracted the files via the .tar archive, you must rename this file to 'GetWebPage.class' on z/OS USS systems. The GetWebPage.class file contained in the preconfigured policy package delivered with SA MP is the correct version for the underlying operating system.
sap_StartAfterRel_db	Sample commands to create StartAfter relations
sap_StartAfterRel_nfs_etc	Sample commands to replace static equivalency of NFS server by dynamic equivalency, create StartAfter relations between the enqueue server (IP) and NFS server, and create DependsOnAny relations between an application server and the NFS server

Table 20. Components of the SA MP high availability policy for SAP (continued)

File	Description
enqt.pf_scs01	Sample profile for SAP enqt (enqueue test) utility
StartABAP_EnqueueReplicationServer	Sample script to start ABAP enqueue replication server manually
StartJ2EE_EnqueueReplicationServer	Sample script to start Java enqueue replication server manually
InstallPolicy.doc	Documents the SAP sample policy installation

Implementing Option 1A (Variant A)

As mentioned above, using Option 1A you automate via *SA MP Application server (AS) instances* and make SAP utilities with an affinity to those Application servers like SAProuter or SAP Web Dispatcher highly available.

We do *not* recommend moving AS instances from one node to another in the cluster, as this would lead to an ‘unacceptable downtime’. The SAP architecture allows to run more than one AS. Run at least two AS's on different hardware to get the necessary AS redundancy.

AS resources are fixed resources whereas utilities (such as SAProuter) are *floating resources*. They require their own virtual hostname, which will move with those resources. NFS server and the SAP Central Services are run on z/OS and USS, outside of the SA MP cluster.

ABAP-only SAP systems and Java-only SAP systems are very similar from the SA MP viewpoint. Each ABAP or Java Application server is modeled as a SA MP resource/group. This is different for SAP systems that support ABAP and Java stacks simultaneously. Such a so-called *double-stack* system has Application servers, which runs both the ABAP and the Java stack. But such an AS is physically *one instance*. Starting an AS means starting the ABAP and the Java stack.

Although it is physically one instance, SA MP separates a double-stack AS into its two logical parts:

- the ABAP application server,
- the Java application server.

In other words, within an SA MP domain, one double-stack application server instance is automated as *two* logical application server instances:

- an ABAP application server instance, and
- a Java application server instance.

However, there is a close relationship between these two logical application servers:

- The Java instance is *always* started *after* the ABAP instance. This StartAfter relationship guarantees that starting the Java instance automatically triggers the prior start of the ABAP instance.
- On the other hand, the SA MP implementation makes sure that stopping the Java application server does *not* stop any of the double-stack server processes, but instead only the monitoring Java program *java GetWebPage* (which does a primitive health check of Java application servers).

Figure 50 on page 202 shows an overview of the SAP policy definitions for the Java part of a double-stack system with SAPSID EP0.

TSA for Multiplatforms HA Policy for SAP (double-stack system)

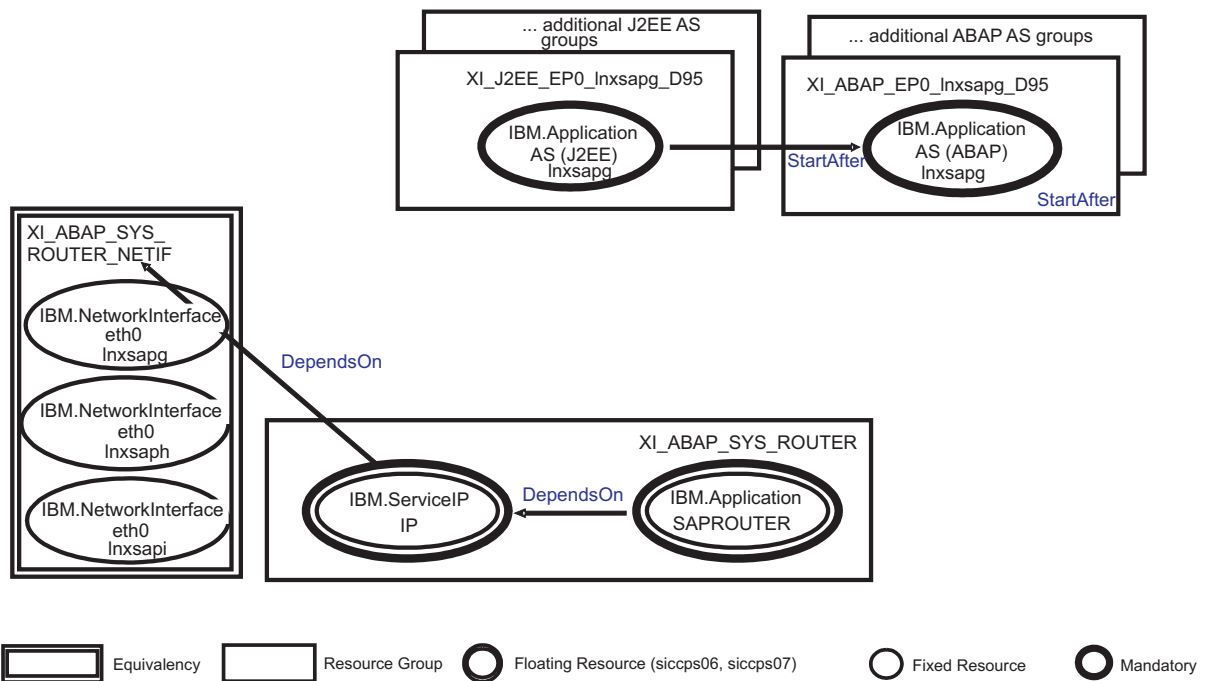


Figure 50. Overview of the SAP policy definitions for double-stack system

In general, the entire SAP application described in this chapter is separated into groups that consist of resources that belong together. The different groups are:

- Two or more ABAP application server groups, containing one application server (AS) each (`<ABAP_PREF>_<SAPSID>_<node>_D<sysnr>`).
- Two or more Java application server groups, containing one application server (AS) each (`<J2EE_PREF>_<SAPSID>_<node>_D<sysnr>` or `<J2EE_PREF>_<SAPSID>_<node>_J<sysnr>`).
- Optionally, you might wish to add a router group (`<ABAP_PREF>_SYS_ROUTER`), containing the router (SAPROUTER) and a service IP address (IP).
- Optionally, you might wish to add a Web Dispatcher group (`<ABAP_PREF>_SYS_WEBDISP`, containing the Web Dispatcher (SAPWEBDISP) and a service IP address (IP).

The naming conventions we use are described in “Tivoli System Automation for Multiplatforms” on page 124. As noted there, the resource names use the group name as the prefix. For example, the ABAP application server, which is a member of a group named `XI_ABAP_EP0_Inxsapg_D95`, is called `XI_ABAP_EP0_Inxsapg_D95_AS`. In the following, the base names of the resources can appear without the group prefix.

All groups have a member location of ‘collocated’, which means that the resources of these groups always run together on the same node.

A network equivalency must be created for each service IP. The name of the equivalency is the name of the group to which the service IP belongs, suffixed by

Customizing the Tivoli System Automation for Multiplatforms (Base)

'_NETIF'. This means for our sample policy: XI_ABAP_SYS_ROUTER_NETIF for the service IP of the router group. Each service IP has a DependsOn relationship to its equivalency.

All above listed SAP components can be modeled as SA MP resources using the predefined SA MP policy. They can be easily created by running the mksap script. In order to do this, you need to adapt two configuration files. The first configuration file is for the ABAP application servers and the SAP utilities SAProuter and/or SAP Web Dispatcher. In the preconfigured policy it is named ABAP_instances.conf. A second configuration file is necessary for the Java application servers. It is named J2EE_instances.conf. The predefined SA MP policy for SAP contains the two sample configuration files. You must adapt the entries in these files to your SAP environment (see the next section).

You create the SA MP policy for a 'double-stack' SAP system by executing the following steps:

1. Adapt the sample ABAP and Java configuration files to your SAP environment.
2. Execute the 'mksap' script multiple times with appropriate command line options as described in the following. Use the ABAP configuration file for ABAP Application servers and SAP utilities and the Java configuration file for Java Application servers.
3. Test the policy.
4. Save the policy.
5. Verify your SAP installation running under SA MP control.

If you have an ABAP-only installation, the Java configuration file adoption and resource creation is *not* required.

If you have a Java-only installation:

- If you do not run any of the SAP utilities SAProuter or SAP Web Dispatcher then you need to adapt only the Java configuration file.
- If you run any of the SAP utilities SAProuter or SAP Web Dispatcher then you need to adapt besides the Java configuration file also the ABAP configuration file for the respective utility.

These steps are described in detail in the following section.

Customizing the HA policy for a double-stack or Java-only SAP system

These are the main steps you should follow.

Step 1: Adapt the sample ABAP and Java configuration files

The preconfigured SA MP policy for SAP comprises a script to create all SA MP resources, resource groups, relationships and equivalencies that are necessary to make an SAP system highly available. Besides command line parameters, the 'mksap' script reads a configuration file that contains all SAP-system-specific parameter values. Two sample configuration files are provided with the preconfigured SA MP SAP policy:

- ABAP_instances.conf, which you can use to create all ABAP resources and SAP-system-independent utility resources, and
- J2EE_instances.conf, which you can use to create all Java resources.

Customizing the Tivoli System Automation for Multiplatforms (Base)

The ABAP_instances.conf file contains the following entries, which you must adapt to your SAP environment. Please comment out all other entries by adding a # in front.

```
INSTALL_DIR="/usr/sbin/rsct/sapolicies/sap"      # installation directory of SAP policy scripts
CLUSTER="HA1_DOMAIN"                          # TSA domain name for below mentioned nodes; not used in HA scripts
NODES="ihlsco1 ihlsco2 ihlsco3"               # list of nodes included in the TSA domain
PREF="SAPECC_ABAP"                            # prefix of all TSA SAP ABAP resources and SAP router/SAP web dispatcher
# for ABAP instance we recommend to use <common prefix>_ABAP

SAPID="HA1"                                    # SAP system ID
SAP_ADMIN_USER="ha1adm"                       # SAP administration user ID
ASNOS="02 03 04"                              # list of instance numbers of the SAP App. Servers
INSTDIRS="DVEBMGS02 D03 D04"                  # list of instance directories of the SAP App. servers

# in our test setup we currently do not have a sap router, so the entries are commented out
#SAPROUTER_IP="9.153.165.xx"                   # SAP router IP address
#SAPROUTER_IP_NETMASK="255.255.255.0"         # SAP router IP address' netmask
#SAPROUTER_IP_INTERFACE="eth0"                # interface on which SAP router IP address is activated on each node as alias
#ROUTTAB="/usr/sap/L0P/SYS/profile/saproustab" # fully qualified SAP router routing table

# in our test setup we currently do not have a sap web dispatcher, so the entries are commented out
#SAPWEBDISP_IP="9.153.165.zz"                  # SAP web dispatcher IP address
#SAPWEBDISP_IP_NETMASK="255.255.255.0"        # SAP web dispatcher IP address' netmask
#SAPWEBDISP_IP_INTERFACE="eth0"               # interface where SAP web dispatcher IP addr activated on each node as alias
#SAPWEBDISP_DIR="/usr/sap/L0P/sapwebdisp"      # directory where the profile sapwebdisp.pfl of the SAP web dispatcher is located
```

Figure 51. Entries to be adapted in the ABAP_instances.conf file

To adapt the file, first create a backup of the original, then load it into a text editor of your choice and change the values to the right of the equal signs as required.

The above entries reflect a test environment, where no SAP router and no SAP web dispatcher was running. If you run a SAP router/web dispatcher in your environment, then remove the comment sign (#) at the beginning of the lines with the SAPROUTER/SAPWEBDISP specific parameters to be able to create TSA resources for SAP router and/or SAP web dispatcher. As each such utility is a floating resource it needs its own floating IP. Make sure that you have a routing table (saproustab) for the SAProuter accessible by all nodes in the cluster. The same applies for the SAP Web Dispatcher profile.

Please note that the number of entries in NODES, ASNOS, and INSTDIRS must be the same, because the sample policy assumes the following:

- the application server with the first instance number (02) and first instance directory (DVEBMGS02) runs on node 1 (ihlsco1).
- the application server with the second instance number (03) and second instance directory (D03) runs on node 2 (ihlsco2).
- the application server with the second instance number (04) and second instance directory (D04) runs on node 3 (ihlsco3).

and so on.

The J2EE_instances.conf file contains the following entries, which you must adapt to your SAP environment. Please comment out all other entries by adding a # in front.

```

INSTALL_DIR="/usr/sbin/rsct/sapolicies/sap"           # installation directory of SAP policy scripts
CLUSTER="HA1_DOMAIN"                               # TSA domain name for below mentioned nodes; not used in HA scripts
NODES="ihlscoh1 ihlscoh2 ihlscoh3"                # list of nodes included in the TSA domain
PREF="SAPECC_JAVA"                                 # prefix of all TSA SAP J2EE resources
# for J2EE instance we recommend to use <common prefix>_J2EE
ABAP_PREF="SAPECC_ABAP"                           # prefix of all TSA SAP ABAP resources; it is only needed, if the J2EE
                                                    # application server is part of an 'double-stack' application server (like for XI);
                                                    # it must have the same value as PREF in ABAP_instances.conf
SAPSID="HA1"                                       # SAP system ID
SAP_ADMIN_USER="haladm"                           # SAP administration user ID
ASNOS="02 03 04"                                  # list of instance numbers of the SAP App. Servers
INSTDIRS="DVEBMGS02 D03 D04"                       # list of instance directories of the SAP App. servers
# setting J2EE_HTTP_PORTS indicates, that profile is for J2EE
J2EE_HTTP_PORTS="50200 50300 50400"               # list of J2EE instance HTTP ports (default: 50000+instance number*100 +0)

```

Figure 52. Entries to be adapted in the `J2EE_instances.conf` file

Entries which differ from the ABAP profile are:

- PREF as we need a different prefix for Java resources.
- ABAP_PREF as we need to know the prefix for ABAP resources (in case that the Java application server is part of an 'double-stack' application server we need to know the ABAP resources prefix in order to be able to generate a StartAfter relationship between the Java instance and the ABAP instance).
- J2EE_HTTP_PORTS as each Double-stack Java instance needs an http port.

The number of entries in NODES, ASNOS, and INSTDIRS must be the same, because the sample policy assumes the following:

- the Java application server with the first instance number (02) and first instance directory (DVEBMGS02) runs on node 1 (ihlscoh1),
- the Java application server with the second instance number (03) and second instance directory (D03) runs on node 2 (ihlscoh2),

and so on.

Step 2: Run the mksap script to create SA MP resources

Step 2a. Create resources for ABAP application servers and optionally for the SAP utilities

The next step is to run the mksap script to create the SA MP resource groups, resources, relationships and equivalencies for ABAP application servers and optionally for the SAP utilities. As the root user, you have to run the mksap script multiple times with different options. Change to your install path of the sample policy script. The mksap script supports the following command line options:

```

./mksap -h
Usage: mksap [-h] [-s] [-c ConfigFile] [-a NodeName] [-t Type] [-i InstType] [-l ABAPCharSet] [-n NodeName[,NodeName]] [-v VirtASHostName]
mksap creates (parts of) the sample TSA policy for a highly available SAP environment.
-h, Print this help.
-s, Use static equivalencies for sample TSA policy (not recommended); default: dynamic equivalencies.
-c, Use configuration file ConfigFile (fully qualified or in current dir); default: saphasalinux.conf in current dir.
-a, Create TSA policy entries for a SAP application server on NodeName and add them to an existing TSA policy.
-t, Define the type of the sample TSA policy to be created; valid types: CentralServices|SAPCommon|AppServer|StandardSCS|ABAPCI;
    default: None, create CentralServices, SAPCommon and AppServer. AppServer and Standard SCS create fixed resources.
-i, Define the installation type; valid types: ABAP|ADDIN|J2EE; default: create entries for ABAP stack.
-l, Define the character set used by ABAP stack for ADDIN installation type; valid sets: ASCII|UNICODE; default: UNICODE.
-n, Create TSA policy entries for all nodes specified; default: create entries for all nodes defined in the configuration file.
-v, Use specified virtual hostname for the AppServer resource/rg and as parameter for Start/Stop/Monitor command; default: None.
-o, Create TSA policy entries for ASCS with ES and MS only (like SAPinst created ASCS); default: None, create ES,MS,GW,CO,SE.

```

Figure 53. mksap options

1. Run as root user the following command. You must adapt the node name. If you have installed the Application server with its own virtual hostname you must in addition use the `-v<virtual hostname>` option.

```
./mksap -c ABAP_instances.conf -t AppServer -i ABAP -n ihlscoh1
```

Customizing the Tivoli System Automation for Multiplatforms (Base)

This step creates the SA MP resources, etc. for an ABAP application server on node ihlscohl. Run the mksap command for each Application server.

Notes:

- a. If you have installed the application server with its own virtual hostname and you want to move the AS from one node to another and you have defined a mount point resource (MPR) for its instance directory in order to also move this MPR, then you must change the sapctrl_as script. Change: `exit ${STATUS_FAILED_OFFLINE}` to `exit ${STATUS_OFFLINE}`, where the comment indicates (# if this is for a floating appl. server, which has a floating mount point then change to `exit ${STATUS_OFFLINE}`).
 - b. Such a floating Application server is *not a recommended setup*.
2. *Optionally*, run the mksap script to create the TSA for MP resources, etc. for the SAP utilities. Run this step also for a Java only system, if you want to run a SAP Web Dispatcher under TSA MP control. Run:

```
./mksap -c ABAP_instances.conf -t SAPCommon
```

This creates the SA MP resources, etc. for the SAProuter and/or SAP Web Dispatcher if the associated entries in the configuration file are active (not commented out).

Step 2b: Create resources for Java application servers

1. As the root user, run the mksap script again to create the SA MP resources, etc. for Java application servers. In this case, use the configuration file `J2EE_instances.conf`. Before you proceed to create the *double-stack* Java application server resources, make sure that the corresponding ABAP application server resources have already been created. If you have installed the Application server with its own virtual hostname, you must in addition use the `-v <virtual hostname>` option.
2. Run the mksap with adapted node name:

```
./mksap -c J2EE_instances.conf -t AppServer -i ADDIN -n ihlscohl
```

This creates the SA MP resources, and so on for a Java application server on node ihlscohl. It also generates the StartAfter relationships between the Java application server and its corresponding ABAP application server. Be aware that this relation can only be created if the ABAP application server has been created previously.
3. Run the mksap command for each Application server.

Note that for a Java-only system you must use the `-i J2EE` option:

```
./mksap -c J2EE_instances.conf -t AppServer -i J2EE -n ihlscohl (-v <virtual hostname>)
```

Step 3: Perform a quick test of the SAP policy

You can now start all your SAP resources by issuing the command:

```
chrg -o online -s "Name like '%'"
```

To check their status, issue the command:

```
lssam -top
```

If the Application server resources get into a 'stuck online' status, you may need to change the `sapctrl_as` script to wait longer. After all resources are up, you can stop them by issuing the command:

```
chrg -o offline -s "Name like '%'"
```

Step 4: Save the policy

To save the policy, issue this command:

```
sampolicy -s <policy-file>.xml
```

where <policy-file> is a name you assign.

To uninstall, you can perform the step “Removing the HA policy” on page 223.

Step 5: Verify your SAP installation running under SA MP control

To verify your HA setup, please perform the steps listed in Chapter 15, “Verifying your implementation on z/OS,” on page 249.

The procedures and failover scenarios verify that SA MP keeps SAP resources available for normal operations, planned and unplanned outages.

Implementing Option 1B (Variant B)

When using Option 1B you automate with SA MP:

- the SAP Central Services (for ABAP and/or for Java),
- the SAP application server (AS) instances,
- the SAP utilities with an affinity to those Application servers like SAProuter or SAP Web Dispatcher, and
- the NFS server (if you decided to run the NFS server within the SAP cluster for just this SAP system). You may also run NFS server in its own SA MP cluster serving several SAP systems in the ‘enhanced NFS Server policy’ variant.

As the SAP database must reside on z/OS, it must be kept highly available there with SA z/OS. Also the sysplex wide SAPOSCOL/SAPCCMSR must be automated with SA z/OS.

The SAP Central Services, the SAP utilities as well as the NFS Server are floating resources needing their own virtual hostname which will move with those resources.

We do not recommend moving AS instances from one node to another in the cluster as this means an ‘unacceptable downtime’. The SAP architecture allows to run more than one AS. Run at least 2 AS on different hardware to get the necessary AS redundancy. Such AS resources are fixed resources.

Before you proceed, ensure that you have read these chapters:

- Chapter 8, “Concepts for a high availability SAP solution,” on page 95
- Chapter 9, “Preparing a high availability SAP solution,” on page 119

Both the above chapters also *apply to Option 1B for SAP on Linux on System z*. They discuss the hardware and software architecture and offer planning information, including:

- File system setup
- SAP installation

Chapter 10, “Customizing SAP for high availability,” on page 135 also discusses the customization of SAP needed for high availability. It describes the following components:

- ABAP SCS and Java SCS
- Application server instances

Customizing the Tivoli System Automation for Multiplatforms (Base)

This chapter also provides you with a good overview of what has to be considered when making an SAP system highly available. For example, it explains in detail how to set up an SAP system to run with highly available SAP Central Services for ABAP and/or Java (also called 'standalone enqueue server' with replication setup), which is a prerequisite for removing the single point of failure represented by SAP enqueue processing.

After installation and customization of SAP, the following steps have to be performed to establish the 'Option 1B' setup:

1. Make NFS server highly available via SA MP
2. Customize the sample SA MP high availability policy for SAP
3. Verify the high availability double-stack installation

These steps are now described.

Making NFS highly available via SA MP

To get a highly available NFS server, you can set up and apply the NFS server HA policy, which is pre-configured by SA MP. See IBM Tivoli System Automation for Multiplatforms: Application Enablement of NFS File Server, which can be downloaded as file **ITSAMP-NFS-Server.pdf** from:

<ftp://ftp.software.ibm.com/software/tivoli/products/sys-auto-linux>

To set up the NFS server HA policy, *a domain must exist*. For details on how to set up a domain, see "Setting up SA MP cluster to manage SAP resources" on page 199.

Mount the NFS-exported directory at all nodes in the SAP domain, including nodes where the NFS server itself can run if the node also runs an SAP application server. You can do this either by:

- adding the NFS client mount to `/etc/fstab`, or
- using automounter (the option *we recommend*).

Alternatively, you can achieve a highly available file system, for example, by setting up a highly available NFS server under z/OS, as described in Part 4, "SAP," on page 93 and Part 5, "System Automation," on page 169.

Run the NFS server in its own cluster

We recommend to run the NFS server within its own cluster to make it highly available. As SAP only starts successfully, if the NFS server is running, you have two options to handle this cross cluster 'StartAfter' relationship:

- Run SA AM (E2E) in order to model the StartAfter relationship between SAP and NFS.
- Do nothing. Then it could occur that the NFS cluster and the SAP cluster are started from SA MP at the same time. As the NFS server is not up and running when SA MP does the SAP start attempt, this start will not be successful. Consequently, SA MP will do another start attempt and if that fails again a final one. This should be enough time to allow the NFS server to come up and to avoid getting the SAP resources into the 'failed offline' state.

You can use the standalone NFS cluster to service several SAP systems. If you want to do that, please do not use the sample NFS policy as this introduces a new Single Point of Failure: If one NFS resource for one SAP system is not starting, then the NFS resources for all other SAP systems are not starting. Please ask IBM for the available 'enhanced NFS server policy' which avoids that SPOF.

Special considerations for AIX

If you have two mount points for the filesystems `/var/lib/nfs` and `/sapmnt` defined in the configuration file of the NFS server resources and if both are on the same logical volume such as:

```
# --common local mountpoint for shared data
#   If more instances of <data_>, add more rows, like: data_tmp, data_proj...
#   Note: the keywords need to be unique!
data_varlibnfs="/nfsctrl1"
data_work="/SAPdata"

# --LVM definitions: VG and optional hdisk ( for AIX only )
#   one entry allowed, like: myvg ... with hdisk like: myvg hdisk5
lvm="nfshavg"
```

In this case you must add a StartAfter relationship between both, such as:

```
mkrel -p StartAfter -S IBM.Application: SA-nfsserver-data-work -G IBM.Application:
SA-nfsserver-data-varlibnfs NFS_varlibnfs_SAPdata_StartAfter
```

This is to avoid, that system automation tries to mount both filesystems at the same time which can cause the `fsck` command to fail. This `fsck` command is done before really mounting the filesystem.

Also, if you are running an HA NFS Server under AIX and use the automounter, then following changes to the system are necessary:

```
[root@achalm43] cat /etc/rc.d/init.d/automount_init
#!/bin/ksh
```

```
#####
# purpose: script that will start|stop the automountd daemon.
#####
```

```
case "$1" in
start )
    if [ -s /etc/auto_master ]; then
        /usr/sbin/automount
    fi
    ;;
stop )
    stopsrc -g autofs
    ;;
* )
    echo "Usage: $0 (start | stop)"
    exit 1
esac
```

```
[root@achalm43] /etc/rc.d/rc2.d
#ln -s /etc/rc.d/init.d/automount_init Sautomount_init
#ln -s /etc/rc.d/init.d/automount_init Kautomount_init
```

Customize the sample SA MP high availability policy for SAP

SAP NetWeaver '04 supports three different installation setups:

- an ABAP-only installation
- a Java-only installation, or
- a so-called 'double-stack' installation (as for SAP's Process Integration)

An *ABAP-only installation* consists of an ABAP CI and ABAP DI application servers. For high availability, the ABAP CI must be replaced by an ABAP SAP Central Services (ASCS) instance with its own virtual host name and an ABAP 'remaining' CI.

Customizing the Tivoli System Automation for Multiplatforms (Base)

A *Java-only installation* consists of a Java CI, Java SCS instance, and Java DI application servers. For high availability the Java SCS instance must be installed (or manually adapted) to run with its own virtual host name.

The most complicated installation is the *'double-stack' installation*. . It comprises all the components of both the ABAP-only and Java-only installations. For high availability, you have an ABAP SCS instance and a Java SCS instance, each with its own virtual host name, an ABAP 'remaining' CI, a Java CI, and Java and ABAP Dialog Instances (DI)s.

Note: For the System z solution, the ABAP and Java SCS instances can also run under z/OS USS.

Java-only SAP systems are very similar to ABAP-only SAP systems from the SA MP viewpoint. This is different for SAP systems that support ABAP and Java stacks simultaneously. Such a so-called 'double-stack' system has two SAP Central Services instances, an ABAP SCS and a Java SCS in parallel. And it runs 'double-stack' application servers. A 'double-stack' application server is physically one instance that runs both the ABAP and the Java stacks. Starting an AS means starting the ABAP and the Java stack. Although it is physically one instance, SA MP separates an 'double-stack' AS into its two logical parts, the ABAP application server and the Java application server. In other words, within a SA MP domain, one 'double-stack' application server instance is automated as two logical application server instances, an ABAP application server instance and a Java application server instance. However, there is a close relationship between these two logical application servers. The Java instance is always started after the ABAP instance. This StartAfter relationship guarantees that starting the Java instance automatically triggers the prior start of the ABAP instance. On the other hand the SA MP implementation makes sure, that stopping the Java application server does not stop any of the 'double-stack' server processes but rather only the monitoring Java program 'java GetWebPage', which does a primitive health check of Java application servers.

A 'double-stack' SAP system comprises the following components from an HA perspective:

Component A: ABAP resources

- ABAP SAP Central Services instance. It is named
ASCS<instance-number>

and comprises, among other things, the ABAP standalone enqueue server, the message server, and its own virtual host name (IP address).

- Two or more ABAP application servers as dialog instances.

Component J: Java resources

- Java SAP Central Services instance. It is named
SCS<instance number>

and comprises the Java (standalone) enqueue server, the message server, and its own virtual host name (IP address).

- Two or more Java application servers as dialog instances.

Component I: SAP-system-independent resources (optional)

- SAProuter program, which is an SAP utility serving SAP ABAP systems.
- SAP Web Dispatcher, which is an SAP utility serving HTTP requests for SAP ABAP and Java systems.

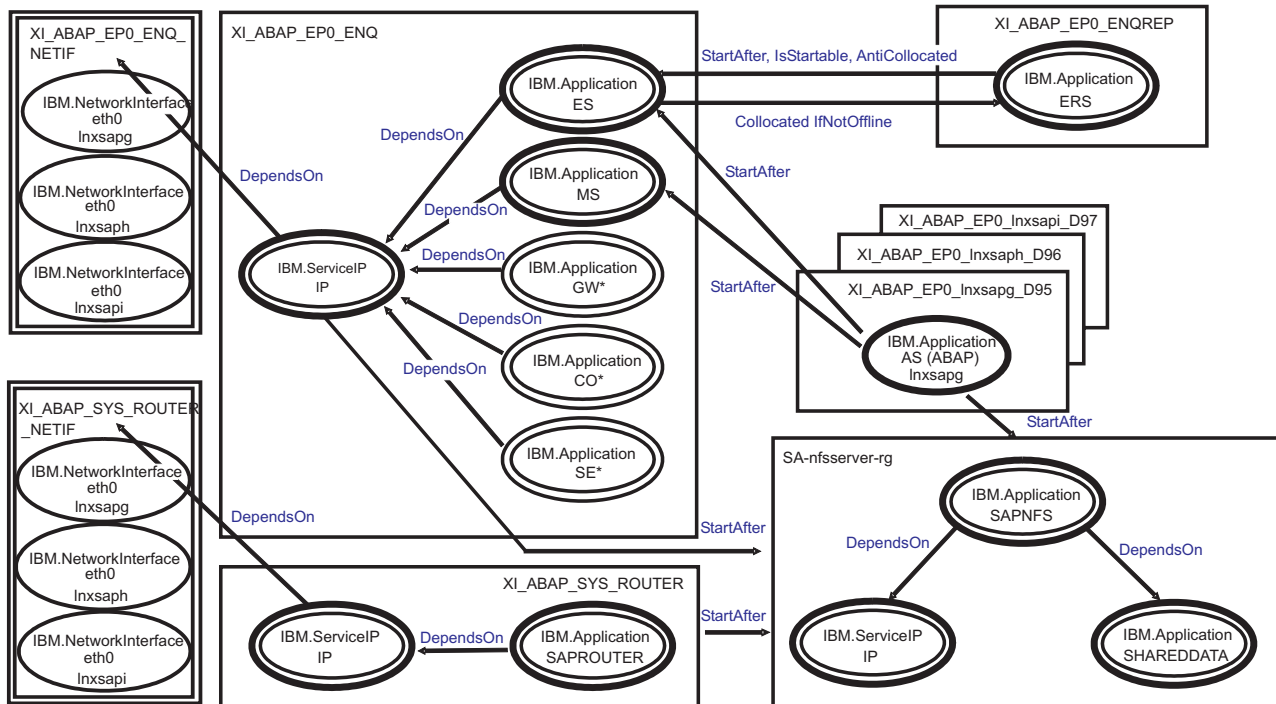
Customizing the Tivoli System Automation for Multiplatforms (Base)

Component D: Dependencies between component resources noted above and the NFS server (if the NFS server runs within the SAP domain)

For a Java-only environment such as Enterprise Portal 6.0 (EP 6.0), there is no component A.

Figure 54 shows an overview of the SAP policy definitions for the ABAP part of a double-stack system with SAPSID EP0.

TSA for Multiplatforms HA Policy for SAP (double-stack system, ABAP part)



* optional IBM enhancements (for compatibility with older SAP HA policy versions)



Figure 54. Overview of the SAP policy definitions for double-stack system (ABAP part)

Figure 55 on page 212 shows an overview of the SAP policy definitions for the Java part of a double-stack system with SAPSID EP0.

TSA for Multiplatforms HA Policy for SAP (double-stack system, Java part)

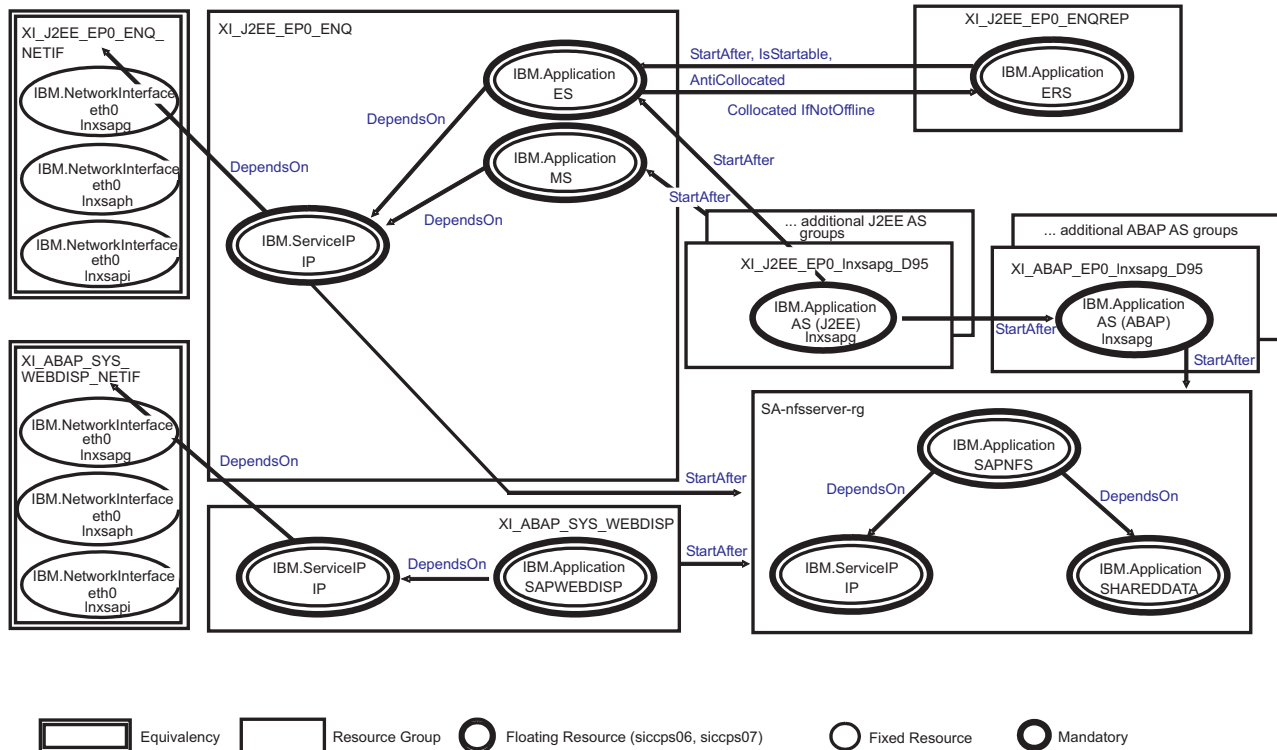


Figure 55. Overview of the SAP policy definitions for double-stack system (Java part)

In general, the entire SAP application described in this chapter is separated into groups that consist of resources that belong together. The different groups are:

- The ABAP enqueue group (<ABAP_PREF>_EP0_ENQ), containing the enqueue server (ES), message server (MS), IP address (IP), and *optionally* a gateway (GW), syslog collector (CO), syslog sender (SE).
- The ABAP enqueue replicator group (<ABAP_PREF>_EP0_ENQREP), containing the enqueue replication server (ERS).
- The router group (<ABAP_PREF>_SYS_ROUTER), containing the router (SAPROUTER) and a service IP address (IP).
- One or more ABAP application server groups, containing one application server (AS) each (<ABAP_PREF>_<SAPSID>_<node>_D<sysnr>).
- The Web Dispatcher group (<ABAP_PREF>_SYS_WEBDISP, containing the Web Dispatcher (SAPWEBDISP) and a service IP address (IP).
- The Java enqueue group (<J2EE_PREF>_EP0_ENQ) containing the enqueue server (ES), message server (MS), and IP address (IP).
- The Java enqueue replicator group (<J2EE_PREF>_EP0_ENQREP) containing the enqueue replication server (ERS)
- One or more Java application server groups, containing one application server (AS) each (<J2EE_PREF>_<sapsid>_<node>_D<sysnr> or <J2EE_PREF>_<sapsid>_<node>_J<sysnr>)

The naming conventions we use are described in “Tivoli System Automation for Multiplatforms” on page 124. As noted there, the resource names use the group name as the prefix. For example, the enqueue server, which is a member of a group named XI_ABAP_EP0_ENQ, is called XI_ABAP_EP0_ENQ_ES. In the following, the base names of the resources can appear without the group prefix.

Customizing the Tivoli System Automation for Multiplatforms (Base)

Note: The base names of some resources are the same in different groups. They are made unique in the entire scenario by the group name.

All groups have a member location of 'collocated', which means that the resources of these groups always run together on the same node.

The main components that this HA solution covers are the ES and the ERS. These two components have the most complex relationships.

A network equivalency is created for both service IPs. The name of the equivalency is the name of the group to which the service IP belongs, suffixed by '_NETIF'. This means for our sample policy: XI_ABAP_EP0_ENQ_NETIF for the service IP of the enqueue group and XI_ABAP_SYS_ROUTER_NETIF for the service IP of the router group. Each service IP has a DependsOn relationship to its equivalency.

Creating the SA MP resources for a SAP system

This section provides a detailed description of the SA MP resource creation for a SA system.

We use the following scenario:

- A three node cluster is implemented on System z hardware.
- A high-availability NFS server runs in the cluster and exports the standard SAP file systems for each node.
- The SAP system ID is EP0.
- The <sapid>adm user ID is ep0adm, with the home directory /home/ep0adm.
- ABAP SCS runs anywhere in the cluster.
- Java SCS runs anywhere in the cluster.
- SAP application servers run on each node of the cluster.
- a SAProuter runs anywhere in the cluster.

Figure 56 on page 214 shows what a recommended Linux on System z cluster looks like. A light background indicates the floating resources that are currently active. In this example, the ABAP SCS group is active on lnxsaph.

Customizing the Tivoli System Automation for Multiplatforms (Base)

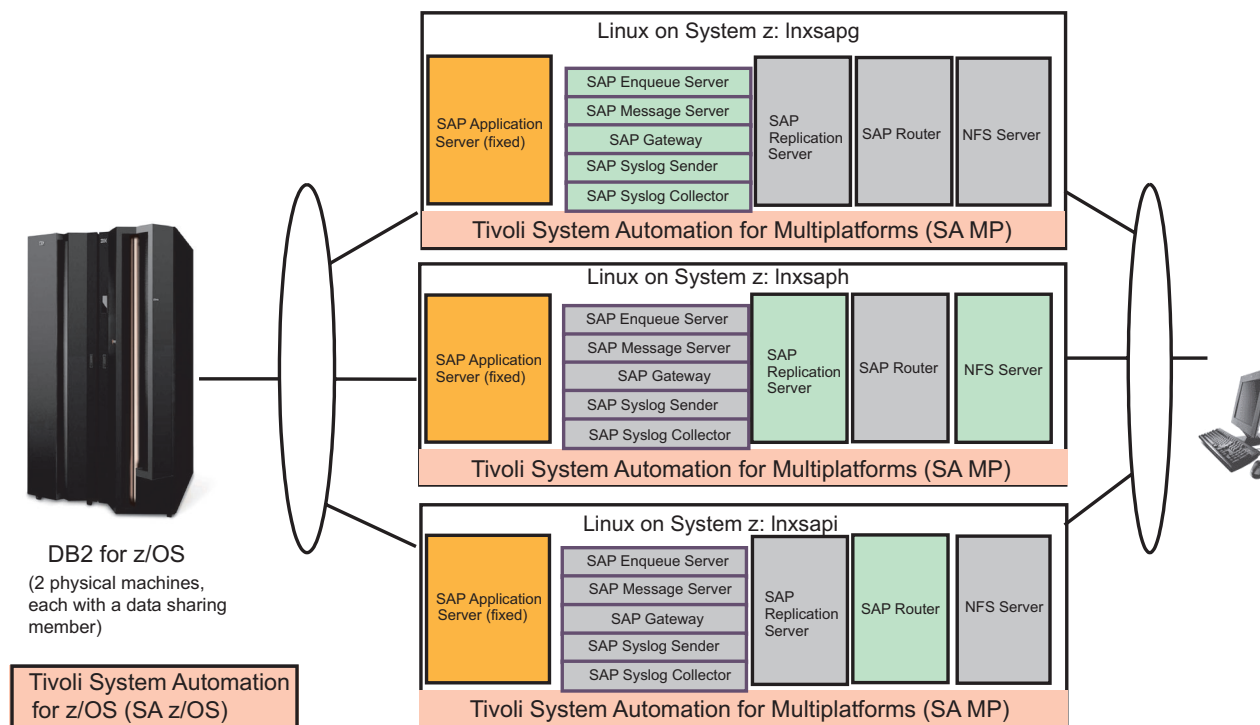


Figure 56. Option 1B: Linux on System z cluster

All the SAP ABAP, Java and utility resources (for components A, J and I) can be created by using the predefined or sample SA MP policy. In order to do this, you need to adapt two configuration files.

- The first configuration file is for the ABAP resources and the optional SAP utilities (if you want to have a highly available SAProuter/SAP Web Dispatcher). In the preconfigured policy it is named ABAP_instances.conf.
- A second configuration file is necessary for Java resources and is named J2EE_instances.conf.

The predefined SA MP policy for SAP contains the two sample configuration files. You must adapt the entries in these files to your SAP environment (see the next section).

You create the SA MP policy for a 'double-stack' SAP system by executing the following steps:

1. Adapt the sample ABAP and Java configuration files to your SAP environment.
2. Execute the 'mksap' script multiple times with appropriate command line options as described in the following. Use the ABAP configuration file for ABAP resources and SAP utilities and the Java configuration file for Java resources.
3. If the NFS server is running in the SAP domain: Adapt the dependency file to define dependencies with respect to the NFS server and execute the native SA MP commands to create the dependencies of the SAP resources to it.
4. Test the policy.
5. Save the policy.
6. Verify your SAP installation running under SA MP control. This is described in "Verifying your high availability implementation with SA MP" on page 223.

Customizing the Tivoli System Automation for Multiplatforms (Base)

If you have an ABAP-only installation then the Java configuration file adoption and resource creation is *not* required.

If you have a Java only installation, then:

1. If you do not run any of the SAP utilities SAProuter or SAP Web Dispatcher then you need to adapt only the Java configuration file.
2. If you run any of the SAP utilities SAProuter or SAP Web Dispatcher then you need to adapt besides the Java configuration file also the ABAP configuration file for the respective utility.

These steps are described in detail in the following section.

Customizing the high availability policy for a double-stack or Java-only SAP system

Step 1: Adapt the sample ABAP and Java configuration files:

Note: For a Java-only environment such as EP 6.0, adaptation of the ABAP_instances.conf file is only necessary if you plan to run an SAP Web Dispatcher under SA MP control.

The preconfigured SA MP policy for SAP comprises a script to create all SA MP resources, resource groups, relationships and equivalencies that are necessary to make an SAP system highly available. Besides command line parameters, the 'mksap' script reads a configuration file that contains all SAP-system-specific parameter values. Two sample configuration files are provided with the preconfigured SA MP SAP policy:

- **ABAP_instances.conf**, which you can use to create all ABAP resources (and optionally all SAP-system-independent resources), and
- **J2EE_instances.conf**, which you can use to create all Java resources.

The ABAP_instances.conf file contains the following entries, which you must adapt to your SAP environment. Please comment out all other entries by adding a # in front.

Customizing the Tivoli System Automation for Multiplatforms (Base)

```
INSTALL_DIR="/usr/sbin/rsct/sapolicies/sap"      # installation directory of SAP policy scripts
CLUSTER="HA1_DOMAIN"                          # TSA domain name for below mentioned nodes; not used in HA scripts
NODES="ihlsc0h1 ihlsc0h2 ihlsc0h3"           # list of nodes included in the TSA domain
PREF="SAPECC_ABAP"                           # prefix of all TSA SAP ABAP resources and SAP router/SAP web dispatcher
# for ABAP instance we recommend to use <common prefix>_ABAP

SAPSID="HA1"                                  # SAP system ID
SAP_ADMIN_USER="ha1adm"                      # SAP administration user ID

ENQSRV_IP="10.101.5.194"                      # ASCS instance IP address
ENQSRV_HOSTNAME="ha1ascsv"                  # virt. hostname for ASCS IP address; if not defined, comment out or set to ""
ENQSRV_IP_NETMASK="255.255.255.0"          # ASCS instance IP address' netmask
ENQSRV_IP_INTERFACE="eth0"                 # interface on which ASCS instance IP address is activated on each node as alias
ENQNO="00"                                  # instance number of the ASCS instance
ENQDIR="ASCS00"                             # instance directory of the ASCS instance

ASNOS="02 03 04"                            # list of instance numbers of the SAP App. Servers
INSTDIRS="DVEBMGS02 D03 D04"               # list of instance directories of the SAP App. servers

# in our test setup we currently do not have a sap router, so the entries are commented out
#SAPROUTER_IP="9.153.165.xx"                # SAP router IP address
#SAPROUTER_IP_NETMASK="255.255.255.0"      # SAP router IP address' netmask
#SAPROUTER_IP_INTERFACE="eth0"             # interface on which SAP router IP address is activated on each node as alias
#ROUTTAB="/usr/sap/LOP/SYS/profile/saproustab" # fully qualified SAP router routing table

# in our test setup we currently do not have a sap web dispatcher, so the entries are commented out
#SAPWEBDISP_IP="9.153.165.zz"              # SAP web dispatcher IP address
#SAPWEBDISP_IP_NETMASK="255.255.255.0"    # SAP web dispatcher IP address' netmask
#SAPWEBDISP_IP_INTERFACE="eth0"           # interface where SAP web dispatcher IP addr. is activated on each node as alias
#SAPWEBDISP_DIR="/usr/sap/LOP/sapwebdisp"  # directory where the profile sapwebdisp.pfl of the SAP web dispatcher is located
```

Figure 57. Entries to be adapted in the `ABAP_instances.conf` file

To adapt the file, first create a backup of the original, then load it into a text editor of your choice and change the values to the right of the equal signs if required.

The above entries reflect a test environment, where no SAP router and no SAP web dispatcher was running. If you run a SAP router/web dispatcher in your environment, then remove the comment sign (#) at the beginning of the lines with the `SAPROUTER/SAPWEBDISP` specific parameters to be able to create TSA resources for SAP router and/or SAP web dispatcher. As each such utility is a floating resource it needs its own floating IP. Make sure that you have a routing table (`saproustab`) for the `SAProuter` accessible by all nodes in the cluster. Same is true for the `SAP Web Dispatcher` profile.

It is important to realize that the number of entries in `NODES`, `ASNOS`, and `INSTDIRS` must be the same, because the sample policy assumes the following:

- the application server with the first instance number (02) and first instance directory (`DVEBMGS02`) runs on node 1 (`ihlsc0h1`).
- the application server with the second instance number (03) and second instance directory (`D03`) runs on node 2 (`ihlsc0h2`).
- the application server with the second instance number (04) and second instance directory (`D04`) runs on node 3 (`ihlsc0h3`)

and so on.

The `J2EE_instances.conf` file contains the following entries, which you must adapt to your SAP environment. Please comment out all other entries by adding a # in front.

Customizing the Tivoli System Automation for Multiplatforms (Base)

```
INSTALL_DIR="/usr/sbin/rsct/sapolicies/sap"      # installation directory of SAP policy scripts
CLUSTER="HA1_DOMAIN"                          # TSA domain name for below mentioned nodes; not used in HA scripts
NODES="ihlscoh1 ihlscoh2 ihlscoh3"            # list of nodes included in the TSA domain

PREF="SAPECC_JAVA"                            # prefix of all TSA SAP J2EE resources
# for J2EE instance we recommend to use <common prefix>_J2EE
ABAP_PREF="SAPECC_ABAP"                      # prefix of all TSA SAP ABAP resources; it is only needed, if the J2EE
                                             # application server is part of an 'double-stack' application server (like for XI);
                                             # it must have the same value as PREF in ABAP_instances.conf

SAPSID="HA1"                                  # SAP system ID
SAP_ADMIN_USER="ha1adm"                      # SAP administration user ID

ENQSRV_IP="10.101.5.195"                      # SCS instance IP address
ENQSRV_HOSTNAME="ha1scsv"                   # virt. hostname for SCS IP address; if not defined, comment out or set to ""
ENQSRV_IP_NETMASK="255.255.255.0"          # SCS instance IP address' netmask
ENQSRV_IP_INTERFACE="eth0"                 # interface on which SCS instance IP address is activated on each node as alias
ENQNO="01"                                  # instance number of the SCS instance
ENQDIR="SCS01"                              # instance directory of the SCS instance

ASNOS="02 03 04"                            # list of instance numbers of the SAP App. Servers
INSTDIRS="DVEBMGS02 D03 D04"               # list of instance directories of the SAP App. servers

# setting J2EE_HTTP_PORTS indicates, that profile is for J2EE
J2EE_HTTP_PORTS="50200 50300 50400"        # list of J2EE instance HTTP ports (default: 50000+instance number*100 +0)
```

Figure 58. Entries to be adapted in the `J2EE_instances.conf` file

Entries which differ from the ABAP profile are:

- PREF as we need a different prefix for Java resources
- ABAP_PREF as we need to know the prefix for ABAP resources (in case that the Java application server is part of a 'double-stack' application server we need to know the ABAP resources prefix in order to be able to generate a StartAfter relationship between the Java instance and the ABAP instance)
- All entries starting with ENQSRV_, ENQNO and ENQDIR as we have for Java an own SCS instance
- J2EE_HTTP_PORTS as each double-stack Java instance needs an http port. -

The number of entries in NODES, ASNOS, and INSTDIRS must be the same, because the sample policy assumes the following:

- the Java application server with the first instance number (02) and first instance directory (DVEBMGS02) runs on node 1 (ihlscoh1)
- the Java application server with the second instance number (03) and second instance directory (D03) runs on node 2 (ihlscoh2)

- and so on.

Step 2: Run the mksap script to create SA MP resources for components A, J and I:

Important

Before you create the highly available SAP system, make sure that the NFS server is running and the NFS mounts necessary for SAP are either active or, if you use the automounter, can be mounted dynamically. If you decided to make the NFS server highly available via the SA MP NFS server HA policy, this is the time you need to start the NFS server resources.

Step 2a. Create resources for ABAP components and optional SAP utilities

The next step is to run the mksap script to create the SA MP resource groups, resources, relationships and equivalencies for ABAP resources and optionally for the SAP utilities. As the root user, you have to run the mksap script multiple times with different options. Change to your install path of the sample policy script. The

Customizing the Tivoli System Automation for Multiplatforms (Base)

mksap script supports the following command line options:

```
./mksap -h
Usage: mksap [-h][-s][-c ConfigFile][-a NodeName][-t Type][-i InstType][-l ABAPCharSet][-n NodeName[,NodeName]][-v VirtASHostName]
mksap creates (parts of) the sample TSA policy for a highly available SAP environment.
-h, Print this help.
-s, Use static equivalencies for sample TSA policy (not recommended); default: dynamic equivalencies.
-c, Use configuration file ConfigFile (fully qualified or in current dir); default: saphasalinux.conf in current dir.
-a, Create TSA policy entries for a SAP application server on NodeName and add them to an existing TSA policy.
-t, Define the type of the sample TSA policy to be created; valid types: CentralServices|SAPCommon|AppServer|StandardSCS|ABAPCI;
    default: None, create CentralServices, SAPCommon and AppServer. AppServer and Standard SCS create fixed resources.
-i, Define the installation type; valid types: ABAP|ADDIN|J2EE; default: create entries for ABAP stack.
-l, Define the character set used by ABAP stack for ADDIN installation type; valid sets: ASCII|UNICODE; default: UNICODE.
-n, Create TSA policy entries for all nodes specified; default: create entries for all nodes defined in the configuration file.
-v, Use specified virtual hostname for the AppServer resource/rg and as parameter for Start/Stop/Monitor command; default: None.
-o, Create TSA policy entries for ASCS with ES and MS only (like SAPinst created ASCS); default: None, create ES,MS,GW,CO,SE.
```

Figure 59. mksap options

Procedure:

1. First run (as root):

```
mksap -c ABAP_instances.conf -t CentralServices -i ABAP
```

This creates the SA MP resources, etc. for the ABAP SCS.

If you have decided not to run the 3 optional services in the ASCS, then run the command with the option `-o`:

```
./mksap -c ABAP_instances.conf -t CentralServices -i ABAP -o
```

2. Run as root user the following command. You must adapt the node name. If you have installed the Application server with its own virtual hostname, you must in addition use the `-v<virtual hostname>` option.

```
./mksap -c ABAP_instances.conf -t AppServer -i ABAP -n ihlscoh1
```

This step creates the SA MP resources, etc. for an ABAP application server on node ihlscoh1. Then run the mksap command for each Application server.

Notes:

- a. If you have installed the application server with its own virtual hostname and you want to move the AS from one node to another and you have defined a mount point resource (MPR) for its instance directory in order to also move this MPR, then you must change the sapctrl_as script. Change: `exit ${STATUS_FAILED_OFFLINE}` to `exit ${STATUS_OFFLINE}`, where the comment indicates "# if this is for a floating appl. server, which has a floating mount point then change to `exit ${STATUS_OFFLINE}`".
 - b. Please note, that such a floating Application server is not a recommended setup.
3. *Optionally* run the mksap script to create the TSA for MP resources, etc. for the SAP utilities. Run this step also for a Java only system, if you want to run a SAP Web Dispatcher under TSA MP control: Run:

```
./mksap -c ABAP_instances.conf -t SAPCommon
```

This creates the SA MP resources, etc. for the SAProuter and/or SAP Web Dispatcher if the associated entries in the configuration file are active (not commented out).

Step 2b: Create resources for Java components

1. As the root user, run the mksap script again to create the SA MP resources, etc. for Java. In this case, use the configuration file J2EE_instances.conf:

```
./mksap -c J2EE_instances.conf -t CentralServices -i ADDIN
```


Customizing the Tivoli System Automation for Multiplatforms (Base)

This creates the SA MP resources, etc. for the Java SCS.

Before you proceed to create the 'double-stack' Java application server resources, make sure that the corresponding ABAP application server resources have already been created.

2. If you have installed the Application server with its own virtual hostname you must in addition use the `-v<virtual hostname>` option. Run the `mksap` with adapted node name:

```
./mksap -c J2EE_instances.conf -t AppServer -i ADDIN -n ihlscohl
```

This creates the SA MP resources, etc. for a Java application server on node `ihlscohl`. It also generates the `StartAfter` relationships between the Java application server and its corresponding ABAP application server. Be aware that this relation can only be created if the ABAP application server has been created previously. Run the `mksap` command for each Application server.

For a Java-only system: you must use the `-i J2EE` option:

```
./mksap -c J2EE_instances.conf -t CentralServices -i J2EE
```

```
./mksap -c J2EE_instances.conf -t AppServer -i J2EE -n ihlscohl (-v <virtual hostname>)
```

If you have installed and activated the SA MP NFS server HA policy within the SAP cluster, to ensure that the NFS server HA policy is always activated before the SAP policy you should remember to perform:

- "Step 3: Adapt the dependency file to create dependencies to the NFS server," and
- "Step 4: Run the commands in the dependency file"

Step 3: Adapt the dependency file to create dependencies to the NFS server:

This step is only needed, if you run your highly available NFS server within the SAP domain, in other words together in the cluster with the SAP resources.

The sample policy comes with a sample dependency file `sap_StartAfterRel_nfs_etc`. We recommend saving the original files. Adapt the sample dependency file `sap_StartAfterRel_nfs_etc` to your environment. In order to be able to do this, you must know the SA MP SAP resource names. You can list all SA MP resources with the command `lssam`.

The sample `sap_StartAfterRel_nfs_etc` file contains SA MP commands to:

- replace the standard static NFS Server equivalency with a dynamic one.
- add relationships to start SAP after the NFS server.
- add a relationship to start SAProuter (or SAP Web Dispatcher) after the NFS server.
- add relationships for the ABAP application server to start after the NFS server.

Step 4: Run the commands in the dependency file:

Before you can run the commands within both files, you must stop all resources referenced therein. The resource groups and resources must be offline, because relationships can only be defined for offline resources and offline resource groups. This means:

1. `chrg -o offline -s "Name like '%'"`
2. All NFS and SAP resources must be offline before running `sap_StartAfterRel_nfs_etc`. Execute the adapted commands in the file with:
`./sap_StartAfterRel_nfs_etc`

Verify using the

Customizing the Tivoli System Automation for Multiplatforms (Base)

lsrel -Ab

command that all relationships defined in both files are indeed active.

Important

We have added a StartAfter relationship between the IP of the ABAP SCS and the NFS server group above. In case you start the NFS server at the same time as your SAP system, it can happen that the Enqueue Server (ES) starts before the mount to the profile directory has been performed by the automounter. Although the ES cannot access its profile, it does not fail under SAP kernel versions up to and including 7.00. In such a case, it starts with a default port of 3200, even if your ABAP SCS instance number is different. No application server can then connect to the ES, because it is listening to the wrong port. Furthermore, if you have an application server with instance number 00, it will not start, because the dispatcher cannot access its default port of service sapdp00, which is also 3200. So your SAP system does not start correctly. We recommend changing the script used to start the NFS server as follows. In

```
/usr/sbin/rsct/sapolicies/nfsserver/nfsserverctrl-server
```

add the following just before the end of the start case:

```
# sleep for 10 seconds to allow the automounter to mount  
sleep 10
```

in order to allow the automounter to perform the mount of the profile directory.

Other Points to consider: Another point to consider is that we have defined a StartAfter and a StopAfter relationship between each SAP application server resource group and the NFS server resource group: All application server start after the NFS server and the NFS server stops after all application server are stopped. This allows the application server (and other SAP resources) to stay up and running in a situation, when the NFS server is moved to the backup node in case of a failure. Be aware that this StopAfter relationship stops the application server in case of an intentional move of the NFS server either explicitly via:

```
- rgreq -o move SA-nfsserver-rg
```

or implicitly via

```
- samctrl -u a<node name>
```

when the NFS server is actually running on <node name>.

Alternatively you can replace the StartAfter and StopAfter relationships with a DependsOnAny relationship. Be aware that this stops the application server in case of an intentional move of the NFS server and in case of a NFS server failure.

Another possibility to consider is to define only a StartAfter relation. If you stop the NFS server at the same time as your SAP system, then it happens, that the NFS server stops faster than the application server (and other SAP resources) which still need access to NFS mounted directories during normal stop processing and the application server processes will not stop. To avoid this you have to change the script, which is used to stop the NFS server in the following way. In `/usr/sbin/rsct/sapolicies/nfsserver/nfsserverctrl-server` add in the stop case as first instructions:

Customizing the Tivoli System Automation for Multiplatforms (Base)

```
# delay stopping of the nfs server for 4 minutes in order to allow other SAP resources to stop...
sleep 240
```

in order to allow the application server and other SAP resources to stop. This ensures that the delay is longer than the stop command timeout of the Java application server. In addition you must increase the stop command timeout for the NFS server resource to 300sec. (60sec. default + 240sec.) with:

```
chrsrc -s "Name='SA-nfsserver-server'" IBM.Application StopCommandTimeout=300
```

Be aware this stops the ASCS during the sleep interval, if you take out the node from the cluster where ASCS etc. and NFS are running. Also a move request takes a long time to complete. Therefore we do not recommend this.

Step 5: Perform a quick test of the SAP policy: You can now start all your SAP resources by issuing the command:

```
chrg -o online -s "Name like '%'"
```

Check their status with:

```
lssam -top
```

If the Application server resources get into a 'stuck online' status you may need to change the sapctrl_as script to wait longer. After all resources are up, you can stop them with:

```
chrg -o offline -s "Name like '%'"
```

Step 6: Save the policy: To save the policy, run this command:

```
sampolicy -s <policy-file>.xml
```

where <policy-file> is a name you assign.

To *uninstall*, you can perform the step "Removing the HA policy" on page 223.

Verify your highly available double-stack installation: The following test simulates an unplanned outage of the ABAP and/or enqueue server comprises two scenarios. The first test is for the ABAP SCS and the second for the Java SCS.

Test unplanned outage of ABAP SCS:

Note: This is *not* applicable for a Java-only system.

Check on which machines your ABAP enqueue server and its replication server are running. Then use SAP transaction SM12 to generate entries in the enqueue table. This is described in Chapter 15, "Verifying your implementation on z/OS," on page 249. To simulate an unplanned outage, kill the ABAP enqueue server process via:

```
killall -9 es.sapL0P_ASCS02
```

Then SA MP must restart the ABAP enqueue server on the machine where the ABAP replication server is active. Perform the listed steps to verify that the enqueue failover was transparent for the SAP system. These steps are also described in Chapter 15, "Verifying your implementation on z/OS," on page 249.

Test unplanned outage of Java SCS: Check on which machines your Java enqueue server and its replication server are running. Use the SAP-provided enqt utility to verify, that enqueue replication works as expected. Run as <sapsid>adm the TSA for MPSA MP preconfigured installation directory (with your adapted enqt profile):

Customizing the Tivoli System Automation for Multiplatforms (Base)

```
enqt pf=enqt.pf_scs01 97
```

Output is similar to:

```
---REQ-----  
EnqId:          EnqTabCreaTime/RandomNumber   = 06.09.2005 00:06:19 1125957979 / 8563  
ReqOrd at Srv:  TimeInSecs/ReqNumberThisSec   = 09.09.2005 13:45:43 1126266343 / 1
```

The EnqId is the unique identifier of the enqueue server and its enqueue table and you should remember it. In addition, run:

```
enqt pf=enqt.pf_scs01 20
```

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.

To simulate an unplanned outage, kill the Java enqueue server process via:

```
killall -9 es.sapLOP_SCS01
```

Then TSA for MP must restart the Java enqueue server on the machine, where the Java replication server is active.

Run both enqt commands again and check if the EnqId and the enqueue table entries are the same as before the Java SCS failover. If so, you have verified, that the enqueue table replication is working.

Starting the SAP system

You can now start your entire SAP system by issuing the command:

```
chrg -o online -s "Name like '%'"
```

After a short delay, you should see the following output when calling:

```
lssam -top
```

```

SAP resources 03/08/06 16:22:19
XI_J2EE_EP0_lnxsapg_D95 Online (Online)
'- XI_J2EE_EP0_lnxsapg_D95 AS Online on {lnxsapg.boeblingen.de.ibm.com}
XI_J2EE_EP0_lnxsapd_D96 Online (Online)
'- XI_J2EE_EP0_lnxsapd_D96 AS Online on {lnxsapd.boeblingen.de.ibm.com}
XI_J2EE_EP0_lnxsapj_D97 Online (Online)
'- XI_J2EE_EP0_lnxsapj_D97 AS Online on {lnxsapj.boeblingen.de.ibm.com}
XI_J2EE_EP0_ENQREP Online (Online)
'- XI_J2EE_EP0_ENQREP_ERS Online on {lnxsapd.boeblingen.de.ibm.com}
XI_J2EE_EP0_ENQ Online (Online)
'- XI_J2EE_EP0_ENQ_IP Online on {lnxsapg.boeblingen.de.ibm.com}
'- XI_J2EE_EP0_ENQ_MS Online on {lnxsapg.boeblingen.de.ibm.com}
'- XI_J2EE_EP0_ENQ_ES Online on {lnxsapg.boeblingen.de.ibm.com}
XI_ABAP_SYS_ROUTER Online (Online)
'- XI_ABAP_SYS_ROUTER_IP Online on {lnxsapg.boeblingen.de.ibm.com}
'- XI_ABAP_SYS_ROUTER_SAPROUTER Online on {lnxsapg.boeblingen.de.ibm.com}
XI_ABAP_EP0_lnxsapg_D95 Online (Online)
'- XI_ABAP_EP0_lnxsapg_D95 AS Online on {lnxsapg.boeblingen.de.ibm.com}
XI_ABAP_EP0_lnxsapd_D96 Online (Online)
'- XI_ABAP_EP0_lnxsapd_D96 AS Online on {lnxsapd.boeblingen.de.ibm.com}
XI_ABAP_EP0_lnxsapj_D97 Online (Online)
'- XI_ABAP_EP0_lnxsapj_D97 AS Online on {lnxsapj.boeblingen.de.ibm.com}
XI_ABAP_EP0_ENQREP Online (Online)
'- XI_ABAP_EP0_ENQREP_ERS Online on {lnxsapg.boeblingen.de.ibm.com}
XI_ABAP_EP0_ENQ Online (Online)
'- XI_ABAP_EP0_ENQ_IP Online on {lnxsapg.boeblingen.de.ibm.com}
'- XI_ABAP_EP0_ENQ_SE Online on {lnxsapg.boeblingen.de.ibm.com}
'- XI_ABAP_EP0_ENQ_CO Online on {lnxsapg.boeblingen.de.ibm.com}
'- XI_ABAP_EP0_ENQ_GW Online on {lnxsapg.boeblingen.de.ibm.com}
'- XI_ABAP_EP0_ENQ_MS Online on {lnxsapg.boeblingen.de.ibm.com}
'- XI_ABAP_EP0_ENQ_ES Online on {lnxsapg.boeblingen.de.ibm.com}
SA-nfssserver-rg Online (Online)
'- SA-nfssserver-data-work Online on {lnxsapg.boeblingen.de.ibm.com}
'- SA-nfssserver-data-varlibnfs Online on {lnxsapg.boeblingen.de.ibm.com}
'- SA-nfssserver-ip-1 Online on {lnxsapg.boeblingen.de.ibm.com}
'- SA-nfssserver-server Online on {lnxsapg.boeblingen.de.ibm.com}

```

Figure 60. List of SAP resources in SA MP policy

Your HA SAP system is now ready for use.

Alternatively, you can use the graphical user interface (GUI) provided with SA MP 2.3 or later to list and control the SAP resources. For more information, see "Using the operations console" in *Tivoli® System Automation for Multiplatforms Base Component User's Guide Version 2.1*, SC33-8210-05.

Verifying your high availability implementation with SA MP

To verify your HA setup, perform the steps listed in Chapter 16, "Verifying your implementation on Linux/AIX," on page 289. The described verification procedure and failover scenarios verify that SA MP keeps SAP resources available in the case of planned and unplanned outages.

The test that simulates an unplanned outage of the enqueue server comprises two scenarios, one for the ABAP SCS and one for the Java SCS.

Removing the HA policy

If the SAP system is no longer required, you can remove the policy from the cluster by issuing the following sequence of commands:

```
# chrg -o offline -s "Name like 'XI_%'"
```

Wait until all resources are offline, and then issue:

```
# ./rmsap -c J2EE_instances.conf  
# ./rmsap -c ABAP_instances.conf
```

Using a tie breaker with SA MP

The setup for a two-node scenario is the same as for a scenario with three or more nodes described in detail in the previous sections, except that in this case a tie breaker must be defined. This tie breaker is needed to decide if a node will survive or not in case of a cluster split. The tie breaker is not needed in normal operation where both nodes are up and running. But in an error condition, where one node cannot reach the other, it is not possible for the nodes to determine if the other node has crashed, or if only the network is broken. In this case, it is essential to protect critical resources such as IP addresses and data resources on a shared disk from being started or accessed from both machines at the same time. This is ensured by SA MP with the quorum functionality.

SA MP will only automate resources on a node that is a member of a subcluster having quorum. A subcluster has the quorum if the subcluster contains the majority of nodes. If the cluster consists of an equal number of nodes, and the cluster is split into two subclusters with each of the subclusters having half the number of nodes of the entire cluster, the quorum is in this subcluster, which wins the tie breaker. For more information about quorum and tie breaker, see *Tivoli System Automation for Multiplatforms on xSeries and zSeries: Guide and Reference*, SC33-8210, and the Reliable Scalable Cluster Technology (RSCT) documentation, available at:

<http://www.ibm.com/servers/eserver/clusters/library>

In conclusion, a tie breaker is strongly needed in a two node cluster. Otherwise, SA MP will not manage resources after a node failure or in case of a cluster split (network disruption).

There are two predefined tie breakers within the IBM.TieBreaker resource class: operator and fail. If the fail tie breaker is used, no subcluster will get quorum. The operator tie breaker, on the other hand, requires manual intervention from an operator who decides which of the two nodes will win the tie breaker and which will not. These two tie breakers are not useful from an automation point of view, because they do not provide an automatic grant of the tie breaker to one of the two subclusters.

However, SA MP allows the definition of a disk tie breaker. This must be a disk that is accessible from each of the nodes of the cluster. In case of a tie situation (network split or node failure) both subclusters try to access this tie breaker disk with a special mechanism (DASD reserve release). Only one subcluster can reserve the disk and then wins the tie breaker and, therefore, gets quorum. Note that all nodes running critical resources on the subcluster that did not win the tie breaker will commit suicide to protect the critical resources. In a two node cluster, the following situations can occur:

- Normal operation: Both nodes are up and can talk to each other.
- Crash of 1 node: The surviving node is in a tie, but will win the tie breaker.
- Network split: Both nodes try to access the tie breaker. One will win and survive, the other will commit suicide if critical resources are currently running on that node.

The setup of a disk tie breaker is described in detail in the *Guide and Reference* publication for the release (see Table 40 on page 347)

Using SA MP Quorums with Linux on System z under z/VM

This section describes the rules and considerations related to using *Quorum nodes* if running SA MP on *Linux on System z* under z/VM and over multiple:

- CECs, and/or
- z/VM LPARs.

By default, SA MP allows each node to be a Quorum node, and expects only a single node to fail at any given point in time. That is because SA MP considers a node to be a self sufficient host/system. However:

1. A *Linux for System z Guest* is dependant on it's z/VM host.
2. The z/VM host is in turn dependant on the underlying System z LPAR and CEC.

By running several virtualized Guests under z/VM, an outage (planned or unplanned) related to the z/VM LPAR the Guests are running under will cause the outage of those Guests at the same point in time.

An SA MP Quorum is required in order for SA MP to function in the expected automated manner. During the outage of a System z CEC or z/VM LPAR, the resulting loss of several Linux for System z Guests could easily result in SA MP being unable to establish the required Quorum. As a result, no automation would take place on the remaining SA MP nodes. This would not be consistent with the aim of using SA MP to establish a Highly Available environment. Therefore:

- For a multiple System z CEC configuration, we highly recommend that a single SA MP Quorum node be assigned *per System z CEC*.
- For a multiple z/VM LPARs on a single System z CEC configuration, we highly recommend that a single SA MP Quorum node be assigned *per z/VM LPAR*.

You can set Quorum nodes using the following command for each "hostname":

```
chrsrc -s 'Name == "hostname"' IBM.PeerNode IsQuorumNode=1
```

You can set non-Quorum nodes using the following command for each "hostname":

```
chrsrc -s 'Name == "hostname"' IBM.PeerNode IsQuorumNode=0
```

You can verify the state of each node using the following command example:

```
ihlscoh2:~ # lsrpnode -QB
Name      OpState  RSCTVersion  Quorum  Tiebreaker
ihlscoh3  Online   2.5.1.2      No      No
ihlscoh1  Online   2.5.1.2      Yes     Yes
ihlscoh2  Online   2.5.1.2      Yes     Yes
```

In the above example, IHLSCOH1 was running on one System z CEC and IHLSCOH2-3 on another System z CEC.

Tie Breaking is required when Quorum nodes are unable to communicate with each other over the available network interfaces. It may be that there is a network problem, or that another node is dead. The problem is that SA MP (RSCT) can't be sure what the reason is: so a Tie-Breaker is used as a means of validating the operational state of the node we are running on.

The probability that a network problem exists, reduces with the:

- number of network interfaces that are used.
- redundancy of network equipment used.

Customizing the Tivoli System Automation for Multiplatforms (Base)

If you use a simple single network between all of your nodes, we highly recommend the use of an SA MP Tie-Breaker DASD which must be shared between all of the Guests across all z/VM LPARs and System z CECs.

- The Reserve/Release DASD commands are utilized 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 in order to allow RESERVE/RELEASE DASD commands to be issued.
- In the USER DIRECTORY of each Linux Guest should be a LINK statement with options MW to allow multiple write access.

The following commands need to be issued in order to make SA MP use the Tie-Breaker DASD *nnnn*:

```
mkrsrc IBM.TieBreaker Name=DasdTieBreaker Type=ECKD
DeviceInfo="ID=nnnn" HeartbeatPeriod=5
```

```
chrsrc -c IBM.PeerNode OpQuorumTieBreaker="DasdTieBreaker"
```

and can be verified using the following command:

```
lsrsrc IBM.TieBreaker
```

If you have multiple redundant networks between all of your nodes, we highly recommend the use of an *SA MP Network Tie-Breaker*. The reason is that the failure of multiple redundant networks is so unlikely that the cause of SA MP (RSCT) not being able to communicate with another node is most likely that that node is dead. In such a case the overhead of a DASD Tie-Breaker is really unnecessary, and a simple network ping to an external gateway suffices to ensure that our own node is operational.

The following commands need to be issued to make SA MP 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:

- SA MP Quorums are documented in *SA MP Administrator's and User's Guide*.
- The commands related to SA MP Quorums are documented in *SA MP Installation and Configuration Guide*.

Chapter 13. Customizing the Tivoli System Automation Application Manager (E2E)

This chapter describes the strategic implementation and design of the SA AM (E2E) product for the heterogeneous SAP on System z environment. The strategic tool to automate a heterogeneous solution environment is the IBM Tivoli System Automation for Application Manager (E2E) product.

Note: Up to release 2.3, the Tivoli System Automation for Application Manager (E2E) was a component of SA MP with the name SA MP End-to-End. This is now a separate product with the name **SA AM** (Automation Manager).

We provide guidance and recommendations based on the sample high availability SAP on System z environments already described in Chapter 11, “Customizing Tivoli System Automation for z/OS,” on page 171 and Chapter 12, “Customizing the Tivoli System Automation for Multiplatforms (Base),” on page 195.

SAP on System z is built around IBM DB2 for z/OS, which is used as the SAP database server. SAP application servers (ABAP or Java or ABAP+Java) are supported on several platforms, such as the 64-bit operating systems Linux on System z, Linux on System x, and AIX. The environment is therefore heterogeneous by nature. The SA AM (E2E) component is ideal for automating such an environment.

This chapter contains these main topics:

- “Overview of end-to-end automation management”
- “Sample high availability environment of the SAP on System z solution” on page 228

Overview of end-to-end automation management

The Tivoli System Automation (TSA) application manager provides you with *two* aspects of application and resource management.

- Monitoring and manual interaction with applications:
 - No e2e policy required.
 - The e2e *Integrated Solutions Console ISC* supports you with monitoring capabilities to monitor each resource you have defined in your SA MP and SA zOS policies.
 - You can actively observe the health state of your applications. If problems do occur, you can navigate down to the resource with an error in order to determine the cause of the problem, obtain additional information, and start/stop resources. For further information, follow the documentation links contained in Table 40 on page 347.
- Fully automated management of applications:
 - Requires e2e policy.
 - An e2e management policy defines your applications, resources, and the dependencies between them. The applications and resources will be managed *automatically*.
 - No manual intervention is required.

Customizing the Tivoli System Automation for Application Manager (E2E)

- Your applications and resources will start, stop, and move according to the rules you set in the e2e policy.
- ISC can help you to create and maintain your e2e policy. For further information, follow the documentation links contained in Table 40 on page 347.

End-to-end automation can be used to automate the operation of resources within heterogeneous environments (called *first-level automation domains*) that each have a local automation technology of their own. A first-level automation domain is defined either as a set of resources managed by SA MP or one managed by SA z/OS. Each first-level automation domain is connected to the end-to-end automation manager by an automation adapter.

The main elements of the SA AM product are the *SA management console* and the *end-to-end automation manager*:

- The *SA management console* is a Web-based front-end to the end-to-end automation domain and the first level automation domains (for example, resources managed by SA z/OS or SA MP).
- The *end-to-end automation manager* consists of the following:
 - Automation J2EE framework in WebSphere® Application Server (WAS)
 - Automation engine
 - Automation engine resource adapter
 - First-level automation manager resource adapter (communicates with the automation adapter)
 - End-to-end automation policy
 - Automation Database

For details about these elements, see the SA MP publications in Table 40 on page 347.

Terms that are used in discussing end-to-end automation have the following meanings in SA z/OS:

- *First-level automation domain* is the group of SA z/OS agents and managers that belong to the same XCF group ID (GRPID).
- *Node* is the SA z/OS equivalent of a system.

Sample high availability environment of the SAP on System z solution

The SAP on System z solution runs the database *on z/OS*. The application servers run on these *non-z/OS operating systems*:

- Linux on System z
- AIX on System p
- Linux on System x
- Microsoft® Windows on System x hardware (for Windows, there is currently no SA MP support).

The following figure shows a sample solution overview of HA with SAP on System z:

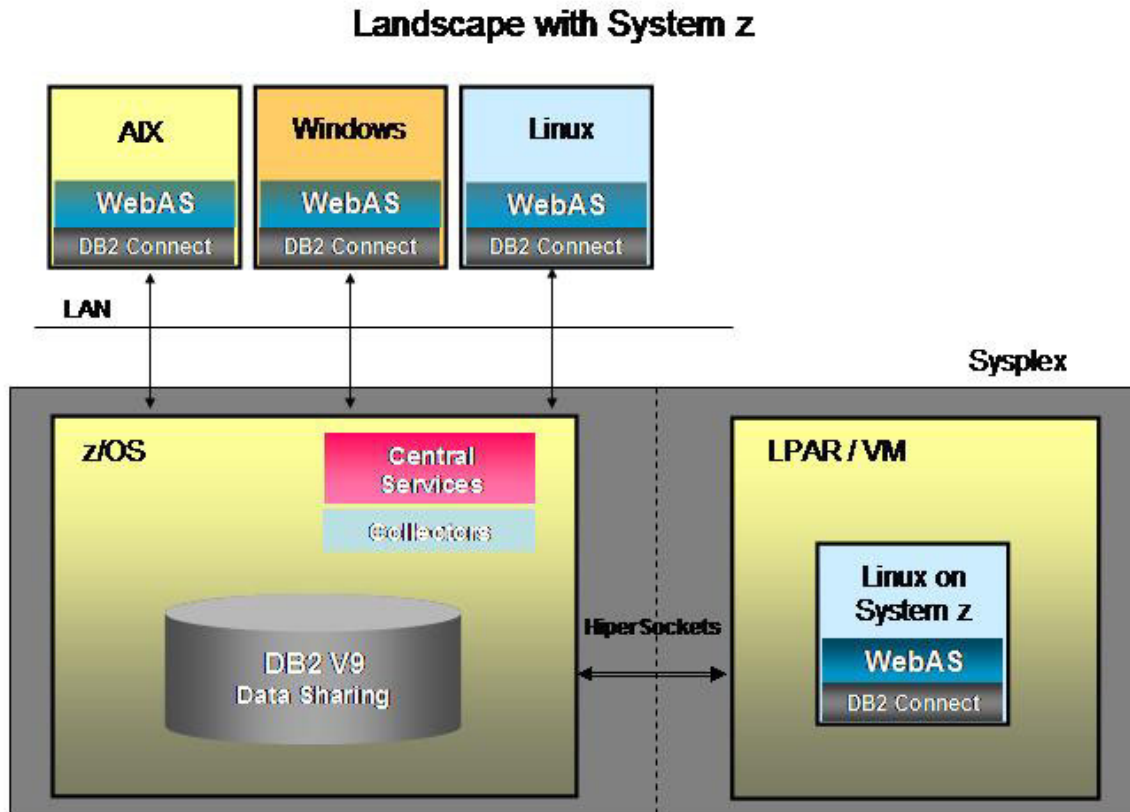


Figure 61. Landscape with sample HA solution of SAP on System z

HA with SAP on System z landscape is heterogeneous by design. SA AM (E2E) is the perfect management product for *heterogeneous system environments* because it:

- allows a heterogeneous SAP system to be easily and reliably started and stopped.
- gives you immediate problem detection across cluster boundaries.
- increases the availability of the SAP on System z solution by allowing dependencies between resources in *different* clusters.

We recommend running the most critical components of the SAP application under z/OS and UNIX System Services (USS) and therefore under control of SA for z/OS.

For High Availability to run under z/OS, we recommend:

- Three or more data sharing members for the DB2 database.
- SAP Central Services
- Infrastructure components such as the NFS server and SAPOSCOL.

Application servers and critical utilities should reside in application server nodes on 'remote' systems. Remote systems are managed by SA MP. *Critical utilities* could be, for example:

- SAProuter

Customizing the Tivoli System Automation for Application Manager (E2E)

- SAP Web Dispatcher
- SAPOSCOL

Setting up the end-to-end product

To install SA AM (E2E), see the appropriate SA MP publication in Table 40 on page 347.

In this chapter we use a sample policy that is the *recommended setup* for a *SAP on System z HA solution*. It is based on the following SAP system setup:

- A three LPAR z/OS cluster, running on two physical machines, is implemented on System z hardware. SA z/OS manages:
 - Three DB2 data sharing members, one in each LPAR, which do *not* move
 - SAP ABAP Central Services under USS (movable between nodes)
 - SAP Java Central Services under USS (movable between nodes)
 - Highly available NFS server under z/OS (movable between nodes)
 - Highly available SAPOSCOL, which collects sysplex-wide information (movable between nodes)
 - Highly available end-to-end adapter (movable between nodes)

A three-node Linux on System z cluster is implemented on System z hardware: SA MP manages:

- Three application servers, one on each node, which do *not* move
- Highly available SAProuter utility (movable between nodes)
- Highly available end-to-end adapter (movable between nodes)
- Three SAPOSCOLs, one on each node, which do *not* move

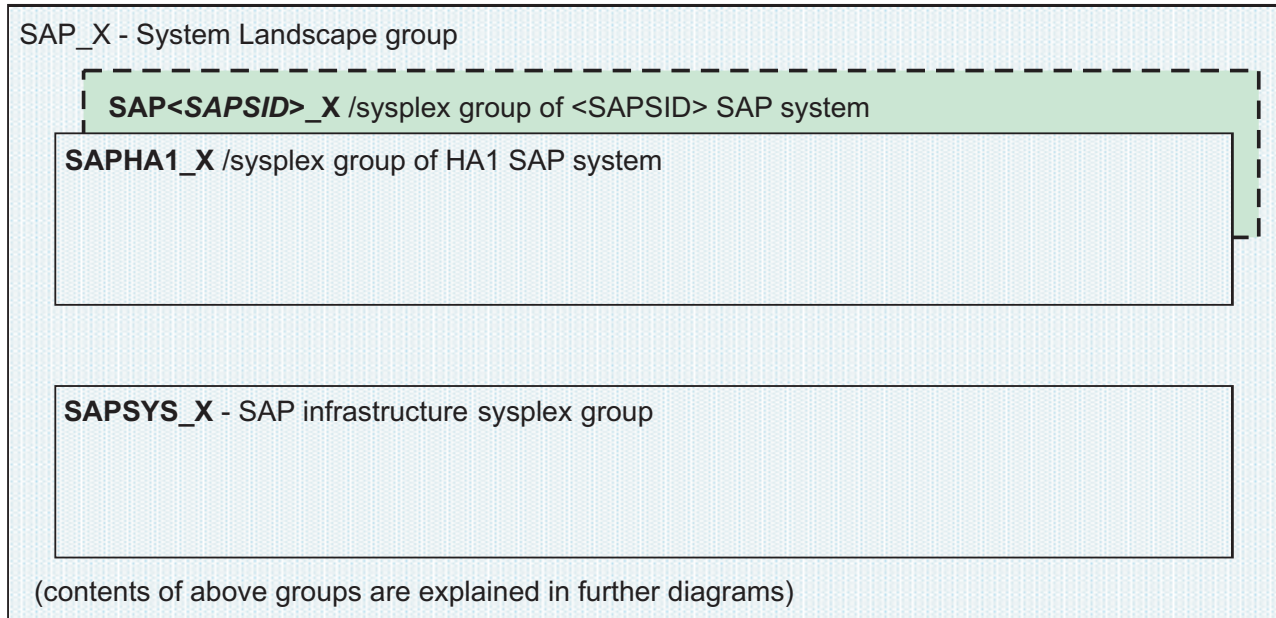
The SAP system ID is **HA1**. User <sapsid>adm is **ha1adm**, with home directory /home/ha1adm.

The following figures show the z/OS cluster and the SA z/OS resource groups and resources. Understanding of this structure is required in order to understand the references to the SA z/OS first-level domain in the end-to-end sample policy.

Customizing the Tivoli System Automation for Application Manager (E2E)

Figure 62 shows the top level **SAP_X** group. It includes:

- One group for SAP system specific resources (**SAPHA1_X**).
- One group for SAP system infrastructure resources (**SAPSYS_X**).



----- indicates optional resources / optional groups


 Sysplex/Basic active group

Figure 62. Best Practice SAP policy adapted for SAP system HA1

Customizing the Tivoli System Automation for Application Manager (E2E)

Figure 63 shows the details of the SAPSID-specific resources (SAPHA1_X group). The SAPSID-specific group comprises the groups for the:

- ABAP enqueue and message server group (SAPHA1AENQX).
- ABAP enqueue replication server group (SAPHA1AER_X).
- JAVA enqueue and message server group (SAPHA1JENQX).
- JAVA enqueue replication server group (SAPHA1JER_X).
- DB2 database server group (SAPHA1_DBX), which includes the DB2 resources (MSTR, DBM1, IRLM, DIST, and so on address spaces).

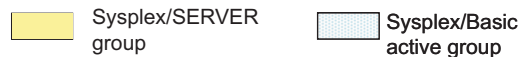
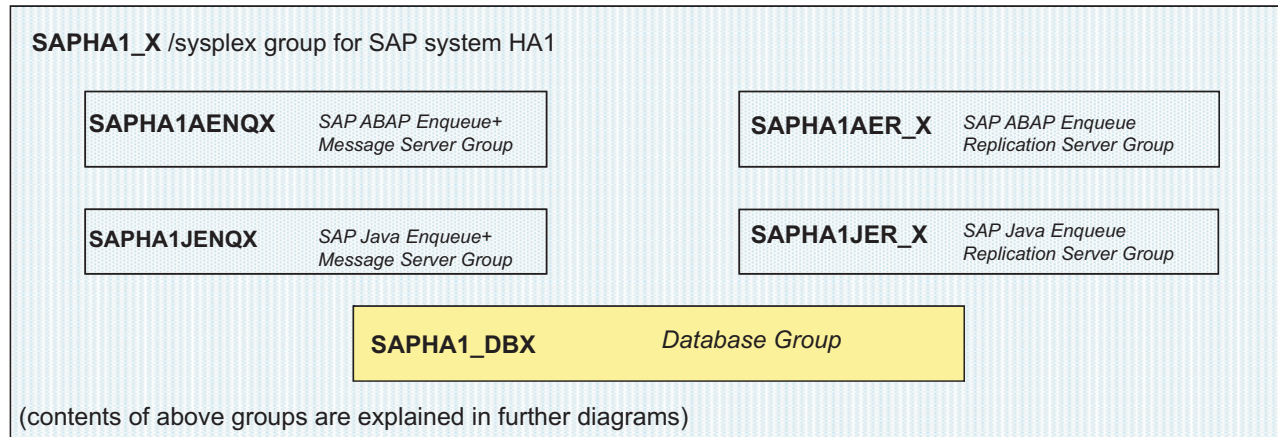


Figure 63. Groups and resources specific to SAP system HA1

Customizing the Tivoli System Automation for Application Manager (E2E)

Figure 64 shows the details of the SAP-system independent groups and resources. It contains these groups:

- The group required to perform SAP monitoring (SAPSYSOSC_X).
- The NFS server group (NFS_SERV_X).
- An optional group for the SAP router (SAPSYSRTE_X).

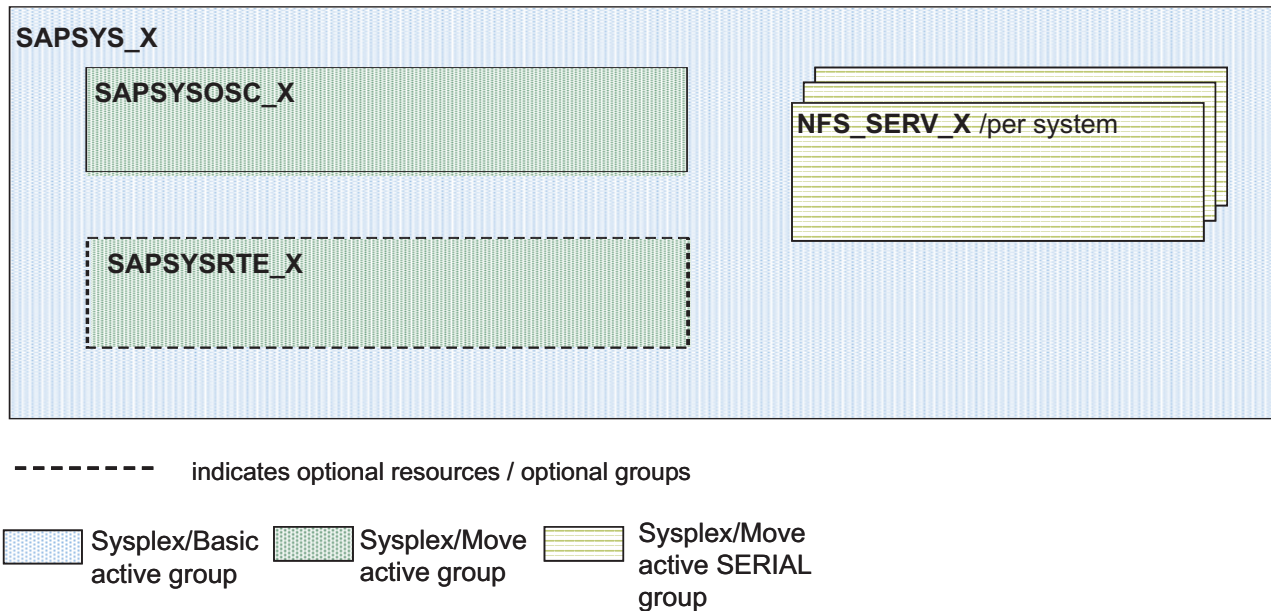


Figure 64. SAP infrastructure group

Customizing the Tivoli System Automation for Application Manager (E2E)

Figure 65 shows the *end-to-end* lowest level in the group structure of the ABAP central services and enqueue replication groups:

- SAPHA1AENQX contains the resources:
 - SAPHA1ASCSX
 - ABAP enqueue server
 - ABAP message server
- SAPHA1AER_X contains the ABAP enqueue replication server resource SAPHA1AER.

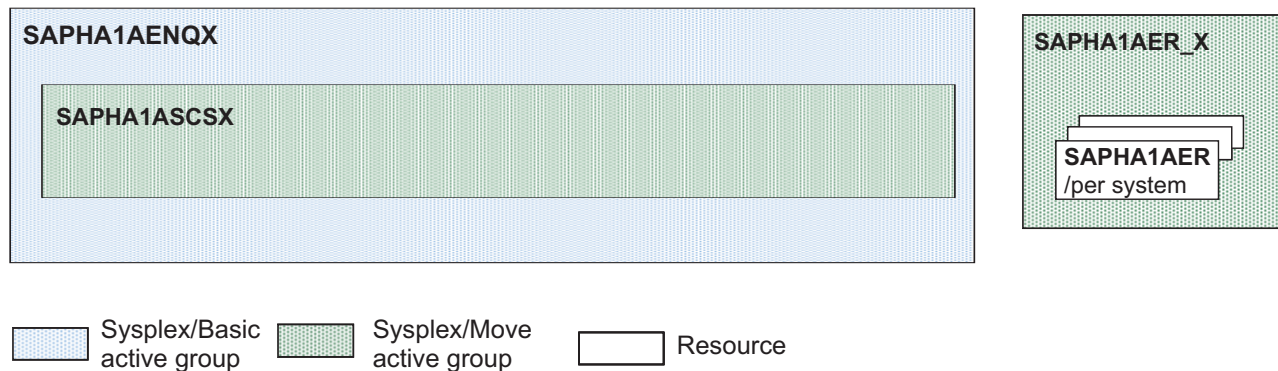


Figure 65. ABAP central services and enqueue replication groups

Figure 66 shows the *end-to-end* lowest level in the group structure of the JAVA central services and enqueue replication groups:

- SAPHA1JENQX contains the resources:
 - SAPHA1JSCSX
 - JAVA enqueue server
 - JAVA message server
- SAPHA1JER_X contains the JAVA enqueue replication server resource SAPHA1JER.

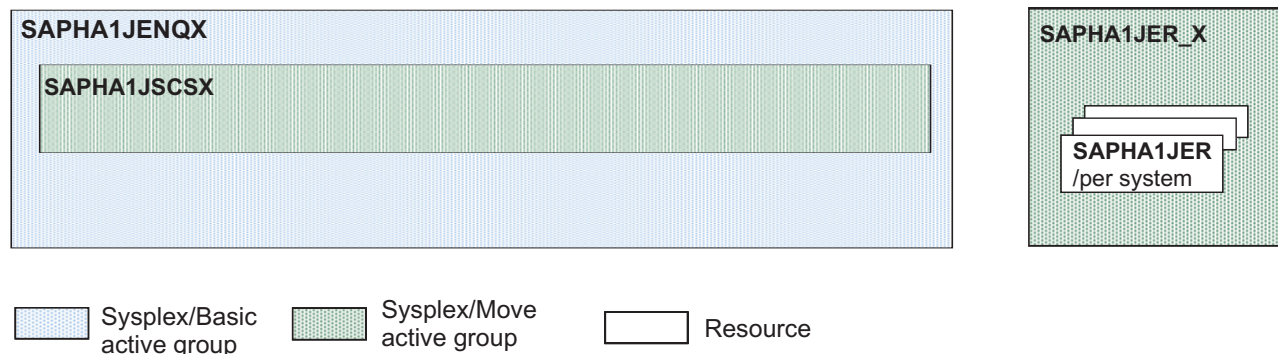


Figure 66. JAVA central services and enqueue replication groups

Customizing the Tivoli System Automation for Application Manager (E2E)

Figure 67 shows the DB2 sysplex group that has been adapted for the HA1 SAP system.

SAPHA1_DBX – SYSPLEX SERVER group containing 3 DB2 datasharing members for SAP System HA1

 Sysplex/SERVER active group

Figure 67. DB2 sysplex group – adapted for SAP system HA1

Figure 68 shows the Linux on System z cluster and the SA MP resource groups and resources for this SAP on System z sample high availability solution.

Understanding of this structure is necessary to understand the references in the end-to-end sample policy to the SA MP first-level domain. The policy includes:

- Three fixed application servers, one on each node
- One end-to-end adapter, movable between the nodes
- Three fixed SAPOSCOLs, one on each node

A light background indicates a floating resource that is currently active. For example, End-to-end adapter is active on node ihlsco2.

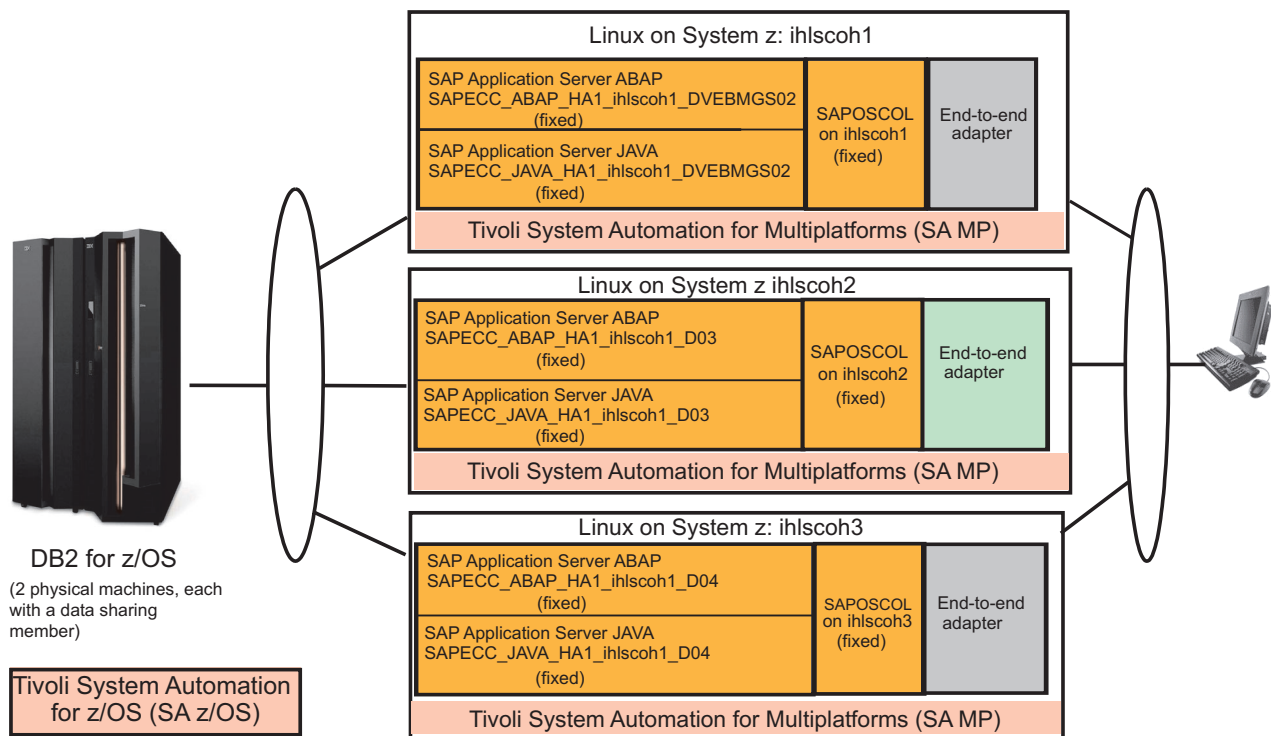


Figure 68. Linux on System z cluster

Based on the first-level domains for SA z/OS and SA MP, the sample end-to-end policy contains resource references:

- "SAPHA1 AppServer HA1 ABAP on ihlsco1" to "SAPHA1 Application Server Group ABAP".

Customizing the Tivoli System Automation for Application Manager (E2E)

- "SAPHA1 AppServer HA1 JAVA on ihlsc0h1" to "SAPHA1 Application Server Group Java" and similarly for each additional application server resource reference.
- "SAPHA1 ABAP Central Services Sysplex Group" and "SAPHA1 ABAP Enqueue Replication Server Sysplex Group" to the "SAPHA1 sysplex group".
- "SAPHA1 Java Central Services Sysplex Group" and "SAPHA1 Java Enqueue Replication Server Sysplex Group" to the "SAPHA1 sysplex group".
- "SAPHA1 NFS Server Sysplex Group" and "SAPHA1 SAPOSCOL Sysplex Group" to the "SAPHA1 infrastructure group".
- "SAPHA1 DB2 Sysplex Group" to the "SAPHA1 sysplex group".

The ABAP application server references are associated in an end-to-end group "SAPHA1 Application Server Group ABAP". This group has StartAfter relations to

- "SAPHA1 ABAP Central Services Sysplex Group"
- "SAPHA1 NFS Server Sysplex Group"

The Java application server references are associated in an end-to-end group "SAPHA1 Application Server Group Java". This group has one StartAfter relation to "SAPHA1 Java Central Services Sysplex Group".

These two groups together with the all other SAP HA1-system-specific resource references are included in a end-to-end group "SAPHA1 sysplex group".

Resource references that are independent of SAP system HA1 are included in a end-to-end group "SAP infrastructure group".

This grouping enables you to add more SAP<SAPSID> systems. Simply add another "SAP<SAPSID> sysplex group" and share the SAP infrastructure group. Optionally, you can put these high level groups into the end-to-end group "SAP Domain".

SA MP automates 3 Linux on System z systems (ihlsc0h1,2,3)

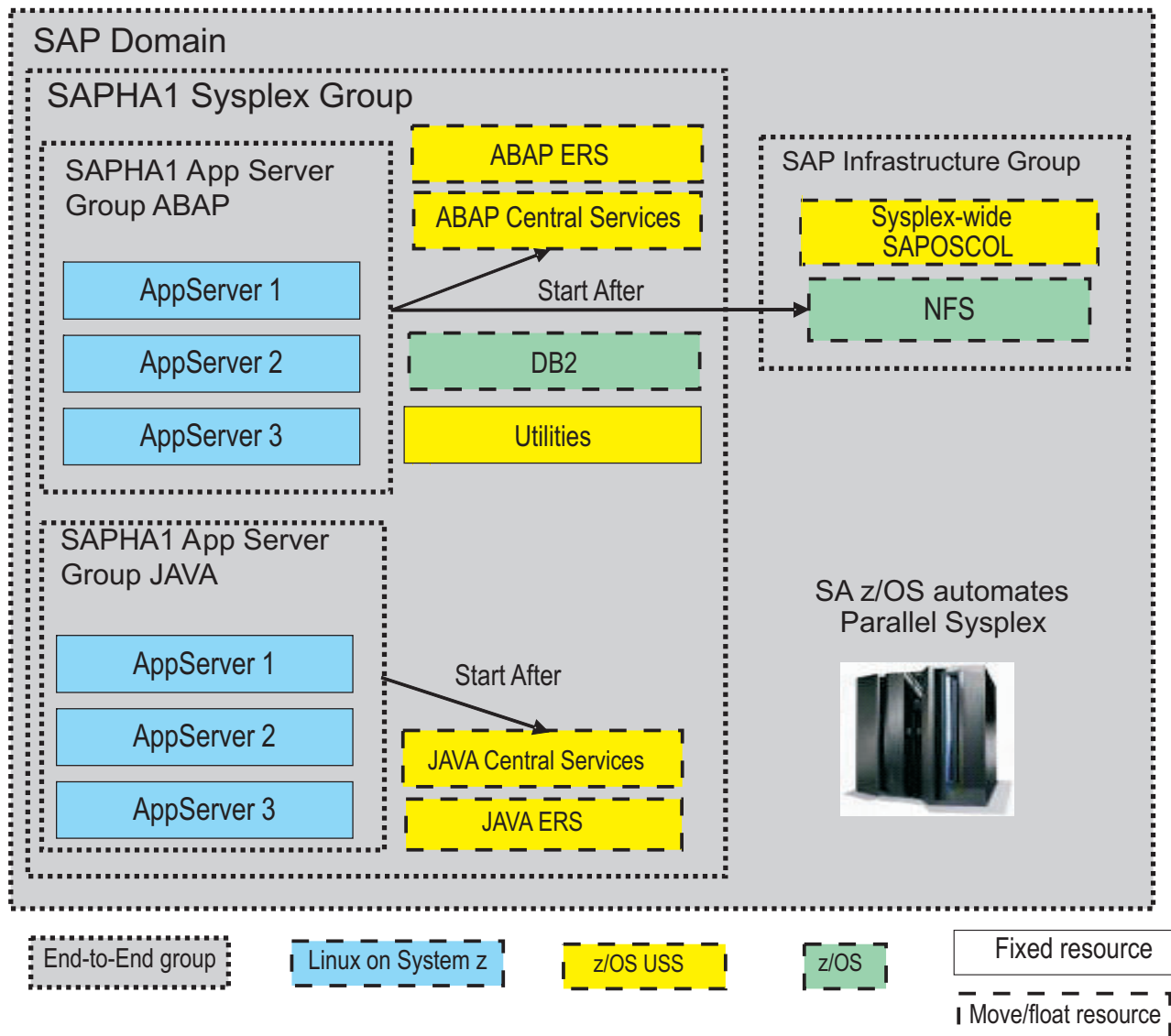


Figure 69. Overview of the end-to-end policy definitions

Defining and installing the end-to-end high availability policy for SAP

Defining and installing an end-to-end policy is described in the *IBM Tivoli System Automation Application Manager, Administrator's and User's Guide* (see Table 40 on page 347). You can either:

- define your own end-to-end policy from scratch, or
- download an end-to-end sample policy from:

<http://www-03.ibm.com/servers/eserver/zseries/software/sap/automation/>

The end-to-end sample policy is provided on an “as is” basis. You must tailor it to meet your SAP system configuration. Move your local end-to-end policy into the policy pool on your end-to-end automation manager system directory.

Customizing the Tivoli System Automation for Application Manager (E2E)

By default, the *policy pool directory* is:

- For AIX and Linux: /etc/opt/IBM/tsamp/eez/policyPool
- For z/OS: /<SYSTEM>/etc/eez/policyPool

The actual activation is done via the user interface as described in the cited publication. During policy activation, a syntax check is done. Make sure that all errors are resolved before actually activating the policy. If you no longer need the policy, you can either inactivate it or activate another one.

The user interface depicts the above described sample policy as shown in the following two graphics:

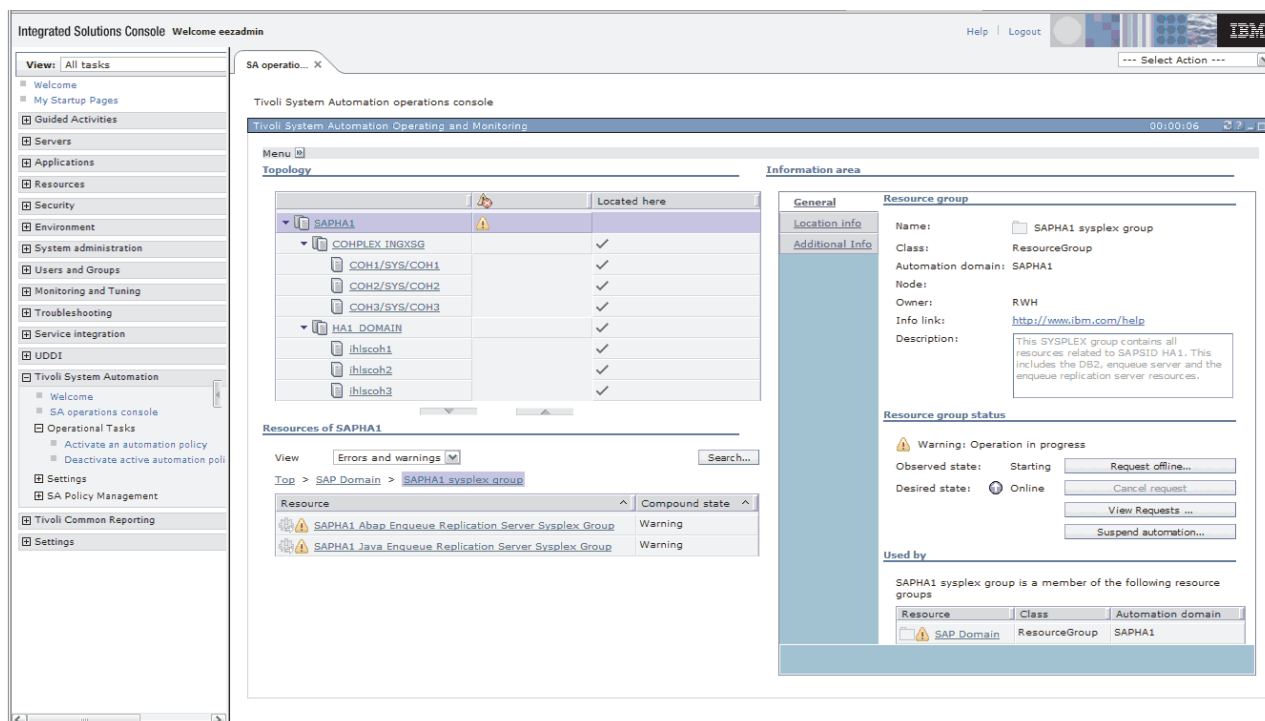


Figure 70. SA Operations Console view on SAP System Topology

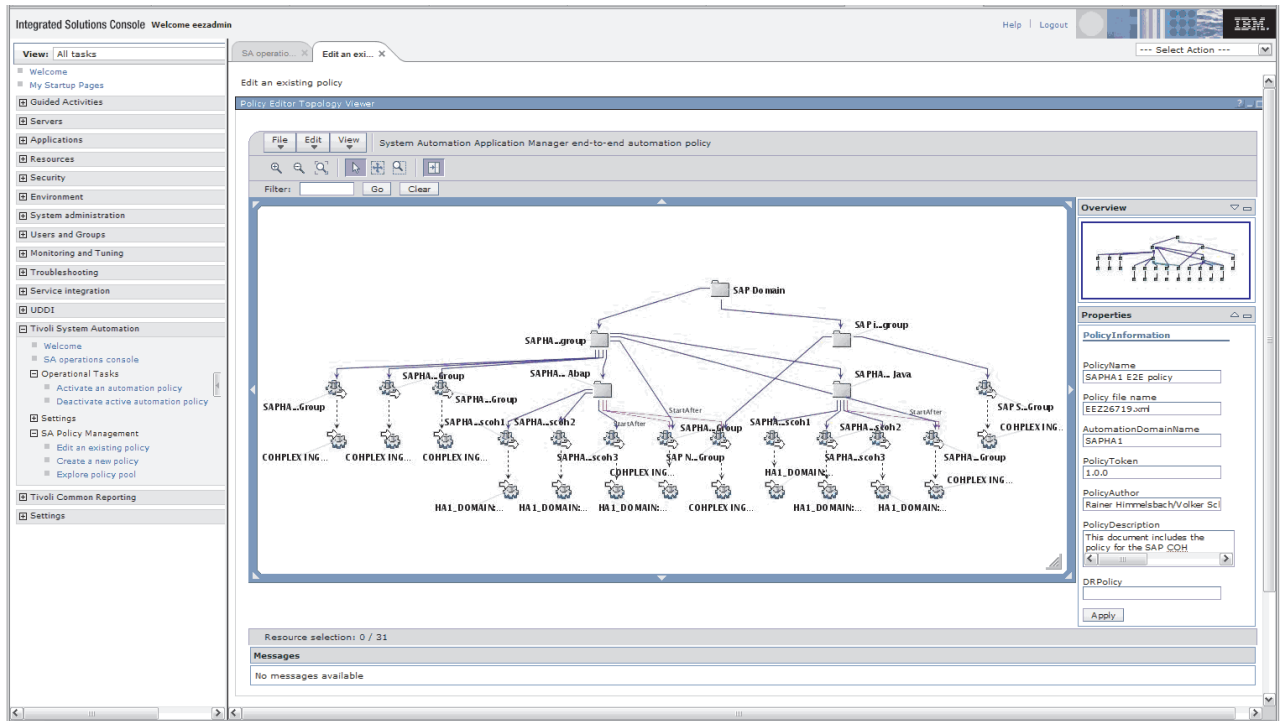


Figure 71. Graphical view of the SAP SA end-2-end policy

Chapter 14. Change management

This chapter discusses the procedures to update the SAP kernel. It also discusses procedures on how to update DB2 and z/OS with minimal impact on the SAP application using z/OS Parallel Sysplex and DB2 data sharing.

This chapter contains these main topics:

- “Updating the SAP kernel”
- “Rolling update of DB2 Connect” on page 244
- “Updating DB2 or z/OS” on page 245

Updating the SAP kernel

It’s important for an SAP system that all application server instances use the same kernel level. For this reason, SAP has implemented a checking mechanism to ensure consistent kernels. In this section, we describe this mechanism in detail so you can understand why kernel updates have to follow a specific sequence.

- Each application server instance registers at the message server. The connection is established by the dispatcher. The dispatcher informs the message server—among other things—about the platform type (for example z/OS, Linux on System z, or AIX 64-bit) and its own patch level.
- The message server stores the patch level of the application server instance that connected first, but separately for each platform type. The value pairs—platform type plus corresponding patch level—are kept in memory as long as the message server is running. The values are *never* reset.
- When another instance registers later, the stored patch level for the corresponding platform is returned by the message server. If the dispatcher of that application server instance detects a mismatch, it stops.

Although SAP strongly recommends that the patch levels of all application server instances are identical, the checking mechanism enforces this rule only among instances of the same platform type. The reason for this is that sometimes a patch level is not available for all platforms.

While using the old central instance concept, this mechanism is very reasonable. The message server is started and stopped with the central instance. Therefore, the stored patch level is the one of the central instance.

However, with the new concept there are some implications. The application server instances might connect in arbitrary order. Furthermore, they are started and stopped independently of the message server. A new patch level for the instance (disp+work) usually does not affect the message server nor the enqueue server.

Beginning with kernel release 4.6D_EXT, SAP has introduced the *rolling kernel upgrade*. This concept handles the implications previously described, and is well suited for the HA environment. See “Rolling kernel upgrade” on page 243 for more information.

Note: A rolling kernel upgrade is not yet available for SAP Web Application Server 6.xx.

Updating the SAP kernel (release 6.40 or later)

As described in the preceding section, the first application server instance that connects to the message server defines the patch level. Application server instances that connect afterwards must match the same patch level. The patch level is fixed until the message server is restarted.

Updating the SAP kernel

If the SAP kernel (disp+work or one of its dynamically-loaded modules) is to be updated, then perform the following steps. The sequence is applicable for UNIX systems including z/OS:

1. Save the old modules, which reside in the executable (exe/run) directory, and copy the new modules to this directory.
2. Stop all application server instances. Wait until all application servers are down.
3. Then stop and restart the message server.
In SA, this is accomplished by a STOP command with RESTART option set to YES .
4. Finally, start the application server instances again.
In SA, this is done by cancelling the STOP votes.

Note: On Windows, load modules cannot be replaced while they are in use. Therefore, first stop the application server instance before replacing the executables and dynamic load modules. On UNIX, shared objects (*.so) are locked and cannot be overwritten while they are in use. However, they can be renamed or moved to another directory.

Updating the enqueue server or replication server, or changing the size of the enqueue table

Updating components of SCS is quite easy, and it is transparent to the rest of the system.

If you want to update the enqueue server (enserver), simply let it fail over to the other system:

1. Save the old module which reside in the executable (exe/run) directory and copy the new module to this directory.
2. Move SCS to the system where the enqueue replication server is running.
In SA, this is accomplished by a STOP command on the enqueue server. Since the enqueue server is member of a MOVE group, it is automatically restarted on the system where the enqueue replication server is running on. Cancel the STOP vote afterwards.

If you want to increase the size of the enqueue table, you can take the same approach:

1. Modify the SCS profile.
2. Move SCS to the system where the enqueue replication server is running.

If you want to update the enqueue replication server (enrepsrver), perform these steps:

1. Save the old module, which resides in the executable (exe/run) directory, and copy the new module to this directory.
2. Then stop and restart the enqueue replication server.

In SA, this is accomplished by a STOP command with RESTART option set to YES. Afterwards, cancel this vote.

Applying SAP/other maintenance when SAP is controlled by SA MP

If you need to install Support Package Stacks (apply SAP maintenance) with the SAP installer tool SAPInst, you must:

1. Start your SAP system with SA MP.
2. Set SA MP into manual mode with

```
samctrl -M T
```

When SA MP is running in this mode, it will monitor all resources in the cluster, but it will not take any actions to start or stop resources. With `lssamctrl` you can check the actual mode.

3. If SAPInst complains that it cannot stop the (A)SCS instance, then go to the (A)SCS instance work directory and delete the file `kill.sap`.
4. After the maintenance is installed, set SAP into the automatic mode again:

```
samctrl -M F
```

If you need to install other maintenance at a node while SAP is not running or you need to reboot a node of your cluster, then please stop all resources before and 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. Please read *Important Remarks to NFS server* above, if the NFS server is currently running on the node which should be excluded, in order to understand the possible consequences to running SAP application servers on other nodes in the cluster.

You can check if a node is currently contained in the excluded node list of the cluster with the `lssamctrl` command. If it is excluded and all resources are offline, then you can apply your maintenance, reboot, 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, SA MP will do nothing to the node. Join the node to the cluster again with

```
samctrl -u d <nodename>
```

Rolling kernel upgrade

The concept described here is valid with SAP kernel release 4.6D_EXT and is planned for 6.x and future releases. It allows you to upgrade the kernel patch level on your application servers without the need to stop all your application servers and thereby generate a planned system outage. It allows you to keep your SAP system running while upgrading the kernel patch level of your application servers.

Note: This applies **only** if you are running your system with the standalone enqueue and enqueue replication servers. In other words, running SCS is a prerequisite. See SAP Note 953653 - Rolling Kernel Switch for further information.

Each patch of the 4.6D_EXT kernel is classified by two numbers, the *update level* and the *patch number*:

Change management

Update level:

Only kernel patches with the same Update Level can be used concurrently in a single SAP system. SAP will bundle incompatible changes. You can expect that a kernel patch with an incremented update level will be shipped once per year. This happens if a communication protocol has changed or the ABAP runtime has a significant change. In this case, proceed as described in “Updating the SAP kernel” on page 242.

Patch number:

If a kernel patch is compatible with its predecessor, only the kernel patch number will be incremented. The kernel patch can be installed and activated in rolling fashion on the application servers by restarting one application server after the other, rather than shutting down the system.

You can perform a rolling kernel upgrade based on a compatible patch as follows:

1. Save the old modules, which reside in the executable (exe/run) directory, and copy the new modules to this directory.
2. Stop and restart the application server instances. This can be done one after the other, or all at the same time.

The rolling kernel upgrade does *not* mean that the SAP system should run for a longer time with different patch levels. The rolling kernel upgrade should preferably be done while there are no active users or batch jobs. Stopping an instance implies that logged-in users have to reconnect and transactions which run on that instance are rolled back.

Rolling update of DB2 Connect

Updating DB2 Connect to a certain FixPak level causes downtime of the SAP application server during the update process. Therefore, we recommend that you update DB2 Connect on each application server, one at a time in sequence.

To install the DB2 FixPak, ensure that:

- You have root authority.
- You have a copy of the DB2 FixPak image downloaded from the SAP Marketplace:

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

1. Navigate to **Download -> Database Patches (from other vendors) -> DB2 for z/OS**.

Note: SAP strongly recommends obtaining all DB2 FixPaks through SAP. *SAP strongly discourages customers from downloading DB2 FixPaks from official IBM Web sites unless explicitly advised to do so by the SAP Support Team.* The SAP supported FixPaks may differ from the ones on the IBM Web site or may not be available from the IBM Web site.

2. Uncompress the FixPak image into a specified temporary directory.

A detailed description of how to install a DB2 Connect FixPak is contained in the readme file of the applicable FixPak, FixPackReadme.txt. To install the DB2 FixPak for Enterprise Server Edition (ESE) on a Linux or UNIX operating system, the following basic steps are required:

1. Stop the SAP application server.
2. Stop all DB2 processes.

3. Change to the temp directory in which the installation image is located and enter the `./installFixPak` command to launch the installation.
4. After the installation, the instance must be updated.
5. Restart the instance.
6. Re-bind the bind files once for a FixPak level, as described in the readme file for the FixPak, `FixPackReadme.txt`.
7. Restart the SAP application server.

Updating DB2 or z/OS

DB2 and z/OS can be updated by applying software maintenance, or upgrading to a newer release of the software. Applying software maintenance is done more often than upgrading the software. Software maintenance can be used to improve the performance of a software feature, or to fix software problems. In some special cases, new features can be added using software maintenance. SMP/E is the system tool used to apply, upgrade, track, and manage software for all z/OS products, including DB2 and z/OS.

At a very high level, SMP/E builds target executable libraries (loadlibs) while the software is executing from different executable libraries. In order to activate the latest changes, the software (z/OS or DB2) must be stopped and restarted using the updated loadlibs. For more detail on how to apply software maintenance using SMP/E, refer to the SMP/E User's Guide for the release of z/OS you are running.

Both DB2 and z/OS support *downward compatibility*. This means, you can run multiple software releases in a Parallel Sysplex data sharing environment. z/OS normally supports N+2 releases and sometimes N+3 - *please check for compatibility PTFs*. This means, that up to four consecutive releases can run in the *same* Parallel Sysplex: for example, z/OS 1.7, 1.8, 1.9 and 1.10.

DB2 supports n+1 releases. For example, DB2 for z/OS V8 can run in parallel with DB2 V9 in the same data sharing group as long as DB2 V9 is run in compatibility mode. The reason for this is that DB2 V8 employs Unicode as the encoding scheme for its catalog, while DB2 V7 stores the character data from the catalog in EBCDIC. Because SAP solutions running on DB2 V8 generally require the new-function mode of DB2 V8, the co-existence of V8 and V7 is very limited. It is allowed only during the migration phase from DB2 V7 to V8.

If both z/OS and DB2 need to be upgraded, the preferred sequence is to upgrade z/OS first, followed by DB2.

When z/OS Parallel Sysplex and DB2 data sharing are being used, the stopping and starting of z/OS and DB2 can be done without stopping the SAP system. This is accomplished by taking advantage of DB2 connection failover. The following steps should be used for each LPAR to be updated:

1. Build new DB2 loadlibs with the DB2 maintenance applied for each DB2 data sharing member.
A suggested name would be
`<db2 member name>.SDSNLOAD.NEW`
2. Stop all SAP batch processing on application servers connected to DB2 in this LPAR. Use SAP transaction RZ03 to choose and switch the application server to an operation mode that does not comprise any batch work process. Such an operation mode will prevent new batch work from getting scheduled on this

Change management

application server. In order to do this, you must have set up an operation mode without any batch work processes.

3. Activate DB2 connection failover by using SAP transaction 'DB2' to move each application server away from the LPAR that is going to be updated.
 - For *ABAP* instances, use SAP transactions ST04 or DBACOCKPIT to move the instance away from the LPAR that is going to be updated.
 - For *JAVA* instances, use the “graceful stop mechanism” to stop the DDF to move the instance away from the LPAR that is going to be updated. For details, refer to SAP Note **1085521**.
4. Stop the DB2 data sharing members in the LPAR.
Issue a DB2 Display Thread command to ensure that there are no active connections to this DB2 member before issuing the Stop DB2 command.
5. Switch from current DB2 loadlibs to new DB2 loadlibs.
This can be accomplished by renaming the loadlibs as follows:
RENAME D7X1.SDSNLOAD to D7X1.SDSNLOAD.OLD
RENAME D7X1.SDSNLOAD.NEW to D7X1.SDSNLOAD
6. At this point, z/OS can be stopped and re-IPLed to activate z/OS updates.
7. Restart the DB2 data sharing members in the LPAR.
8. Switch back to the normal configuration using SAP transaction DB2.
9. Restart all SAP batch processing on application servers connected to DB2 in this LPAR. Use “Opt Mode Switch” to add batch work processes.
10. Repeat step 1 through step 10 for each LPAR in the sysplex.

Part 6. Verification

Chapter 15. Verifying your implementation on z/OS

Verification procedures and failover scenarios	249
Overview of the test scenarios	249
Classification of the test scenarios	249
Test scenarios to verify the SA z/OS policy	250
Planned outage scenarios	250
Unplanned outage scenarios	250
Executed test scenarios	251
Planned outage scenarios	251
Unplanned outage scenarios	251
Test methodology	251
Purpose of the test	251
Expected behavior	251
Setup of the test environment	252
Verification of resource status	252
Preparation for the test (unplanned outage only)	254
Execution of the test	256
Verifications after the test	257
Analyzing problems	257
Planned outage test scenarios	258
Stop and start of the entire SAP RED system	258
Startup of all LPARs one after the other	259
Shutdown and restart of an LPAR	260
Unplanned outage test scenarios	264
Failure of the enqueue server	264
Failure of the message server	267
Failure of the NFS server	269
Failure of a TCP/IP stack	270
Failure of an LPAR	273
Problem determination methodology	276
SA z/OS problem determination	276
NetView netlog	276
z/OS syslog	277
Message Processing Facility	277
Problem determination in SA z/OS	278
SDF or NMC	278
DISPINFO	278
INGINFO	279
UNIX messages	280
If nothing happens	280
When you are really lost	280
Getting help from the Web	281
Where to check for application problems	281
Checking the network	282
Checking the configuration	282
Checking network devices	283
Dynamic VIPA	283
Routing tables and OSPF	283
Checking active connections	284
Checking the status of the Shared HFS and of NFS	284
Checking the status of DB2 and SAP connections	285
Check that DB2 is running	285

Check the SAP database connections	285
Availability test scenarios	286

Chapter 16. Verifying your implementation on Linux/AIX

Verification procedure and failover scenarios	289
Test setup	289
Scenarios	289
Testing an unplanned outage of an ABAP SCS	293
Testing an unplanned outage of a Java SCS	294

Chapter 15. Verifying your implementation on z/OS

This chapter contains these main topics:

- “Verification procedures and failover scenarios”
- “Problem determination methodology” on page 276
- “Availability test scenarios” on page 286

Verification procedures and failover scenarios

This chapter describes the test scenarios we designed and ran to test the SA z/OS policy.

Our z/OS test environment employed DB2 V9. With SAP NetWeaver '04 and DB2 V8. The ICLI client/server is no longer supported. Instead:

- DB2's DDF address space takes over the function of the ICLI server (the DDF address space is part of the standard z/OS DB2 automation policy).
- DB2 Connect takes over the function of the ICLI client on remote SAP application servers.

Overview of the test scenarios

Before defining and running test scenarios to verify the SA z/OS policy, we made the following assumptions:

- The z/OS and network configuration had been done.
- The high availability solution had been installed.
- The SA z/OS and NetView configuration had been done.
- The complete environment was available.

Classification of the test scenarios

The scenarios must cover both *planned outages* (or planned activities) and *unplanned outages* (or failures). And for each category, tests must be run at the *component* level (the component can be related to SAP, z/OS, or the network) and at the *LPAR* level.

The following table depicts, in the form of a matrix, some examples of test scenarios.

Table 21. 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

Test scenarios to verify the SA z/OS policy

We built a list of test scenarios, including planned and unplanned outages, to verify the SA z/OS policy.

Planned outage scenarios:

- Controlled operator intervention against SAP-related components:
 - Start and stop of all the SAP-related components
 - Start and stop of the entire SAP RED system
 - Start and stop of SCS
 - Move of SCS from one LPAR to the other
 - Start and stop of the enqueue replication server
 - Move of the enqueue replication server from one LPAR to another (if more than two LPARs)
 - Start and stop of the enqueue server
 - Start and stop of the message server
 - Start and stop of the NFS server
 - Move of the NFS server from one LPAR to the other
 - Start and stop of all DB2 subsystems belonging to the SAP system
 - Start and stop of a single DB2 subsystem
 - Start and stop of an SAP application server on z/OS
 - Start and stop of an SAP application server on Linux on System z
- Startup of the entire sysplex:
 - Startup of all LPARs one after the other
- Planned shutdown and restart of an LPAR containing SAP critical components:
 - Shutdown and restart of the LPAR where the enqueue server and the NFS server are running
 - Shutdown and restart of the LPAR where the enqueue replication server is running

Unplanned outage scenarios:

- Failure of an SAP component:
 - The enqueue server
 - The enqueue replication server
 - The message server
 - An SAP application server on z/OS
 - An SAP application server on Linux on System z
 - A DB2 subsystem
 - The NFS server
 - The syslog collector
 - A syslog sender
 - The SAP gateway
 - saprouter
 - saposcol
 - sapccmsr
- Failure of a network component:
 - A TCP/IP stack on z/OS
 - OSPF (OMPROUTE)

- A network adapter on System z
- 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

Executed test scenarios

The following scenarios were tested in our test environment:

Planned outage scenarios:

- Controlled operator intervention against SAP-related components:
 - Start and stop of the entire SAP RED system
- Startup of the entire sysplex:
 - Startup of all LPARs, one after the other
- Planned shutdown and restart of an LPAR containing critical SAP components:
 - Shutdown and restart of the LPAR where the enqueue server and the NFS server are running

Unplanned outage scenarios:

- Failure of a critical SAP component:
 - The enqueue server
 - The message server
- Failure of a critical network resource:
 - The NFS server
 - A TCP/IP stack
- Failure of an LPAR containing critical SAP components:
 - The LPAR where the enqueue server and NFS server are running

Test methodology

Although each scenario is different, many of the steps that need to be executed before, during, and after the test are similar. We describe these steps in the following section in the form of a methodology that we followed all through our tests, and which you can apply for any scenario you may want to test in your own environment.

Purpose of the test

We characterize the purpose of the test with two points:

- The *scope* of the test: Is the test run against a single component (for example, the enqueue server), a group of resources (for example, the whole SAP system), or an entire LPAR?
- The *action* to be tested: Do we want to test a normal startup or shutdown, a controlled movement, or do we want to simulate a failure?

Expected behavior

We describe the expected behavior of every component impacted during the test: Should it stop, restart in the same LPAR, move to the other LPAR, what should happen to the SAP application servers, what about transparency for the running workload?

Setup of the test environment

We prepare the test environment knowing which resources must be stopped, which must be up, and in which LPAR each component must be running.

Verification of resource status

Before each test, we used the following checklist to review the status of all the SAP-related resources defined in SA z/OS:

1. Do all the resources monitored by SA z/OS have a compound status SATISFACTORY?

Tip: The NetView command INGLIST SAP/APG displays the status of the application group SAP. If the compound status is SATISFACTORY, then we know that all resources belonging to that group have a compound state SATISFACTORY. Otherwise, we can drill down the tree of resources using option G (Members).

The following is a sample output of the NetView command INGLIST SAP/APG, showing the application group SAP with a compound status of SATISFACTORY:

```

INGKYST0          SA OS/390 - Command Dialogs      Line 1    of 1
Domain ID   = SC04A      ----- INGLIST -----      Date = 06/03/02
Operator ID = NETOP1          Sysplex = WTSCPLX1          Time = 16:04:34
CMD: A Update   B Start    C Stop      D INGRELS  E INGVOTE  F INGINFO
      G Members   H DISPTRG  I INGSCHED J INGGROUP      / scroll
CMD Name      Type System   Compound   Desired   Observed   Nature
-----
SAP           APG           SATISFACTORY AVAILABLE  AVAILABLE  BASIC
    
```

2. Are there any outstanding votes in SA z/OS?

Tip: The NetView command INGVOTE displays the list of all the votes in the system. The list should be empty.

The following is a sample output of the NetView command INGVOTE, showing that there are no outstanding votes:

```

INGKYRQ2          SA OS/390 - Command Dialogs      Line 1    of 5
Domain ID   = SC04A      ----- INGVOTE -----      Date = 06/03/02
Operator ID = NETOP1          Sysplex = WTSCPLX1          Time = 16:24:31
Cmd: C Cancel request  K Kill request  S Show details  V Show votes
Cmd Name      Type System   Request Data
-----
    
```

3. Are there any outstanding excludes in SA z/OS?

Note: There is no command to display all the excludes in SA z/OS at once. Individual INGINFO commands must be issued against every application group defined as SYSPLEX/MOVE groups.

In our configuration, we used the following commands:

```

INGINFO RED_EMPLX
INGINFO RED_ERSPLEX
INGINFO NFS_HAPLEX
INGINFO RED_RASPLX
INGINFO SAP_RTPLX
INGINFO RED_COPLEX
INGINFO RED_VPLX
    
```

The following shows a sample output of the NetView command INGINFO. We look more specifically at the section Group Details (on the third screen of the display). It shows that SC42 is in the exclude list of the application group RED_EMPLEX.

```

INGKYIN0          SA OS/390 - Command Dialogs          Line 43  of 189
Domain ID   = SC42A  ----- INGINFO -----          Date = 06/06/02
Operator ID = NETOP2          Sysplex = WTSCPLX1          Time = 11:03:14

Resource ==> RED_EMPLEX/APG          format: name/type/system
System ==>          System name, domain ID or sysplex name
Group Details...
Nature      : MOVE
Members     :
  RED_EMGRP/APG/SC04          Enqueue Group
    PREF = 700
    PREFADJ = 0
    SYSTEMS = SC04
  RED_EMGRP/APG/SC42          Enqueue Group
    PREF = 700
    PREFADJ = 0
    SYSTEMS = SC42
Policy      :
  PASSIVE = NO
  EXCLUDE = SC42

```

We usually do not want any excludes before the test. Therefore, this exclude should be removed by issuing the NetView command INGGROUP, as shown:
 INGGROUP RED_EMPLEX/APG ACTION=INCLUDE SYSTEMS=SC42

Tip: Instead of seven INGINFO commands, we used a special-purpose REXX procedure called SANCHK to display and remove all the outstanding excludes in SA z/OS. The code source for this procedure can be found in “SANCHK” on page 315. 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 NetView’s DSICLD data definition concatenation. Check the NetView startup procedure’s JCL DD statement for the DSICLD and add it to a data set (in our environment, the data set USER.CNMCLST, for example).

The following shows the output of the REXX procedure SANCHK. It shows that we have two outstanding excludes: SC42 is in the exclude list of the application groups RED_EMPLEX and NFS_HAPLEX.

```

* SC04A  SANCHK
| SC04A  Gathering data step 1 ...
| SC04A  Gathering data step 2 ...
| SC04A  Nothing to display ...
* SC04A  SANCHK
| SC04A  Gathering data step 1 ...
| SC04A  Gathering data step 2 ...
| SC04A

-----
Group      = NFS_HAPLEX/APG
Excluded   = SC42
Avoided    =

-----
Group      = RED_EMPLEX/APG
Excluded   = SC42
Avoided    =

-----
End of Sanity Check

```

Verification (z/OS)

We can also use the REXX procedure SANCHK with the option CLEAR to remove all the excludes:

```
* SC04A SANCHK CLEAR
| SC04A Gathering data step 1 ...
| SC04A Gathering data step 2 ...
| SC04A Processing CLEAR ...
| SC04A Processing CLEAR for NFS_HAPLEX/APG
U SC04A AOF099I FUNCTION SUCCESSFULLY COMPLETED
| SC04A Processing CLEAR for RED_EMPLX/APG
U SC04A AOF099I FUNCTION SUCCESSFULLY COMPLETED
| SC04A Finished CLEAR processing
```

4. Where are the enqueue server, message server, enqueue replication server and NFS server running before the test?

The following is a sample screen showing, on the left-hand side, the SAP-related components that are associated with each system. On the right-hand side, it shows the SAP-related components that can be moved from one system to the other.

In this example, the enqueue server (RED_ES), the NFS server (MVSNFSSA), and SAProuter (SAP_ROUTER) are running on SC04, and the enqueue replication server (RED_ERS) is running on SC42.

```

                S A P High Availability
Local Applications          Moving Applications
SC04          SC42          SC04          SC42
-----
RED_DB2MSTR  RED_DB2MSTR    MVSNFSSA    MVSNFSSA
RED_DB2DBM1  RED_DB2DBM1
RED_DB2IRLM  RED_DB2IRLM    SAP_RTVIPA  SAP_RTVIPA
RED_DB2DIST  RED_DB2DIST    SAP_ROUTER  SAP_ROUTER
              RED_VIPA    RED_VIPA
              RED_ES      RED_ES
              APPSRV06  APPSRV06
              APPSRV07  APPSRV07
              APPSRV08  APPSRV08
                                06/06/02 13:40
```

5. Are the NFS file systems accessible that are mounted on the remote SAP application server?

Tip: We either logon to the remote SAP application server and display the available file systems (using the UNIX command *df*), or we use the SAP transaction AL11 to check that we can access the files in the SAP directories.

Preparation for the test (unplanned outage only)

During the unplanned outage scenarios, we want to verify the impact of the failure for end users and for any workload that would be running on the system. Therefore, before each test, we execute the following preparation steps:

1. Log on to all the SAP application servers.
2. Create an SAP workload.

Note: To generate a workload you may use, for example, special-purpose batch jobs, or start a client copy.

We used a workload generated by a tool called ZAP1. The program goes through an insert/update/delete cycle several times. We set a sleep time between every step. During sleep time, the current work process is

released (to be available for other tasks). After sleep time, the program gets a work process again and continues with the next step. Our workload consisted of five of these programs running in parallel.

3. Generate entries in the enqueue table.

Tip: We use transaction SM12 to generate entries in the enqueue table. From the primary panel of transaction SM12, Select Lock Entries, enter *test* in the transaction field, as shown in the following panel:

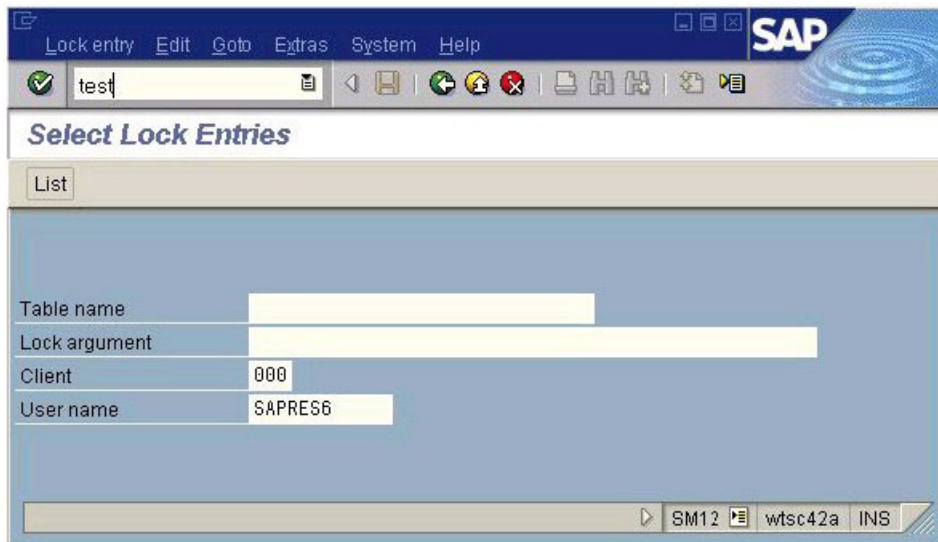


Figure 72. SM12 primary panel

A new selection appears in the menu bar: “Error handling”.

Click **Error handling** → **Test tools** → **Mass calls**

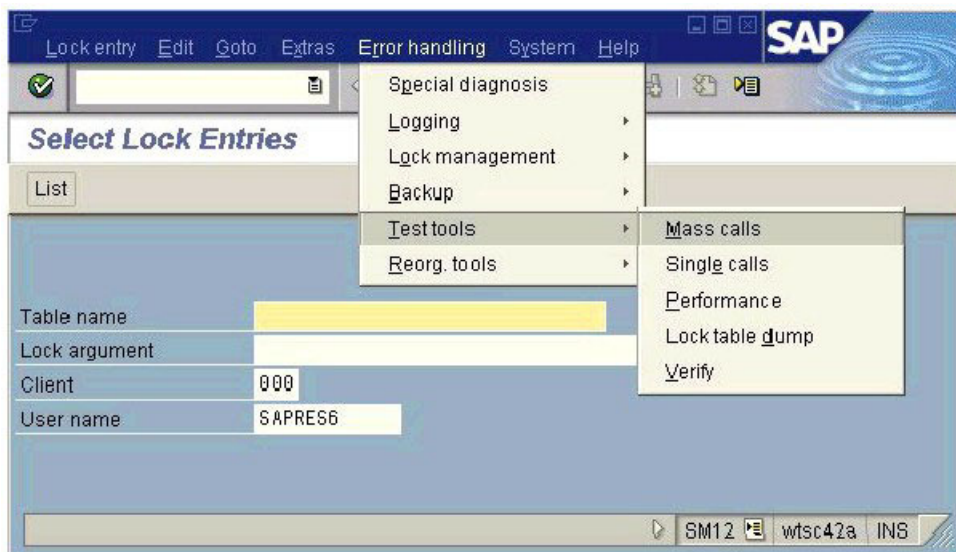


Figure 73. Error handling menu

Choose the number of lock entries you want to create (for our test purposes, we always used the default of 10 lock entries), then click Execute:

Verification (z/OS)

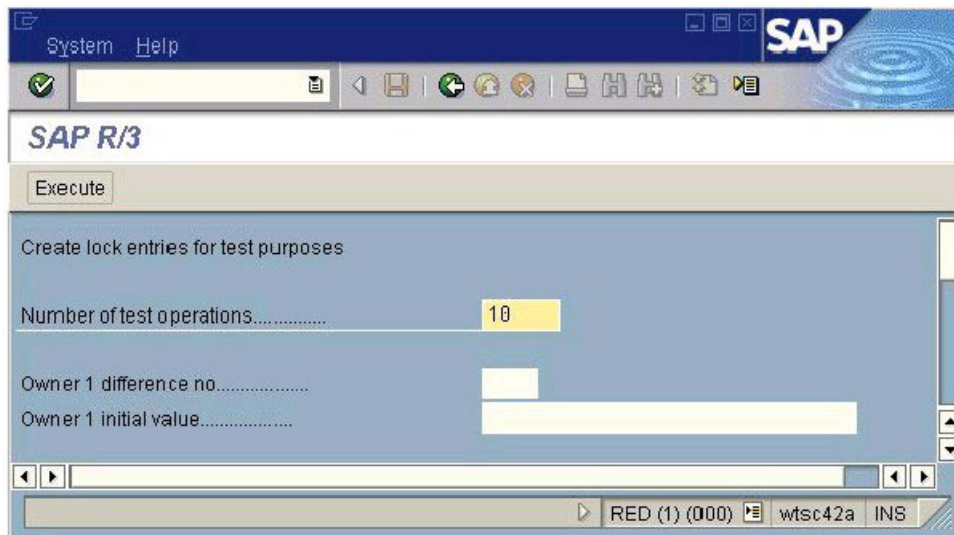


Figure 74. Enqueue test: start mass enqueue operations

The screen must stay open for the duration of the test. From *another* screen, we use SM12 to list the entries in the enqueue table:

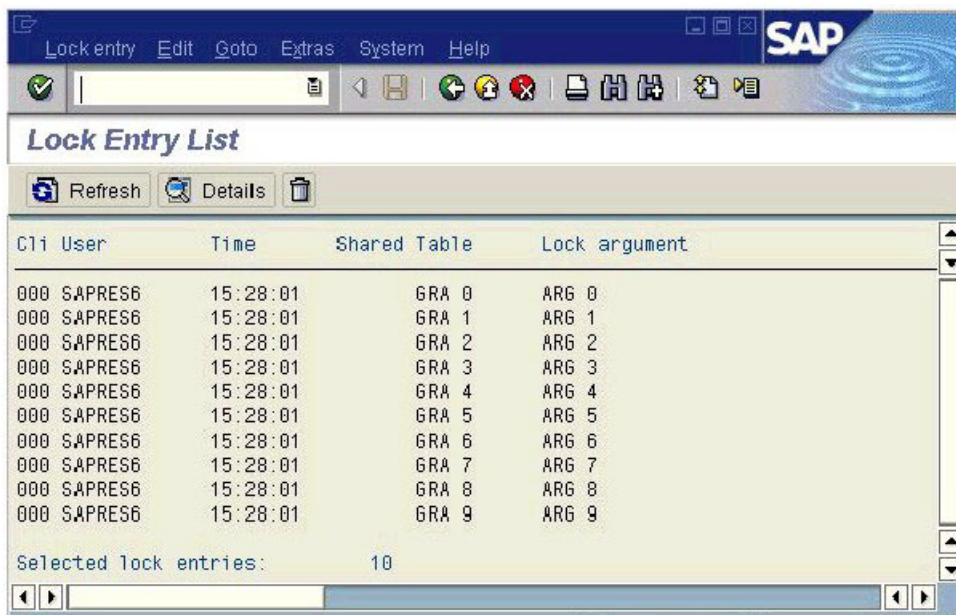


Figure 75. List of entries in the enqueue table

Execution of the test

The initiation of the test depends on the type of scenario.

- For a planned outage or a controlled move of resources, SA 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, we first reviewed the status of all the components using the same checklist as the one used before the test (see “Verification of resource status” on page 252).

Then, depending on the type of scenario (usually in the case of a failure), we did some additional verifications, such as:

- Looking at the SAP system log (SM21)
- Searching the SAP developer trace files for error messages
 - The file an error is recorded in may vary with the release of SAP. With SAP 4.6D, we exploit the following files:
 - dev_ms and dev_enqserv for errors regarding the message server and the enqueue server
 - dev_disp for errors regarding the connection to the message server
 - dev_w0 (for example) for errors regarding the connection to the enqueue server and the message server
- Displaying the status of internal and TCP/IP connections (SM59)
- Checking whether the workload we created is still running (SM66)
- Checking the number of lock entries in the enqueue table (SM12)
- Displaying the DB2 threads using the DB2 command -DIS THREAD(*)

Note: A trace file called enqueolog has been introduced to log the activity of the enqueue server and the status of the replication.

The following is an extract of the enqueue server log file. In our configuration, this file is located in directory:

/usr/sap/RED/ASCS00/work/enqueolog

```

Start: Thu May 30 11:34:57 2002: enqueue server started
RepAct: Thu May 30 11:41:22 2002: replication activated
RepDea: Thu May 30 14:15:20 2002: replication deactivated
Stop: Thu May 30 14:15:36 2002: enqueue server stopped: normal shutdown
Start: Thu May 30 14:16:20 2002: enqueue server started
RepAct: Thu May 30 14:21:39 2002: replication activated

```

Analyzing problems

If the results differ from the expected behavior, it is necessary to understand why. We put together some tips to help you with this complex troubleshooting phase in “Problem determination methodology” on page 276)

Planned outage test scenarios

This section describes the planned outage test scenarios we chose to perform in order to verify the SA z/OS policy.

For each scenario, we specified the following:

- purpose of the test
- expected behavior
- initial setup
- phases of the execution
- results we observed

In “Verification of resource status” on page 252, we describe the verification tasks that we performed before and after each test to check the status of the SAP-related components. In this section, we do not repeat these steps. However, the description of each test may contain additional verification tasks that are specific to the scenario.

Stop and start of the entire SAP RED system

In this scenario, we wanted to test the normal stop and restart of the entire SAP RED system (including application servers, enqueue servers, database servers, etc.) using SA z/OS. We split this scenario into two parts: first the stop of the SAP system, and then the restart.

The following table summarizes the execution of the stop phase.

Table 22. Stop of the entire SAP system with SA z/OS

Purpose	Scope: The entire SAP RED system Action: Planned stop using SA z/OS
Expected behavior	All RED-related resources should come down properly. The NFS server, SAProuter, and saposcol should stay up.
Setup	SC42 and SC04 must be up, including all required z/OS resources and SAP-related resources.
Execution	Issue a STOP request in SA z/OS against the application group RED_SAPPLEX.
Results	All RED-related resources came down properly. The NFS server, SAProuter, and saposcol stayed up.

Table 23 summarizes the execution of the start phase.

Table 23. Start of the entire SAP system with SA z/OS

Purpose	Scope: The entire SAP RED system Action: Planned start using SA z/OS
Expected behavior	All RED-related resources should come up properly.
Setup	SC42 and SC04 must be up, with all required z/OS resources, but all RED-related resources are stopped.
Execution	Kill the STOP request in SA z/OS against the application group RED_SAPPLEX.
Results	All RED-related resources came up properly.

To stop the entire SAP system, we issued a STOP request against the application group RED_SAPPLEX (option C):

```

INGKYST0          SA OS/390 - Command Dialogs          Line 1 of 1
Domain ID = SC04A ----- INGLIST ----- Date = 06/06/02
Operator ID = NETOP1          Sysplex = WTSCPLX1          Time = 19:21:01
CMD: A Update  B Start  C Stop  D INGRELS  E INGVOTE  F INGINFO
      G Members  H DISPTRG  I INGSCHED  J INGGROUP          / scroll
CMD Name      Type System  Compound  Desired  Observed  Nature
-----
C RED_SAPPLEX APG          SATISFACTORY  AVAILABLE  AVAILABLE  BASIC

```

We wanted a *normal* stop of the SAP RED system. Thus, we stayed with the default type NORM.

Note: Because of our SA z/OS definitions, only the monitor for the remote SAP application server running on Linux has stopped. The SAP application server itself stayed idle until the system was up again, and then it reconnected.

If we wanted to stop the remote SAP application server, we needed to issue a STOP request with the option FORCE on the application group RED_RASPLEX before stopping the group RED_SAPPLEX.

The status of the RED-related resources is that all the resources went from an UP status to a STOPPING status, and then finally to an AUTODOWN status. The NFS server, saposcol, and SAProuter are still running.

Note: The SA z/OS resource APPSRV06 appears with an AUTODOWN status although the remote SAP application server is still running on Linux. Only the monitor has stopped.

To restart the SAP system, we had to kill the remaining MakeUnavailable vote on the application group RED_SAPPLEX:

```

INGKYRQ0          SA OS/390 - Command Dialogs          Line
Domain ID = SC04A ----- INGVOTE ----- Date = 06/06/02
Operator ID = NETOP1          Sysplex = WTSCPLX1          Time = 19:25:57
Resource ==> RED_SAPPLEX/APG
System ==>          System name, domain id or sysplex name
Cmd: C cancel request  K Kill request  S show request details
Cmd Action WIN Request/Vote Data
-----
K STOP  Y Request   : MakeUnavailable
        Created  : 2002-06-06 19:21:23
        Originator : OPER_NETOP1(NETOP1)
        Priority   : 01720000    Should Be Down - Operator
        Status    : Winning/Satisfied

```

After some time, all SAP-related resources are up and running again. The local applications are in UP status and the enqueue server is running on SC04, whereas the enqueue replication server is running on SC42.

Startup of all LPARs one after the other

In this scenario, we wanted to test the normal startup of the LPARs, one after the other. We split this scenario into two parts: the startup of the first LPAR (in our case SC42), and then the startup of the second LPAR (in our case SC04).

Table 24 on page 260 summarizes the startup of the first LPAR.

Table 24. Startup of the first LPAR

Purpose	Scope: One LPAR Action: Planned startup of an LPAR while the other one is down
Expected behavior	The LPAR should come up with all required address spaces including all SAP-related resources: database server, sapccmsr, and saposcol, plus NFS server and enqueue server, but not enqueue replication server.
Setup	Both LPARs must be down. An HMC is required.
Execution	IPL SC42
Results	SC42 came up with all required address spaces including all SAP-related resources: database server, sapccmsr, and saposcol, plus NFS server and enqueue server, but not enqueue replication server.

Table 25 summarizes the startup of the second LPAR.

Table 25. Startup of the second LPAR

Purpose	Scope: One LPAR Action: Planned startup of an LPAR while the other one is up
Expected behavior	The LPAR should come up with all required address spaces including all SAP-related resources: database server, sapccmsr, and saposcol, plus enqueue replication server.
Setup	The first LPAR must be up with all required z/OS resources and SAP-related resources: database server, sapccmsr, and saposcol, plus NFS server and enqueue server. The second LPAR must be down. An HMC is required.
Execution	IPL SC04
Results	SC04 came up with all required address spaces including all SAP-related resources: database server, sapccmsr, and saposcol, plus enqueue replication server.

Shutdown and restart of an LPAR

In this scenario, we wanted to test the shutdown and restart of the LPAR where the enqueue server and the NFS server are running. We split this scenario into two parts: first the shutdown, and then the restart of the LPAR.

Table 26 summarizes the execution of the shutdown phase.

Table 26. Shutdown of the LPAR where the ES and NFS servers are running

Purpose	Scope: One LPAR Action: Planned shutdown of the LPAR where the enqueue server and the NFS server are running
---------	---

Table 26. Shutdown of the LPAR where the ES and NFS servers are running (continued)

Expected behavior	<p>The NFS server should move to the other LPAR.</p> <p>The enqueue server should move to the other LPAR.</p> <p>The enqueue replication server should stop or move to another LPAR if more than two LPARs are available.</p> <p>The SAP application server on the remaining LPAR should reconnect to the message server and enqueue server.</p> <p>The LPAR should come down properly to the point where we can enter the following command to remove the LPAR from the sysplex:</p> <pre>/V XCF,<sysname>,OFFLINE</pre>
Setup	<p>SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with:</p> <ul style="list-style-type: none"> • The enqueue server running on SC04. • The enqueue replication server running on SC42. • The NFS server running on SC04.
Execution	<p>Move the SAP critical components running on SC04 to SC42 (NFS server, enqueue server, and SAProuter).</p> <p>Stop the remaining SAP-related resources on SC04 (application server, sapccmsr, saposcol, and database server).</p> <p>Issue a STOP request in SA z/OS against the system group SC04 using the NetView command SHUTSYS ALL.</p>
Results	<p>The NFS server moved from SC04 to SC42.</p> <p>The enqueue server moved from SC04 to SC42.</p> <p>The enqueue replication server stopped (the application group RED_ERSPLEX has a status INHIBITED).</p> <p>The SAP application server APPSRV10 on SC42 reconnected to the message server and enqueue server.</p> <p>SC04 came down properly to the point where we can enter:</p> <pre>/V XCF,SC04,OFFLINE</pre>

Table 27 summarizes the execution of the restart phase.

Table 27. Restart of the LPAR where the ES and NFS servers are running

Purpose	<p>Scope: One LPAR</p> <p>Action: Restart after planned shutdown of the LPAR where the enqueue server and the NFS server are running (in our case SC04)</p>
Expected behavior	<p>SC04 should come up with all required address spaces including database server, SAP application server, sapccmsr, and saposcol.</p> <p>The enqueue server and the NFS server should stay on SC42.</p> <p>The enqueue replication server should restart to SC04.</p>

Verification (z/OS)

Table 27. Restart of the LPAR where the ES and NFS servers are running (continued)

Setup	SC42 must be up, including all required z/OS resources and SAP-related resources: database server, SAP application server, sapccmsr, and saposcol, plus NFS server and enqueue server. SC04 must be down and an HMC is required.
Execution	IPL SC04
Verifications	The enqueue replication server reconnects to the enqueue server.
Results	SC04 came up with all required address spaces including database server, SAP application server, sapccmsr, and saposcol. The enqueue server and the NFS server stayed on SC42. The enqueue replication server was restarted on SC04.

All the SAP-related resources are in UP status. The NFS server and the enqueue server are running on SC04. The enqueue replication server is running on SC42.

First, we moved the NFS server, the enqueue server, and the SAProuter from SC04 to SC42. We used the NetView command INGGROUP to exclude the system SC04 for the associated SA z/OS resources:

```

INGKYGRA          SA OS/390 - Command Dialogs
Domain ID = SC04A  ----- INGGROUP -----      Date = 06/07/02
Operator ID = NETOP1      Sysplex = WTSCPLX1      Time = 19:38:32
Specify or revise the following data:
System =>          System name, domain id or sysplex name
Action => EXCLUDE  EXCLUDE-AVOID-INCLUDE or ACTIVATE-PACIFY or RESET
Group(s) => NFS_HAPLEX/APG RED_EMPLX/APG RED_ERSPLEX/APG
              RED_RASPLEX/APG SAP_RTPLEX/APG
System(s)=> SC04
  
```

Note that we also excluded SC04 for the resource RED_ERSPLEX. If we had had a third system, the enqueue replication server would have moved to that system. In our configuration, the enqueue replication server stopped and the application group RED_ERSPLEX remained in an INHIBITED status:

```

Domain ID = SC04A  ----- INGLIST -----      Date = 06/07/02
Operator ID = NETOP1      Sysplex = WTSCPLX1      Time = 19:43:16
CMD: A Update  B Start  C Stop  D INGRELS  E INGVOTE  F INGINFO
      G Members  H DISPTRG  I INGSCHED  J INGGROUP  / scroll
CMD Name      Type System  Compound  Desired  Observed  Nature
-----
RED_ERSPLEX  APG          INHIBITED  AVAILABLE  SOFTDOWN  MOVE
  
```

Then we stopped the SAP-related resources that were still running on SC04: the SAP application server APPSRV11, saposcol, sapccmsr, and the DB2 subsystem.

Because of all the dependencies defined in the SA z/OS policy, issuing a STOP request against the application group RED_DB2GRP on SC04 not only stops the DB2 subsystem, but if the parameter scope is set to ALL (default value), it also stops all the children: the SAP application server APPSRV11, and sapccmsr. SA z/OS lists all the resources affected by the STOP request and asks for confirmation; see the next panel:

```

AOFKVFY1          SA OS/390 - Command Dialogs      Line 1   of 8
Domain ID = SC04A ----- INGREQ -----         Date = 06/07/02
Operator ID = NETOP1                               Time = 19:41:07
Verify list of affected resources for request STOP
CMD: S show overrides  T show trigger details  V show votes
Cmd Name          Type System  TRG SVP  W Action Type  Observed Stat
-----
APPSRV11         APL  SC04                Y STOP  NORM  AVAILABLE
RED_DB2GRP       APG  SC04                Y STOP  NORM  AVAILABLE
RED_RFC          APL  SC04                Y STOP  NORM  AVAILABLE
RED_RASGRP       APG  SC04                Y              SOFTDOWN

```

Then we issued a STOP request against the application group SAP_RTGRP on SC04. This stopped the SAProuter, and there were no longer any SAP-related resources active on SC04.

We were now able to take the system down using the NetView command SHUTSYS ALL:

```

INGKYRU0          SA OS/390 - Command Dialogs      Page 1 of 2
Domain ID = SC04A ----- INGREQ -----         Date = 06/07/02
Operator ID = NETOP1                               Time = 19:45:08
Resource => SC04/SYG/SC04                          format: name/type/system
System =>                               System name, domain ID or sysplex name
Request => STOP                                     Request type (START, UP or STOP, DOWN)
Type => NORM                                         Type of processing (NORM/IMMED/FORCE/user) or ?
Scope => ALL                                         Request scope (ONLY/CHILDREN/ALL)
Priority => LOW                                       Priority of request (FORCE/HIGH/LOW)
Expire => ,                                          Expiration date(yyyy-mm-dd), time(hh:mm)
Timeout => 0 / MSG                                   Interval in minutes / Option (MSG/CANCEL)
AutoRemove =>                                       Remove when (SYSGONE, UNKNOWN)
Restart => NO                                         Restart resource after shutdown (YES/NO)
Override => NO                                       (ALL/NO/TRG/FLG/DPY/STS/UOW/INIT)
Verify => YES                                         Check affected resources (YES/NO/WTOR)
Precheck => YES                                       Precheck for flags and passes (YES/NO)
Appl Parm =>

```

SC04 came down to the point where we could enter the following MVS command to remove SC04 from the sysplex:

```
/V XCF,SC04,OFFLINE
```

We checked that the SAP application server APPSRV10 on SC42 reconnected successfully to the message and enqueue servers by examining the developer trace file of work process 0 (dev_w0).

The second part of the test can now be performed: the restart of the LPAR SC04.

We re-IPLed the LPAR. SA z/OS was started automatically and restarted all the resources on the system, including the DB2 subsystem, the SAP application server APPSRV11, sapccmsr, and saposcol.

The enqueue replication server was not restarted because we still had the exclude of SC04 on the application group RED_ERSPLEX. To restart it, we removed this exclude (and all the outstanding excludes) using the NetView command INGGROUP:

Verification (z/OS)

```
INGKYGRA          SA OS/390 - Command Dialogs
Domain ID = SC04A ----- INGGROUP ----- Date = 06/07/02
Operator ID = NETOP1          Sysplex = WTSCPLX1      Time = 20:06:16
Specify or revise the following data:
System =>          System name, domain id or sysplex name
Action => INCLUDE  EXCLUDE-AVOID-INCLUDE or ACTIVATE-PACIFY or RESET
Group(s) => NFS_HAPLEX/APG RED_EMPLX/APG RED_ERSPLEX/APG
              RED_RASPLEX/APG SAP_RTPLEX/APG
System(s)=> SC04
```

As described in “Verification of resource status” on page 252, we could also have used our special-purpose REXX procedure SANCHK to remove the outstanding excludes.

The enqueue replication server started immediately on SC04.

Because we did not set any preferences in the policy to favor one LPAR or the other, the enqueue server and the NFS server stayed in place, on SC42.

We looked at the enqueue server log file

/usr/sap/RED/ASCS00/work/enqueolog

to verify that the enqueue replication server reconnected to the enqueue server and that the replication was active. Following is the extract of this file corresponding to the time interval of our test.

```
RepDea: Fri Jun 7 19:38:40 2002: replication deactivated
Stop: Fri Jun 7 19:38:43 2002: enqueue server stopped: normal shutdown
Start: Fri Jun 7 19:38:58 2002: enqueue server started
RepAct: Fri Jun 7 20:06:26 2002: replication activated
```

Unplanned outage test scenarios

This section describes the unplanned outage test scenarios we chose to perform in order to verify the SA z/OS policy.

For each scenario, we specified the following:

- Purpose of the test
- Expected behavior
- Initial setup
- Preparation for the test
- Phases of the execution
- Results we observed

In “Verification of resource status” on page 252, we describe the verification tasks that we performed before and after each test to check the status of the SAP-related components. In this section, we do not repeat these steps. However, the description of each test may contain additional verification tasks that are specific to the scenario.

Failure of the enqueue server

In this scenario, we wanted to simulate the failure of the enqueue server and test the behavior of SA z/OS. We also wanted to measure the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 28. Failure of the enqueue server

Purpose	Scope: Enqueue server Action: Unplanned outage
Expected behavior	SA z/OS should show a PROBLEM/HARDDOWN status for the resource RED_ES and restart SCS (that is, all the members of the application group RED_EMGRP) on the LPAR where the enqueue replication server is running. The enqueue replication server should stop or move to another LPAR if more that two LPARs are available. The failure should be transparent to the SAP workload.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • The enqueue server running on SC42. • The enqueue replication server running on SC04. • The NFS server running on SC42.
Preparation	Log on to all SAP application servers. Create a workload on one SAP application server (APPSRV11 on SC04). Create entries in the enqueue table.
Execution	Use the UNIX command <i>kill -9</i> to kill the enqueue server process externally (from SA z/OS).
Verifications	Check that the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Look for error messages in the enqueue log file , in the dev_enqserv file, in the developer traces dev_disp and dev_wx, and in the system log (SM21).
Results	SA z/OS showed a PROBLEM/HARDDOWN status for RED_ES on SC42 and restarted SCS (that is, all the members of the application group RED_EMGRP) on SC04. The enqueue replication server stopped. The failure was transparent to the SAP workload.

Before the test, all SAP-related resources are in UP status. The NFS and enqueue servers are running on SC42, and the enqueue replication server is running on SC04.

As described in “Preparation for the test (unplanned outage only)” on page 254, we logged on to all SAP application servers, created a workload on APPSRV11 (five parallel tasks), and generated 10 lock entries in the enqueue table.

Then we simulated the failure: we killed the enqueue server process from SA z/OS, using the UNIX command *kill -9 <pid>*:

Verification (z/OS)

```
SC42>ps -ef | grep EM
redadm      852529   17632351   - 15:23:30 ?           0:00 se.sapRED_ASCS00 -F pf=/
usr/sap/RED/SYS/profile/RED_ASCS00
DFS         852860   17629600   - 16:10:01 tty00002  0:00 grep EM
redadm      853628   34408072   - 15:23:33 ?           0:00 co.sapRED_ASCS00 -F pf=/
usr/sap/RED/SYS/profile/RED_ASCS00
redadm      853637   34408062   - 15:23:33 ?           0:06 es.sapRED_ASCS00 pf=/usr
/sap/RED/SYS/profile/RED_ASCS00
redadm      855155   51186817   - 15:23:29 ?           0:00 gw.sapRED_ASCS00 pf=/usr
/sap/RED/SYS/profile/RED_ASCS00
redadm      855172   34408031   - 15:23:30 ?           0:00 ms.sapRED_ASCS00 pf=/usr
/sap/RED/SYS/profile/RED_ASCS00
SC42> kill -9 853637
```

After the failure the resource RED_ES on SC42 has the status PROBLEM/HARDDOWN.

All resources of SCS (all members of the RED_EMGRP) are in UP status on SC04 after the failover. The NFS server is still running on SC42. The enqueue replication server has stopped.

Using transaction SM66, we verified that the five parallel tasks of our workload were still running after the failure.

When the enqueue server restarts on SC04, it reads the enqueue replication table from shared memory and rebuilds the enqueue table. Using the transaction SM12, we verified that the 10 lock entries we had generated were still in the enqueue table.

Looking at the enqueue server log file (enqueolog), we verified that the enqueue server restarted and the enqueue replication server was not running (there was no message specifying that replication was active).

Looking at the developer trace file dev_disp, we were able to verify that the dispatcher lost its connection with the message server and reconnected later on.

We also looked at the developer trace file of one of the work processes running our workload, for example dev_w2. We could see that the work process lost its connection with the enqueue server and reconnected just after the enqueue server restarted.

The following log output shows the messages written in the SAP system log (SM21) during the interval of the test.

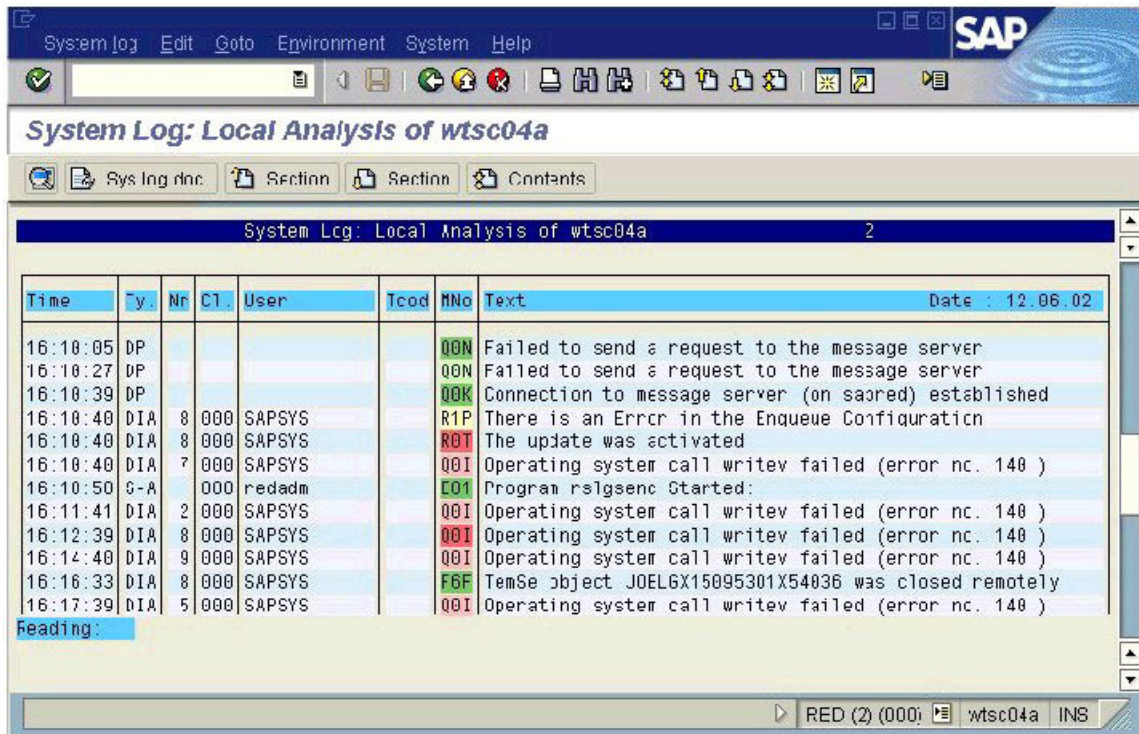


Figure 76. SAP system log (SM21)

Note that the system log shows a 6-minute interval before complete reconnection of the SAP application server. This was due to a bug in TCP/IP (probably related to our multiple-stack environment). After we changed the VIPARANGE statement on SC42 to NONDISRUPTIVE mode in the TCP/IP configuration, the recovery time was reduced to less than a minute.

Because we had only two systems, the enqueue replication server is stopped and the application group RED_ERSPLEX remains in an INHIBITED status.

If we had used a third system, SA z/OS would have restarted the enqueue replication server on that system.

We used the NetView command SETSTATE to tell SA z/OS that the resource RED_ES on SC42 should be in the AUTODOWN state (because we knew the source of the failure and did not need to investigate it).

As a result of this command, the resource RED_ES on SC24 is set to the status AUTODOWN, and the enqueue replication server immediately restarts on SC42.

Failure of the message server

In this scenario we wanted to simulate the failure of the message server and test the behavior of SA z/OS. We also wanted to measure the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 29. Failure of the message server

Purpose	Scope: Message server Action: Unplanned outage
---------	---

Table 29. Failure of the message server (continued)

Expected behavior	SA z/OS should try to restart the message server in place until the critical threshold is reached (5 failures in 10 minutes). If the critical threshold is reached, SA z/OS should show a PROBLEM/HARDDOWN status for the resource RED_MS and the entire SCS will move to the other system. The failure should be transparent to the SAP workload.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • The enqueue server running on SC42. • The enqueue replication server running on SC04. • The NFS server running on SC42.
Preparation	Log on to all SAP application servers. Create a workload on one SAP application server (APPSRV11 on SC04). Create entries in the enqueue table.
Execution	Use the UNIX command <i>kill -9</i> to kill the message server process externally (out of SA z/OS).
Verifications	Check that the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Look for error messages in the developer trace dev_disp and in the system log (SM21).
Results	SA z/OS restarted the message server in place, on SC42. The failure was transparent to the SAP workload.

Before the test, all SAP-related resources are in UP status. The NFS and enqueue servers are running on SC42, and the enqueue replication server is running on SC04.

As described in “Preparation for the test (unplanned outage only)” on page 254, we logged on to all SAP application servers, created a workload on APPSRV11 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

Then we simulated the failure: we killed the message server process out of SA z/OS, using the UNIX command *kill -9 <pid>*:

```
SC42>ps -ef | grep ms.sapRED_ASCS00
redadm 34408866 854437 - 09:47:44 ? 0:00 ms.sapRED_ASCS00 pf=/usr
/sap/RED/SYS/profile/RED_ASCS00
DFS 854747 51186380 - 10:54:55 tty0003 0:00 grep ms.sapRED_ASCS00
SC42>kill -9 34408866
```

Because the critical threshold was not reached, SA z/OS immediately restarted the message server in place, on SC42.

The failure was transparent: the workload was still running (SM66), and the lock entries that we generated were still in the enqueue table (SM12).

Looking at the trace file of the dispatcher (dev_disp), we verified that it lost its connection with the message server and reconnected a few seconds later.

The following shows the messages written in the SAP system log (SM21) during the interval of the test.

Time	Ty.	Nr	Cl	User	Tcod	MNo	Text	Date
10:55:02	DP					106	Request (type DIA) cannot be processed	13.06.02
10:55:02	DIA	6	000	SAPSYS		30Z	The update dispatch info was reset	
10:55:02	DP					10N	Failed to send a request to the message server	
10:55:02	DIA	6	000	SAPSYS		30R	The connection was de-activated after a DB error	
10:55:09	DP					10K	Connector to message server (on sapred) established	
10:55:09	DIA	5	000	SAPSYS		31P	There is an Error in the Enqueue Configuration	
10:55:09	DIA	5	000	SAPSYS		30T	The update was activated	

Figure 77. SAP system log (SM21)

Failure of the NFS server

In this scenario, we wanted to simulate the failure of the NFS server and test the behavior of SA z/OS. We also wanted to measure the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 30. Failure of the NFS server

Purpose	Scope: NFS server Action: Unplanned outage
Expected behavior	SA z/OS should restart the NFS server. Existing NFS mounts should be reestablished. The failure should be transparent to the SAP workload.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • The enqueue server running on SC42. • The enqueue replication server running on SC04. • The NFS server running on SC42.
Preparation	Log on to all SAP application servers. Create a workload on a remote SAP application server (APPSRV06). Create entries in the enqueue table.
Execution	Cancel the address space MVSNFSSA on SC42.

Verification (z/OS)

Table 30. Failure of the NFS server (continued)

Verifications	<p>Check that the workload is still running (SM66).</p> <p>Verify the number of entries in the enqueue table (SM12).</p> <p>Check that the file systems are accessible (AL11).</p> <p>Look for error messages in the system log (SM21).</p>
Results	<p>SA z/OS restarted the NFS server.</p> <p>Existing NFS mounts were reestablished.</p> <p>The failure was transparent to the SAP workload.</p>

Before the test, all SAP-related resources are in UP status. The NFS and enqueue servers are running on SC42, and the enqueue replication server is running on SC04.

As described in “Preparation for the test (unplanned outage only)” on page 254, we logged on to all SAP application servers, created a workload on the remote SAP application server APPSRV06 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

Then we simulated the failure by cancelling the address space of the NFS server on SC42 using the following command:

```
/C MVSNFSSA
```

Because, at the time of the test, the effective preference of SC04 was higher than that of SC42, SA z/OS immediately restarted the NFS server on SC04 (along with its VIPA) and put the resource MVSNFSSA on SC42 in a RESTART status:

AOFKSTA5		SA OS/390 - Command Dialogs				Line 1 of 2	
Domain ID = SC04A		----- DISPSTAT -----				Date = 06/13/02	
Operator ID = NETOP1						Time = 11:30:12	
A ingauto B setstate C ingreq-stop D thresholds E explain F info G tree							
H trigger I service J all children K children L all parents M parents							
CMD	RESOURCE	STATUS	SYSTEM	JOB NAME	A I S R D RS	TYPE	Activity
-----	-----	-----	-----	-----	-----	-----	-----
	MVSNFSSA	UP	SC04	MVSNFSSA	Y Y Y Y Y Y	MVS	--none--
	MVSNFSSA	RESTART	SC42	MVSNFSSA	Y Y Y Y Y Y	MVS	--none--

The failure is transparent: the workload is still running (SM66) and the lock entries that we generated are still in the enqueue table (SM12). All the file systems that are NFS-mounted on VMLINUX6 are accessible (AL11). No error messages are written to the SAP system log (SM21).

Failure of a TCP/IP stack

In this scenario, we wanted to simulate the failure of the TCP/IP stack on the system where the enqueue server and the NFS server are running, and test the behavior of SA z/OS. We also wanted to measure the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 31. Failure of a TCP/IP stack

Purpose	Scope: TCP/IP stack Action: Unplanned outage
Expected behavior	SA z/OS should try to restart the TCP/IP stack until the critical threshold is reached. If the critical threshold is reached, SA z/OS should show a PROBLEM/HARDDOWN status and the TCP/IP stack will not be restarted. The NFS server should fail and SA z/OS should restart it. SCS should fail and SA z/OS should restart it on the LPAR where the enqueue replication server is running. SA z/OS should try to restart the enqueue replication server on a different LPAR. The SAP application server running on the LPAR where the failure occurs should fail and SA z/OS should restart it. 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 be transparent.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • The enqueue server running on SC42. • The enqueue replication server running on SC04. • The NFS server running on SC42.
Preparation	Log on to all SAP application servers. Create a workload on APPSRV11 running on SC04 and on APPSRV06 running on VMLINUX6 and connected to SC42. Create entries in the enqueue table.
Execution	Cancel the address space TCPIPA on SC42.
Verifications	Check if the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Look for error messages in the enqueue log file, in the developer traces dev_disp and dev_wx, and in the system log (SM21).

Verification (z/OS)

Table 31. Failure of a TCP/IP stack (continued)

Results	<p>SA z/OS could restart the TCP/IP stack on SC42.</p> <p>The NFS server failed and SA z/OS restarted it on SC04.</p> <p>SCS failed and SA z/OS restarted it on SC04.</p> <p>SA z/OS tried to restart the enqueue replication server on SC42 but failed because the resource RED_ES on SC42 was in a STUCK status because its USS process was hanging. After we manually cancelled the process, the enqueue replication server was able to start on SC42.</p> <p>APPSRV10 running on SC42 failed. SA z/OS restarted it.</p> <p>For APPSRV06 running on VMLINUX6 and connected to the database server on SC42, running transactions were rolled back and, because the TCP/IP stack was restarted before failover time-out detection, work processes could reconnect to the database server on SC42.</p> <p>For APPSRV11 running on SC04, the failure was transparent.</p>
---------	--

Before the test, all SAP-related resources are in UP status. The NFS and enqueue servers are running on SC42, and the enqueue replication server is running on SC04.

As described in “Preparation for the test (unplanned outage only)” on page 254, we logged on to all SAP application servers, created a workload on APPSRV11 (5 parallel tasks) as well as on APPSRV06 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

We simulated the failure by stopping TCPIPA on SC42 using the following command:

```
/P TCPIPA
```

Because the critical threshold was not reached, SA z/OS immediately restarted TCPIPA on SC42:

```
10:20:26.37 SAPRES6 00000290 P TCPIPA
...
10:20:31.37 STC11046 00000090 $HASP395 TCPIPA ENDED
...
10:20:32.15 AWRK0942 00000290 S TCPIPA
...
10:20:32.76 STC11974 00000090 $HASP373 TCPIPA STARTED
```

The failure of the TCP/IP stack led to the failure of the NFS server, SCS, the SAProuter, and the SAP application server APPSRV10 running on SC42.

SA z/OS immediately restarted the NFS server on SC04.

SA z/OS restarted SCS on the LPAR where the enqueue replication server was running, that is, SC04. The enqueue replication server stopped and SA z/OS tried to restart it on SC42.

During our test, although SA z/OS had successfully restarted SCS on SC04, the resource RED_ES on SC42 remained in a STUCK status—the USS process was hanging and we had to cancel it using the following command:

```
/C REDADMES,A=3FE
```

As soon as the process was cancelled, the enqueue replication server started on SC42 and the replication was activated:

```
RepDea: Mon Jun 17 10:20:27 2002: replication deactivated
Start: Mon Jun 17 10:21:37 2002: enqueue server started
RepAct: Mon Jun 17 10:33:12 2002: replication activated
```

We believe that this problem has to do with the fact that we were running with multiple TCP/IP stacks. Instead of recovering manually, we could have added the CANCEL command in the SA z/OS policy, as last shutdown command for the resource RED_ES.

SA z/OS immediately restarted the SAProuter on SC04.

The SAP application server APPSRV10 running on SC42 went down and was immediately restarted by SA z/OS. All the sessions connected to this SAP application server were, of course, lost and needed to be restarted.

The SAP application server APPSRV06 running on VMLINUX6 lost the connection to the database server on SC42. The five running transactions received a DB2 SQL error 0 and were rolled back. The work processes were put in a reconnect status. The running sessions were lost and needed to be restarted by the users. Within seconds, the work processes reestablished the connection and left the reconnect status.

The transaction DB2 showed that the current DB host was still wtsc42a. We used the DB2 command -DIS THREAD(*) to check that all the threads are connected to SC42. Connection information for each work process can be found in the developer trace files dev_wx.

Note: During our test, we observed that the work processes could reconnect to the primary database server. This was because the TCP/IP stack was restarted before failover time-out detection. However, especially in the case of a heavy workload, you could experience a failover to the standby database server.

For the SAP application server APPSRV11 running on SC04, the failure is transparent—the workload is still running (SM66) and the lock entries that we generated are still in the enqueue table (SM12). The developer trace dev_disp shows that the dispatcher lost its connection with the message server and reconnected later on.

The developer trace dev_w0 shows that the work process lost its connection with the enqueue server and reconnected later on as soon as the enqueue server was available.

All the SAP-related resources are in UP status after the failover. The NFS and enqueue servers are running on SC04. The enqueue replication server is running on SC42.

Failure of an LPAR

In this scenario, we wanted to simulate the failure of the LPAR where the enqueue server and the NFS server were running and test the behavior of SA z/OS. We also wanted to measure the impact of the failure on the SAP workload.

Verification (z/OS)

The following table summarizes the execution of the test.

Table 32. Failure of the LPAR where the ES and NFS servers are running

Purpose	Scope: One LPAR Action: Unplanned outage
Expected behavior	ARM should restart the failing DB2 subsystem on another LPAR with the option RESTART(LIGHT). The DB2 subsystem will go down after successful startup. SA 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.
Expected behavior (continued)	For the remote SAP application server connected to the database server running on the failing LPAR, running transactions should be rolled back and work processes should failover to the standby database server. For the SAP application server running on the other LPAR, the failure should be transparent.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • The enqueue server running on SC42. • The enqueue replication server running on SC04. • The NFS server running on SC42.
Preparation	Log on to all SAP application servers. Create a workload on APPSRV11 running on SC04 and on APPSRV06 running on VMLINUX6 and connected to the database server on SC42. Create entries in the enqueue table.
Execution	System reset at the HMC for SC42.
Verifications	Check if the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Look for error messages in the enqueue log file, in the developer traces dev_disp and dev_wx, and in the system log (SM21).
Results	ARM restarted the failing DB2 subsystem D7X1 on SC04 with the option RESTART(LIGHT). The DB2 subsystem went down after successful startup. SA z/OS restarted the NFS server on SC04. SA z/OS restarted SCS on SC04. The enqueue replication server stopped. For APPSRV06 running on VMLINUX6 and connected to the database server on SC42, running transactions were rolled back and work processes reconnected to the standby database server on SC04. For APPSRV11 running on SC04, the failure was transparent.

Before the test, all SAP-related resources are in UP status. The NFS and enqueue servers are running on SC42, and the enqueue replication server is running on SC04.

As described in “Preparation for the test (unplanned outage only)” on page 254, we logged on to all SAP application servers, created a workload on APPSRV11 (5 parallel tasks) as well as on APPSRV06 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

We simulated the failure by doing a system reset at the HMC.

We used the NetView command INGLIST */*/SC42 to display the status of the resources on SC42. They all appeared with a status INHIBITED/SYSGONE. The following panel shows, as an example, the status of the application group RED_DB2GRP.

INGKYST0 SA OS/390 - Command Dialogs Line 1 of 8						
Domain ID = SC04A		----- INGLIST -----			Date = 06/18/02	
Operator ID = NETOP1		Sysplex = WTSCPLX1			Time = 14:50:00	
CMD: A Update B Start C Stop D INGRELS E INGVOTE F INGINFO						
G Members H DISPTRG I INGSCHED J INGGROUP / scroll						
CMD Name Type System Compound Desired Observed Nature						

RED_DB2GRP	APG	SC42	INHIBITED	AVAILABLE	PROBLEM	BASIC

Automatic Restart Manager (ARM) restarted the DB2 subsystem D7X1 on SC04 with the option RESTART(LIGHT) in order to quickly release the retained locks. When the start-up was complete, D7X1 stopped.

SA z/OS restarted the NFS server on SC04.

SA z/OS restarted SCS on the LPAR where the enqueue replication server was running (SC04).

Because we had only two LPARs, the enqueue replication server stopped. If a third LPAR had been available, SA z/OS would have restarted the enqueue replication server on that LPAR.

The SAP application server APPSRV06 running on VMLINUX6 lost the connection to the database server on SC42. The five running transactions received a DB2 SQL error 0 and were rolled back. The work processes were put in a reconnect status. The running sessions were lost and needed to be restarted. The work processes did a failover to the standby database server, reestablished the connection and left the reconnect status.

The transaction DB2 showed that the current DB host was now wtsc04a, as shown in the following. We also checked, with the DB2 command -DIS THREAD(*), that all the threads are connected to SC04. Connection information for each work process can be found in the developer trace files dev_wx.

Verification (z/OS)

```
Settings:
Primary DB host          wtsc42a
Standby DB host         wtsc04a
Present DB host         wtsc04a

Operation:
Operation completed successfully.
New DB host             wtsc04a
```

For the SAP application server APPSRV11 running on SC04, the failure is transparent—the workload is still running (SM66) and the lock entries that we generated are still in the enqueue table (SM12). The developer trace dev_disp shows that the dispatcher lost its connection with the message server and reconnected later on.

The developer trace dev_w3 shows that the work process lost its connection with the enqueue server and reconnected later on as soon as the enqueue server was available.

```
M Tue Jun 18 14:52:46 2002
M MBUF info for hooks: MS component DOWN
M ***LOG R0Z=> ThResetVBDISP, reset update dispatching info () ./thxxvb.c 69
M *** ERROR => ThCheckReqInfo: message send/receive failed ./thxxhead.c 13681
M *** ERROR => ThMsOpcode: ThOpcodeToMsg failed (1) ./thxxmsg.c 2769
M ThVBHd1MsgDown: msg down
M ThIVBChangeState: update deactivated
M ***LOG R0R=> ThIVBChangeState, update deactivated () ./thxxvb.c 9810
M
M Tue Jun 18 14:52:53 2002
M MBUF info for hooks: MS component UP
M *** ERROR => ThSetEnqName: no enqueue server active ./thxxtool.c 4163
M ***LOG R1P=> ThSetEnqName, bad enq configuration () ./thxxtool.c 4167
S server '@>SSRV:wtsc42a_RED_10@<' vanished
S server '@>SSRV:vmlinux6_RED_00@<' vanished
M ThVBHd1MsgUp: msg up
M ThIVBChangeState: update activated
M ***LOG R0T=> ThIVBChangeState, update activated () ./thxxvb.c 9796
M
M Tue Jun 18 14:55:13 2002
M ***LOG Q0I=> NiPRead: rcv (1121: EDC8121I Connection reset.) ./niuxi.c 1198
M ENSA_DoRequest (): Reconnect
```

All SAP-related resources are in UP status after the failover and running on SC04, including the NFS and enqueue servers. The enqueue replication server is stopped.

Problem determination methodology

In this section, we describe how to perform problem determination for SA z/OS and for each of the critical SAP components.

SA z/OS problem determination

SAP HA is a complex environment, and in such an environment problems can occur. In this chapter we direct you to areas where you can check for problems if you encounter various errors.

NetView netlog

All messages flowing to NetView are kept in two VSAM log files, NETLOGP (primary netlog), and NETLOGS (secondary netlog). These log files are used in a wraparound manner. Depending on their size, these log files typically keep from a few hours of data, up to several days of data.

To browse through the active log file, enter this command on the NetView NCCF command line:

```
BR NETLOGA
```

There is also a front-end panel for the netlog browse, which you call by entering this command on the NetView NCCF command line:

```
BLOG
```

BLOG allows for all kinds of filtering. For help information, enter the following command on the NetView NCCF command line:

```
HELP BLOG
```

To save the contents of the netlogs to a printer or a sequential file, you might want to use the procedure CNMPRT, which resides in PROCLIB.

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:

```
.
.
.DEFAULT,SUP(YES),RETAIN(YES),AUTO(YES)
BPXF024I, SUP(YES),RETAIN(YES),AUTO(YES)
```

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).

Verification (z/OS)

- *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
- DISPINFO
- INGINFO

SDF or NMC: The first indication of an unusual situation is often the dynamic display of SDF or NMC. This display shows the status of the resource in question. You can use the help function to learn more about the meaning of the status color of each resource. You can also use the EXPLAIN command on the NetView NCCF command line to see possible statuses and their meanings.

DISPINFO: The DISPINFO screen is not normally called directly from the command line (although it is possible), but rather out of the DISPSTAT panel. Thus you do not have to remember all the parameters; you can use convenient line commands instead. To get to the DISPINFO panel, enter: *f* as indicated in the following:

```
AOFKSTA5 SA OS/390 - Command Dialogs Line 21 of 45
Domain ID = SC04A ----- DISPSTAT ----- Date = 06/21/02
Operator ID = HTWANDR Time = 10:10:28
A ingauto B setstate C ingreq-stop D thresholds E explain F info G tree
H trigger I service J all children K children L all parents M parents
CMD RESOURCE STATUS SYSTEM JOB NAME A I S R D RS TYPE Activity
-----
RED_ERS AUTODOWN SC04 REDADMER Y Y Y Y Y MVS --none--
f RED_ES INACTIVE SC04 REDADMES Y Y Y Y Y MVS --none--
RED_GW INACTIVE SC04 REDADMGW Y Y Y Y Y MVS --none--
RED_MS INACTIVE SC04 REDADMMS Y Y Y Y Y MVS --none--
RED_RFC UP SC04 REDADMR1 Y Y Y Y Y MVS --none--
RED_SE INACTIVE SC04 REDADMSE Y Y Y Y Y MVS --none--
RED_VIPA ENDED SC04 TCPVIPA1 Y Y Y Y Y TRANS --none--
```

The following shows the DISPINFO panel:

```
AOFKINFO SA OS/390 - Command Dialogs Line 1 of 118
Domain ID = SC04A ----- DISPINFO ----- Date = 06/21/02
Operator ID = HTWANDR Time = 10:17:38
Subsystem ==> RED_ES System ==> SC04 System name, domain ID
or sysplex name
Subsystem : RED_ES on System : SC04
Description : SAP Enqueue Server
Class : USS_APPL
Job Name : REDADMES
Job Type : MVS
Category : USS
Current status : INACTIVE
Last Monitored : 10:15:46 on 06/21/02
Last Changed : 15:33:54 on 06/20/02
Last Message :
AOF571I 15:33:54 : RED_ES SUBSYSTEM STATUS FOR JOB REDADMES IS
INACTIVE - FAILED DURING START UP
Monitor : AOFUXMON
Monitor Status : INACTIVE

(---- truncated ---)
```

The DISPINFO panel provides useful information such as the following:

- Actual application status
- Date and time of last status change
- Start and stop commands
- Timeout values and threshold values for this application

INGINFO: The INGINFO screen is not normally called directly from the command line (although it is possible), but rather from the INGLIST panel. Thus you don't have to remember all the parameters; you can use convenient line commands instead:

```

INGKYST0 SA OS/390 - Command Dialogs Line 22 of 45
Domain ID = SC04A ----- INGLIST ----- Date = 06/21/02
Operator ID = HTWANDR Sysplex = WTSCPLX1 Time = 10:24:51
CMD: A Update B Start C Stop D INGRELS E INGVOTE F INGINFO
G Members H DISPTRG I INGSCHED J INGGROUP / scroll
CMD Name Type System Compound Desired Observed Nature
-----
RED_ERS APL SC04 SATISFACTORY UNAVAILABLE SOFTDOWN
f RED_ES APL SC04 PROBLEM AVAILABLE SOFTDOWN
RED_GW APL SC04 PROBLEM AVAILABLE SOFTDOWN
RED_MS APL SC04 PROBLEM AVAILABLE SOFTDOWN
RED_RFC APL SC04 SATISFACTORY AVAILABLE AVAILABLE
RED_SE APL SC04 PROBLEM AVAILABLE SOFTDOWN
RED_VIPA APL SC04 SATISFACTORY AVAILABLE AVAILABLE
RESOLVER APL SC04 SATISFACTORY AVAILABLE AVAILABLE
RMF APL SC04 SATISFACTORY AVAILABLE AVAILABLE

```

The following shows an example of the INGINFO panel.

```

INGKYIN0 SA OS/390 - Command Dialogs Line 1 of 553
Domain ID = SC04A ----- INGINFO ----- Date = 06/21/02
Operator ID = HTWANDR Sysplex = WTSCPLX1 Time = 10:25:32
Resource ==> RED_ES/APL/SC04 format: name/type/system
System ==> System name, domain ID or sysplex name
Resource : RED_ES/APL/SC04
Category : USS
Description : SAP Enqueue Server
Status...
Observed Status : SOFTDOWN
Desired Status : AVAILABLE
Automation Status : PROBLEM
Startable Status : YES
Compound Status : PROBLEM
Dependencies...
PreStart : Satisfied
Start : Satisfied
PreStop : Satisfied
Stop : Satisfied
Startability : Satisfied
(--- truncated ---)

```

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

UNIX messages

By default, UNIX messages will not be sent to the z/OS syslog or to the NetView log. To send UNIX syslogd messages to the z/OS syslog, you must add an entry in the syslogd configuration file `/etc/syslog.conf`.

To forward all messages to the z/OS syslog, add the following entry:

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

The UNIX messages will appear in the z/OS syslog with a BPXF024I message id. To send them further to NetView, you might have to modify MPF (see “Message Processing Facility” on page 277).

If nothing happens

You may encounter a failure situation in which you enter a command to SA for 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. In this section, we describe these causes and show how you can determine where the problem lies:

- A pending vote
 - Use the INGVOTE command to look for open votes.
- Missing supporting applications
 - Check the relationships of the failing application. Are there any unresolved dependencies?
- Pending excludes or avoids against groups
 - Use the INGGROUP command or the SANCHK REXX to find excludes and avoids
- Auto flags in the SA z/OS agent
 - Enter: *DISPSTAT application name* and examine the automation flags. Using SA z/OS 2.1, they usually have to be switched on (Y).
- Disabled automation in the Automation Manager
 - Use the *a* line command on the INGLIST screen against the failing application, and check under action 3 for the automation manager auto flag.

When you are really lost

The last step before giving up and calling IBM support could be to do a cold start of the automation manager (HSAMPROC). A cold start will usually get rid of possible deadlocks, but note the following caveat.

Important: An automation manager cold start will also delete all dynamic overrides to thresholds, automation flags, schedules, preference values, and votes for all systems managed by the automation manager.

Usually the name of the automation managers started task is HSAMPROC, so after shutting down all automation managers (first the secondary, then the primary), enter the following start command at the z/OS system console:

```
s HSAMPROC,sub=mstr,type=cold
```

After the primary automation manager initializes, start the secondary automation managers.

Getting help from the Web

A very useful table called “Tips for startup and shutdown problems” can be found at the following site:

<http://www.ibm.com/servers/eserver/zseries/software/sa/techresources/hint02.html>

It is always worthwhile to browse through this table.

Where to check for application problems

This section describes where to look if SA 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, and it identifies resources and the system where the command has been executed. In our configuration, they are all located in the home directory /u/redadm.

The following command shows the log files in chronological order:

```
ls -rtl *.log
```

- Log file does not exist.

In this case, SA 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 277).
- Check the USS system log (syslogd) for messages.
- Check the availability of file systems. Are the SAP global, profile, and exe directories accessible?
- Logon to USS and execute the command manually.
- For remote resources, the log files usually indicate the reason that SA failed to manage the resource. It may be that the remote resource is not truly unavailable—instead, remote monitoring, or remote execution, may be inhibited.
 - Check that the remote system is available.
 - Check that remote execution works.
 - Log on to the remote system and check the status.

• The SAP application server does not start

- Check messages in the startappsrv*.log file. This file contains the output of the startup command invoked by SA.

For debugging purposes, the script startappsrv_v5 contains an *env* command and a *ulimit* command at the beginning. This way, the process environment is made visible. You may add other commands as needed.

In our configuration, these files are located in the home directory /u/redadm.
- Check messages in the startsap_*.log file. This file contains the output of the *startsap* command, which is invoked by startappsrv_v5.
- Check the SAP development traces in the work directory of the application server instance. List the files in chronological order to see which ones have been written last.

In our configuration, they are located in the directory
/usr/sap/RED/<appserver>/work

Verification (z/OS)

- Check the home directory and the instance work directory for core files or CEEDUMP files indicating an abnormal termination of a UNIX process. Such files are also written if a DLL was not found due to an incorrect LIBPATH environment variable, or a module could not be loaded because of a missing STEPLIB definition.

- **SAP enqueue server, message server, gateway or syslog collector does not come up**

Problem determination in this case is similar to the application server case.

Check messages in the log file that you defined for the SA resource in the SA policy. For example, if the ES was started on the COH1 LPAR then our policy uses `startsap_asc00_ES.COH1.log`.

In our configuration, they are located in the directory

```
/usr/sap/RED/ASCS00/work
```

- For the enqueue server, browse the `enquelog` file in the work directory. It shows when the enqueue server has been started and stopped, and whether the enqueue replication server is activated.
- A common startup problem of the syslog collector is that the global syslog file has become corrupted (this can happen, for example, if the file system is filled up).

The syslog file is located in the global directory and is named `SLOGJ`. Delete the file (the syslog collector will rebuild it automatically on its next startup).

In our configuration, it is located in the directory `/usr/sap/RED/SYS/global`.

- **The application 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 our configuration, the command looks as follows:

```
ensmon -H sapred -I 00 1
```

The command writes further trace information into file `dev_ensmon` in the current directory. If `ensmon` fails on a remote system—but succeeds on the system where the enqueue server is running—the cause is probably a network problem.

Checking the network

Describing how to troubleshoot network problems could probably fill an entire volume. In this section, we mention just a few useful commands that you can use to verify the configuration and the connectivity between the systems. We also list commands to check the existence and location of dynamic VIPAs and the actual routing.

Note: You can issue these commands from different environments, such as: z/OS operator commands (OPER) format, TSO commands, and USS commands.

Checking the configuration

First, check the setup. The following command performs a basic consistency check:

```
TSO: HOMETEST
```

The following commands display the network configuration and attributes.


```
OPER: D TCPIP,,N,CONFIG
TSO: NETSTAT CONFIG
USS: netstat -f
```

The above command allows you to verify what you thought you had specified in the TCP/IP profile. In particular, check the following settings:

- *FORWARDING YES*
- *IGREDIRECT 1*
- *SOURCEVIPA 1*
- *PATHMTUDSC 1*

Note: If you use multiple TCP/IP stacks, you have to specify the name of the stack as the second parameter in the operator commands, as shown in the following example:

```
D TCPIP,TCPIPA,NE,CONFIG
```

Checking network devices

The following commands list the status of the interfaces:

```
OPER: D TCPIP,,N,DEV
TSO: NETSTAT DEV
USS: netstat -d
```

From the above commands, you can see the device status (for example, *READY*) and important facts such as whether it is configured as the PRI router (*CFGROUTER*), and whether it is currently used as the PRI router (*ACTROUTER*).

The next commands display the status of the interfaces, from an OSPF point of view:

```
OPER: D TCPIP,,OMPR,OSPF,IFS
```

Once you know the name of the interface from the second column of the display, you can gather more details by specifying the interface name as an additional parameter on this command:

```
OPER: D TCPIP,,OMPR,OSPF,IFS,NAME=<interface>
```

The *DESIGNATED ROUTER* for this interface is the router that makes all routing table changes for this interface (LAN) and broadcasts them. Of further interest are the *STATE*, the *MAX PKT SIZE*, and the number of *NEIGHBORS* and *ADJACENCIES*.

Dynamic VIPA

The following command displays the location and status of all VIPAs in the sysplex:

```
OPER: D TCPIP,,SYSPLEX,VIPADYN
```

In the USS environment, use the following command to display the list of home addresses (inclusive the VIPAs):

```
USS: netstat -h
```

or just the dynamic VIPAs on the system:

```
USS: netstat -v
```

Routing tables and OSPF

To display routing tables:

Verification (z/OS)

```
OPER: D TCPIP,,N,ROUTE
TSO: NETSTAT ROUTE
USS: netstat -r
```

To display gateways, you can use:

```
TSO: NETSTAT GATE
USS: netstat -g
```

To display OSPF tables:

```
OPER: D TCPIP,,OMPR,RTTABLE
```

Apart from the interface display that was previously explained, you may also want to see if OSPF is talking to its neighbors:

```
OPER: D TCPIP,,OMPR,OSPF,NBRS
```

You can even see statistical counters that show the quality of the conversations:

```
OPER: D TCPIP,,OMPR,OSPF,STATS
```

On AIX and Linux systems, the following command proved to be useful to watch the VIPA takeover among the z/OS systems. The -R option shows the current routing and indicates when the routing changes.

```
ping -R <hostname>
```

Checking active connections

To display all active IP connections on the system:

```
OPER: D TCPIP,,N,CONN
USS: netstat -c (or simply: netstat)
```

With this command you also see whether a static or dynamic VIPA is used as a source address or a target address, allowing you to easily verify that the 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 the Shared HFS and of NFS

With the introduction of the Shared HFS, additional attributes have been added to the file system. They can be checked with the following command:

```
df -kv <filename>
```

Following is an example of the output and as you can see, the file system is currently owned by SC04 and is movable:

```
wtsc04a:/u/redadm (7)>df -kv /usr/sap/RED
Mounted on Filesystem Avail/Total Files Status
/sap/RED (SAPRED.SHFS.SAPUSR) 2069924/2156400 4294966691 Available
HFS, Read/Write, Exported
File System Owner : SC04 Automove=Y Client=N
Filetag : T=off codeset=0
```

The following command allows the operator to check whether NFS clients have mounted a file system, (<MVS NFS> stands for the jobname of the NFS server):

```
F <MVS NFS>,LIST=MOUNTS
```

Consider the case where clients may not have done an explicit unmount (for example, if the connection was disrupted, or the client system was switched off). This usually does not impact the NFS server.

However, if an HFS 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: All clients will need to subsequently remount this NFS file system.

Checking the status of DB2 and SAP connections

In this section, we discuss basic techniques for identifying problems related to the SAP connections to DB2, or DB2 itself; we do not provide a comprehensive description of the general topic of problem determination. Additional problem determination information can be found in the *SAP Database Administration Guide* and the respective *Planning Guide*.

Check that DB2 is running

Use the SDSF DA command to show the status of DB2. (Prior to issuing this command, you can set your SDSF prefix to limit the display to DB2.)

For our configuration, we issued the following SDSF command:

```
pre d7x*
da
```

The following figure shows the results of these commands for our configuration. If the display doesn't show the DB2 systems running, then check the z/OS system log for messages (refer to "z/OS syslog" on page 277).

Display Filter View Print Options Help											

SDSF DA SC42 SC42 PAG 0 SIO 86 CPU 11/ 10 LINE 1-5 (5)											
COMMAND INPUT ==>											
PREFIX=D7X* DEST=(ALL) OWNER=* SORT=JOBNAME/A SYSNAME=SC42											
SCROLL ==> CSR											
NP	JOBNAME	StepName	ProcStep	JobID	Owner	C	Pos	DP	Real	Paging	SIO
	D7X1DBM1	D7X1DBM1	IEFPROC	STC16226	STC	NS	FE	127T	0.00	0.00	
	D7X1DIST	D7X1DIST	IEFPROC	STC16229	STC	NS	FE	369	0.00	0.00	
	D7X1IRLM	D7X1IRLM		STC16222	STC	NS	FE	862	0.00	0.00	
	D7X1MSTR	D7X1MSTR	IEFPROC	STC16221	STC	NS	FE	773	0.00	0.18	
	D7X1SPAS	D7X1SPAS	IEFPROC	STC16231	STC	NS	FE	1136	0.00	0.00	

Figure 78. Results of SDSF DA command

Check the SAP database connections

- Use the DB2 Display Thread command to show the connections to DB2 from the SAP application server on USS. The following is the command we used:

```
-D7X1 DISPLAY THREAD(*)
```

The following figure shows the results of this command for our configuration. Notice that we have two application servers connected to DB2, wtsc42a (the USS application server), and vmlinux6 (the Linux application server).

```

Display Filter View Print Options Help
-----
SDSF OPERLOG DATE 07/02/2002 6 WTORS COLUMNS 52- 131
COMMAND INPUT ==> SCROLL ==> CSR
000290 -D7X1 DISPLAY THREAD(*)
000090 DSNV401I -D7X1 DISPLAY THREAD REPORT FOLLOWS -
000090 DSNV402I -D7X1 ACTIVE THREADS - 159
000090 NAME ST A REQ ID AUTHID PLAN ASID TOKEN
000090 RRSAF T 700 172021011001 REDADM CRED46C 0083 40
000090 V437-WORKSTATION= , USERID=*,
000090 APPLICATION NAME=wts42a
000090 RRSAF T 4302 172021011001 REDADM CRED46C 0083 41
000090 V437-WORKSTATION= # 6 h , USERID=*,
000090 APPLICATION NAME=wts42a
000090 RRSAF T 36 172021011001 REDADM SAPR346D 0070 38
000090 V437-WORKSTATION=6 00014 0000852704, USERID=*,
000090 APPLICATION NAME=wts42a
000090 RRSAF T 3067 172021011001 REDADM SAPR346D 007D 37
000090 V437-WORKSTATION=2 00013 0000852703, USERID=*,
000090 APPLICATION NAME=wts42a
.....
.....
000090 RRSAF T 23362 192168050006 REDADM FOME46D 008A 14
000090 V437-WORKSTATION=1 00001 0000006748, USERID=10D6FA0000000006,
000090 APPLICATION NAME=vmlinux6
000090 RRSAF T 10362 192168050006 REDADM FOME46D 008A 15
000090 V437-WORKSTATION=1 00002 0000006749, USERID=10D78E20000000007,
000090 APPLICATION NAME=vmlinux6
000090 RRSAF T 220 192168050006 REDADM FOME46D 008A 17
000090 V437-WORKSTATION=1 00003 0000006750, USERID=10D7BF80000000008,
000090 APPLICATION NAME=vmlinux6
000090 RRSAF T 224 192168050006 REDADM FOME46D 008A 18
000090 V437-WORKSTATION=1 00005 0000006752, USERID=10D7D83000000000A,
000090 APPLICATION NAME=vmlinux6
.....
.... Shortened ....

```

Figure 79. Results of DB2 Display Thread command

Availability test scenarios

This section lists defined test scenarios for availability. It is to serve as a reference list to assist you in your availability planning. The test results shown here are the results for a single specific environment. We believe they show what is achievable, but you should select the items most important in your installation and test those rather than rely on these results.

The following table shows an extensive list of failure scenarios. Tests were performed in a specific environment to determine the impact of such a failure. The impact shown in the “Effect” column assumes the availability plan for that environment is followed.

Table 33. High availability test scenarios

High availability scenario	Effect
z/OS failure	No impact / SQL 0000
z/OS system upgrade	No impact / SQL 0000
DB2 failure and automatic restart	No impact / SQL 0000
DB2 upgrade	No impact / SQL 0000
Coupling Facility failure	No impact
Coupling Facility link failure	No impact

Table 33. High availability test scenarios (continued)

High availability scenario	Effect
Coupling Facility takeover	No impact
Channel path (CHPID) failure	No impact
OSA-Express failure	No impact / SQL 0000
Central processor complex failure	No impact / SQL 0000
CPU engine failure	No impact
DB2 online full image copy utility	No impact
Incremental copies and merge copy full	No impact
DB2 online REORG utility	Slow responses
Recover SAP tablespace to current state	Short term outage
DB2 point in time recovery	Outage during recovery
Dynamically adjust the hardware CPU capacity	No impact
Dynamically add XCF signaling paths	No impact
Dynamically adjust dispatching priority	No impact
Loss of redundant power supply or cooling unit	No impact
Loss of redundant utility power	No impact
Dynamically add DASD volumes	No impact
Dynamically add DASD work space (2 TESTS)	No impact
Hardware management console concurrent patch	No impact
Support Element (SE) failure	No impact
Support Element concurrent patch	No impact
Activate daylight savings time	No impact
De-activate daylight savings time	No impact
Sysplex timer link failure	No impact
Sysplex timer failure	No impact

Verification (z/OS)

Chapter 16. Verifying your implementation on Linux/AIX

Verification procedure and failover scenarios

The scenarios cover both planned outages (normal operation, maintenance) and unplanned outages (failures). Each scenario should be verified for proper operation.

Test setup

The following scenarios expect the topology, as defined in the sample policy, to be a cluster with three nodes (lnxsapg, lnxsaph, and lnxsapi). We have floating groups for the SAProuter and the enqueue and replication servers, and fixed groups for one application server on each node.

You can use the `lssap` command to monitor the reaction of the system to the actions taken.

Scenarios

Table 34 and Table 35 on page 292 list the important scenarios. The shaded cells describe the preconditions for executing the scenario. Each scenario is divided into subactions, where each subaction's precondition is the execution of its predecessor. The commands to be executed are listed. If you change the naming, you might have to adapt the commands accordingly. The last column of the tables lists the expected result.

Table 34. Planned outages

Scenario	Action	Command	Expected Result
Normal operation	Precondition: All groups offline		
	Start an SAP system and related components (SAProuter)	<code>chrg -o online -s "Name like 'XI_%'"</code>	ROUTER, ENQ and D95 groups start on lnxsapg. ENQREP and D96 groups start on lnxsaph. D97 group starts on lnxsapi.
	Stop SAP system EP0	<code>chrg -o offline -s "Name like 'XI_%_EP0%'"</code>	ENQ, ENQREP, D95, D96, and D97 groups stop.
	Start SAP system EP0	<code>chrg -o online -s "Name like 'XI_%_EP0%'"</code>	ENQ and D95 groups start on lnxsapg. ENQREP and D96 groups start on lnxsaph. D97 group starts on lnxsapi.
	Stop entire SAP	<code>chrg -o offline -s "Name like 'XI_%'"</code>	All groups stop.

Verification (Linux/AIX)

Table 34. Planned outages (continued)

Scenario	Action	Command	Expected Result
Maintenance	Precondition: ROUTER, ENQ, and D95 groups online on lnxsapg. ENQREP and D96 online on lnxsaph. D97 online on lnxsapi.		
	Move all resources away from one node in order to apply operating system or hardware maintenance.	samctrl -u a lnxsapg	ROUTER, ENQ and D95 groups stop. D95 groups failed offline. ROUTER group starts on lnxsaph. ENQ group starts on lnxsaph. ERS terminates. ENQREP group stops. ENQREP group starts on lnxsapi.
		(Apply maintenance, reboot, etc.)	
		samctrl -u d lnxsapg ...	D95 group starts on lnxsapg.
	Stop and restart ES and/or ERS in order to apply SAP maintenance (code or profile changes).	chrg -o offline XI_ABAP_EPO_ENQREP chrg -o offline XI_J2EE_EPO_ENQREP	ENQREP groups stop
		chrg -o online XI_ABAP_EPO_ENQREP chrg -o online XI_J2EE_EPO_ENQREP	ENQREP groups start on lnxsapg
		chrg -o offline XI_ABAP_EPO_ENQ chrg -o offline XI_J2EE_EPO_ENQ	ENQ groups stop, but MS, ES, and IP stay online because of relation from ERS and AS. ENQ group status is Pending Offline.
		chrg -o online XI_ABAP_EPO_ENQ chrg -o online XI_J2EE_EPO_ENQ	Offline resources of ABAP ENQ groups restart on lnxsaph. Note: This is <i>not</i> the way to move the ENQ groups! You need to initiate a failover by stopping ES outside SA control.

Table 34. Planned outages (continued)

Scenario	Action	Command	Expected Result
Maintenance	Stop and restart ES and/or ERS in order to apply SAP maintenance (code or profile changes).	<pre>rgreq -o move -n lnxsaph XI_ABAP_LOP_ENQ</pre>	<p>For releases of SA MP up to and including 2.2 ENQREP stops first on lnxsapg, because SA MP first stops all collocated resources, and the ERS is collocated because of the 'collocation if not offline' relationship. This means that the enq.table data is no longer replicated. The ES and ENQ groups then stop on lnxsaph. The ES and ENQ groups are restarted on lnxsapg. However, because the ERS is no longer running, the ES <i>cannot</i> recreate the enqueue table data. Finally, the ERS starts on lnxsaph.</p> <p>Depending on the nodelist of the ENQ group, SA MP may restart the ES on lnxsapi, where no ERS was running at all. Also, in this case the ES clearly <i>cannot</i> recreate the enqueue table data.</p> <p>Note: For releases of SA MP up to and including 2.2 this is <i>not</i> the way to move the ENQ groups! You need to initiate a failover by stopping ES outside System Automation control.</p> <p>For releases of SA MP starting with release 2.3, the behavior of the move has changed. Now, a node exclusion approach is used in which only resources within the move scope that are currently running on the nodes specified in the move request or which have a dependency on these resources are stopped. This means, that the ERS is not stopped during the move. Move can now be used to move the ENQ groups.</p>
		<pre>stopprsrc -s "Name= 'XI_ABAP_EP0_ENQ_ES'" IBM.Application</pre>	ES stops, ENQ group stops and restarts on lnxsapg, ERS terminates, ENQREP group stops and restarts on lnxsaph.
	Stop and restart D95 in order to apply SAP maintenance (code or profile changes).	<pre>chrg -o offline XI_J2EE_EP0_lnxsapg_D95 chrg -o offline XI_ABAP_EP0_lnxsapg_D95</pre>	D95 groups stop
		<pre>chrg -o online XI_ABAP_EP0_lnxsapg_D95 chrg -o online XI_J2EE_EP0_lnxsapg_D95</pre>	D95 groups restart on lnxsapg.
Stop and restart z/OS LPAR or DB2 in order to apply maintenance.	<p>This is transparent to SA MP. It will be handled by SA z/OS, by the built-in z/OS or SAP failover mechanisms, or both.</p> <p>Note: SAP currently does not support an SAP failover mechanism for Java application servers.</p>		

Verification (Linux/AIX)

Table 35. Unplanned outages

Scenario	Simulation Action/Command	Expected Result
Precondition: ROUTER, ENQ, and D95 groups online on Inxsapg. ENQREP and D96 online on Inxsaph. D97 online on Inxsapi.		
Failure of the ABAP enqueue server	Inxsapg: killall -9 es.sapEP0_ASCS00	ABAP ENQ group stops and restarts on Inxsaph. ABAP ERS terminates. ABAP ENQREP group stops and restarts on Inxsapg. See also "Testing an unplanned outage of an ABAP SCS" on page 293.
Failure of the enqueue replication server.	Inxsapg: killall -9 ers.sapEP0_ASCS00	ABAP ENQREP group stops and restarts on Inxsapg.
Failure of the message server	Inxsaph: killall -9 ms.sapEP0_ASCS00	ABAP MS restarts on Inxsaph.
Failure of an ABAP or double-stack application server	For an ABAP only stack: Inxsapg: killall -9 dw.sapEP0_D95 For double-stack: Inxsapg: killall -2 -g dw.sapEP0_D95 killall -2 -g /usr/sap/EP0/D95/igs/bin /igs* killall -2 se.sapEP0_D95 killall -2 -g jcontrol	D95_AS restarts on Inxsapg.
Failure of the Java enqueue server	Inxsapg: killall -9 es.sapEP0_SCS01	Java ENQ group stops and restarts on Inxsaph Java ERS terminates Java ENQREP group stops and restarts on Inxsapg See also "Testing an unplanned outage of a Java SCS" on page 294.
Failure of the Java enqueue replication server	Inxsapg: killall -9 ers.sapEP0_SCS01	Java ENQREP group stops and restarts on Inxsapg
Failure of the Java message server	Inxsaph: killall -9 ms.sapEP0_SCS01	Java MS restarts on Inxsaph
Failure of a Java-only application server	Inxsapg: killall -2 jc.sapEP0_J95	J95 AS restarts on Inxsapg

Table 35. Unplanned outages (continued)

Scenario	Simulation Action/Command	Expected Result
Failure of the node where ES is running	Inxsaph: reboot	softdog kills Inxsaph because IP is a critical resource. ENQ groups start on Inxsapg. ERS terminates. ENQREP groups stop and restart on Inxsapi. D97 group starts on Inxsaph as soon as Inxsaph is back in the cluster.
Failure of the node where ERS is running.	Inxsapi: reboot	Inxsapi reboots (no critical resource online there). ENQREP group starts immediately on Inxsaph. D97 group starts on Inxsapi as soon as Inxsapi is back in the cluster.
Failure of TCP/IP, OSPF, network adapter on z/OS. Failure of a z/OS LPAR.	This is transparent to SA MP. It will be handled by SA z/OS and/or the built-in z/OS or SAP failover mechanisms. Note: SAP does not currently support the SAP failover mechanism for Java application servers.	
Simulate failure of a node, where HA NFS server and Master IBM.RecoveryRM were active.	<ol style="list-style-type: none"> Find the Master RecoveryRM node, for example through: <pre>lssrc -ls IBM.RecoveryRM grep Master</pre> This produces an output similar to: Master Node Name : Inxsapi (node number = 3) Move the HA NFS server to the node, where the Master is active, here to Inxsapi. Reboot using <code>reboot -nf</code> or crash the VM guest, using for example, <code>SIGNAL SHUTDOWN [guest]</code> and <code>XAUTOLOG [guest]</code>. 	The new Master RecoverRM should restart the NFS server on one of the online nodes. Also all other floating resources which were online on the crashed node should failover.

Testing an unplanned outage of an ABAP SCS

Note: This test does not apply to a Java-only environment such as EP 6.0.

Check which machines your ABAP enqueue server and its replication server are running on. Then use SAP transaction SM12 to generate entries in the enqueue table. This is described in “Preparation for the test (unplanned outage only)” on page 254. To simulate an unplanned outage, kill the ABAP enqueue server process via:

```
killall -9 es.sapEP0_ASCS00
```

SA MP should then restart the ABAP enqueue server on the machine where the ABAP replication server is active. Perform the listed steps to verify that the enqueue failover was transparent for the SAP system. These steps are also described in “Verifications after the test” on page 257.

Testing an unplanned outage of a Java SCS

Check which machines your Java enqueue server and its replication server are running on. Use the 'enqt' utility to verify that enqueue replication works as expected. As <sapsid>adm, run the following from the predefined-policy installation directory for SA MP (with your adapted enqt profile):

```
enqt pf=enqt.pf_scs01 97
```

You will see output such as:

```
-----REQ-----  
EnqId:           EnqTabCreaTime/RandomNumber   = 06.09.2005 00:06:19 1125957979 / 8563  
ReqOrd at Srv:  TimeInSecs/ReqNumberThisSec    = 09.09.2005 13:45:43 1126266343 / 1  
-----
```

The EnqId is the unique identifier of the enqueue server and its enqueue table, and you should remember it. Also run:

```
enqt pf=enqt.pf_scs01 20
```

You will see output like:

```
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.

To simulate an unplanned outage, kill the Java enqueue server process via:

```
killall -9 es.sapEP0_SCS01
```

SA MP should then restart the Java enqueue server on the machine where the Java replication server is active. Run both 'enqt' commands again and check that the EnqId and enqueue table entries are the same as before the Java SCS failover. If so, you have verified that the enqueue table replication is working properly.

Part 7. Appendixes

Appendix A. Network setup

This appendix briefly describes a setup of a highly available network that was part of a test implementation of a high availability solution for SAP on DB2 for z/OS. It lists samples of important configurations (or portions thereof). In a highly available network, all network components are eliminated as a single point of failure. This can be achieved by duplicating all network components to obtain the necessary redundancy. The following figure shows our test setup:

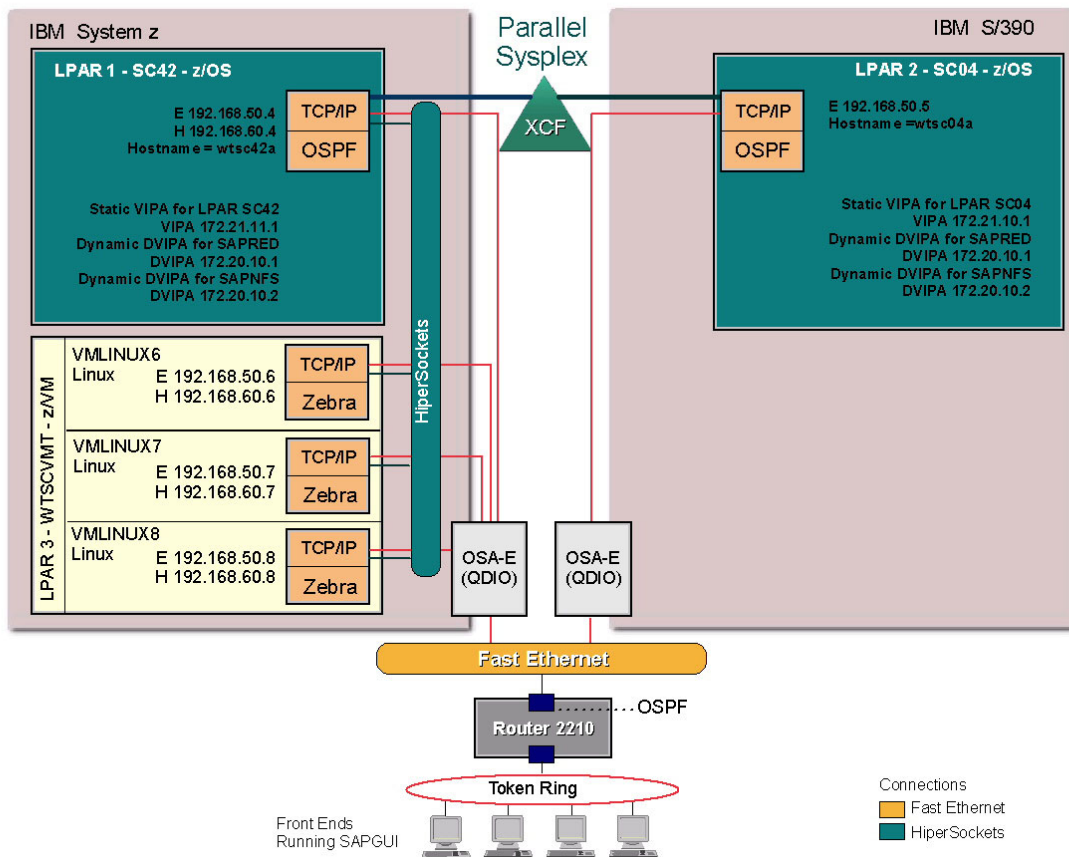


Figure 80. Networking configuration for the high availability solution for SAP

Network hardware components for the test setup

We used the following hardware:

- OSA-Express Fast Ethernet adapter (shared between LPARs)
- HiperSockets

The OSA-Express Fast Ethernet adapter and HiperSockets give us more than one path between the remote (Linux on System z) application servers and the database servers.

Networking software components for the test setup

z/OS network settings for the test setup

The software components and important settings used on z/OS in the test environment depicted above are described below.

z/OS VIPAs

We implemented the following VIPAs:

- Dynamic VIPA definitions for:
 - SCS
 - NFS server and/or DFS SMB
 - SAP network interface router (saprouter)

z/OS UNIX System Services setup - BPXPRMxx

Following is a portion of the BPXPRMxx parmlib member used by both LPARs. It shows the network definitions for the TCP/IP stacks and NFS client definitions. It is executed on both LPARs.

```

/*****
/*          BPXPRMxx  PARMLIB Member          */
/*****
FILESYSTYPE TYPE(HFS) /* HFS */
ENTRYPOINT(GFUAINIT)
FILESYSTYPE TYPE(TFS) /* TFS */
ENTRYPOINT(BPXTFS)
FILESYSTYPE TYPE(IBMUDS)
ENTRYPOINT(BPXTUINIT)
FILESYSTYPE TYPE(INET)
ENTRYPOINT(EZBPFINI)
FILESYSTYPE TYPE(NFS) /* NFS */
ENTRYPOINT(GFSCINIT)
PARM('DISABLELLA(y)')
ASNAME(MVSNFSCS,'SUB=MSTR')
FILESYSTYPE TYPE(AUTOMNT) /* AMD */
ENTRYPOINT(BPXTAMD)
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)
/*****

```

z/OS LPAR SC42

In this section, we describe the network settings for LPAR SC42. These settings were also used for LPAR SC04 by replacing those values specific to SC04.

File /etc/resolv.conf - SC42:


```
TCPIPJobname TCPIPA      ;
Datasetprefix TCPIPA    ;
Messagecase mixed       ;
HostName wtsc42a        ;
DomainOrigin itso.ibm.com ;
NSinterAddr 9.12.2.7    ;
NSportAddr 53           ;
ResolveVia UDP          ;
ResolverTimeout 10      ;
ResolverUdpRetries 1    ;
```

TCP/IP profile - SC42:

```

;
-----
;
; Flush the ARP tables every 20 minutes.
;
ARPAGE 20
; GLOBALCONFIG: Provides settings for the entire TCP/IP stack
;
GLOBALCONFIG NOTCPIPSTATISTICS
;
; IPCONFIG: Provides settings for the IP layer of TCP/IP.
;
IPCONFIG
;
  ARPTO 1200          ; In seconds
DATAGRamfwd
SOURCEVIPA
VARSUBNETTING        ; For RIP2
PATHMTUDISCOVERY
SYSPLEXRouting
DYNAMICXCF 192.168.40.4 255.255.255.0 2
IGNORERedirect
REASSEMBLTimeout 15 ; In seconds
STOPONclawerror
TTL 60                ; In seconds, but actually Hop count
SACONFIG COMMUNITY MVSsub1
ENABLED
AGENT 161
; Dynamic VIPA definitions
VIPADYNAMIC
VIPARANGE DEFINE MOVEABLE DISRUPTIVE 255.255.255.0 172.20.10.0
ENDVIPADYNAMIC

;SOMAXCONN: Specifies maximum length for the connection request queue
; created by the socket call listen().
;
SOMAXCONN 10
;
; TCPCONFIG: Provides settings for the TCP layer of TCP/IP.
;
TCPCONFIG INT 10 SENDG TRUE UNRESTRICTL TCPCV 131072 TCPSENDB 131072
;
; UDPCONFIG: Provides settings for the UDP layer of TCP/IP
;
UDPCONFIG UNRESTRICTLOWPORTS
;
-----
-
;
; Reserve low ports for servers
;
TCPCONFIG RESTRICTLOWPORTS
UDPCONFIG RESTRICTLOWPORTS
;
;
-----
--
;
; AUTOLOG the following servers
AUTOLOG 5
FTPJOBNAME FTPDA      ; FTP Server
PMAPA                 ; Portmap Server
OMPROUTE              ; OMPROUTE (OSPF)
MVSNFSSA ;;;;;;;;;;; Only for primary
ENDAUTOLOG

```

```

;
;
;
;
-----
--
; Reserve ports for the following servers.
;
; NOTES:
;
;   A port that is not reserved in this list can be used by any
user.
;   If you have TCP/IP hosts in your network that reserve ports
;   in the range 1-1023 for privileged applications, you should
;   reserve them here to prevent users from using them.
;
;   The port values below are from RFC 1060, "Assigned Numbers."
;
PORT
  20 TCP OMVS           ; OE FTP Server
      DELAYACKS        ; Delay transmission acknowledgements
  21 TCP OMVS           ; OE FTPD control port
  23 TCP OMVS           ; OE Telnet Server
  80 TCP OMVS           ; OE Web Server
  111 TCP OMVS          ; Portmap Server
  111 UDP OMVS          ; Portmap Server
  135 UDP LLBD          ; NCS Location Broker
  161 UDP SNMPD         ; SNMP Agent
  162 UDP SNMPQE        ; SNMP Query Engine
  512 TCP RXPROCA       ; Remote Execution Server
  514 TCP RXPROCA       ; Remote Execution Server
  520 UDP OMPROUTE      ; OMPROUTE Server
  580 UDP NCPROUTE      ; NCPROUTE Server
  750 TCP MVSKERB       ; Kerberos
  750 UDP MVSKERB       ; Kerberos
  751 TCP ADM@SRV       ; Kerberos Admin Server
  751 UDP ADM@SRV       ; Kerberos Admin Server
  2000 TCP IOASRV       ; OSA/SF Server
  2049 UDP MVSNFSSA     ; Our NFS Server
;
-----
;
;
; Hardware definitions:

DEVICE OSA2880 MPCIPA      PRIROUTER
LINK   OSA2880LNK IPAQENET OSA2880

DEVICE STAVIPA1 VIRTUAL 0      ; Static VIPA
definitions
LINK   STAVIPA1L VIRTUAL 0 STAVIPA1

DEVICE IUTIQDEE MPCIPA
LINK   HIPERLEE IPAQIDIO      IUTIQDEE
;
-----
--
;
; HOME internet (IP) addresses of each link in the host.
;
; NOTE:
;
;   The IP addresses for the links of an Offload box are specified
in
;   the LINK statements themselves, and should not be in the HOME
list.
;
; HOME

```

Network setup

```
172.21.11.1    STAVIPA1L
192.168.60.4   HIPERLEE
192.168.50.4   OSA2880LNK
;
-----
--
;
; IP routing information for the host. All static IP routes should
; be added here.
;
GATEWAY
  192.168.50 =   OSA2880LNK  1500    0
  192.168.60 =   HIPERLEE    32768  0
;
DEFAULTNET 192.168.50.75 OSA2880LNK 1500    0
;
-----
--
; Turn off all tracing. If tracing is to be used, change the
; following
; line. To trace the configuration component, for example, change
; the line to ITRACE ON CONFIG 1

ITRACE OFF
;
;
-----
-
; The ASSORTEDPARMS NOFWD will prevent the forwarding of IP packets
; between different networks. If NOFWD is not specified, IP packets
; will be forwarded between networks when this host is a gateway.
;
; Even though RESTRICTLOWPORTS was specified on TCPCONFIG and
; UDPCONFIG,
; ASSORTEDPARMS default would have been to reset RESTRICTLOWPORTS to
; off
; So it is respecified here.
; If the TCPCONFIG and UDPCONFIG followed ASSORTEDPARMS,
; RESTRICTLOWPORT
; would not have to be done twice.

ASSORTEDPARMS
; NOFWD
; RESTRICTLOWPORTS
ENDASSORTEDPARMS
; Start all the defined devices.
;
START OSA2880
START IUTIQDEE
```

OMPROUTE started task - SC42:

```

//OMPROUTA PROC
//OMPROUTE EXEC PGM=BPXBATCH,REGION=4096K,TIME=NOLIMIT,
// PARM='PGM /usr/lpp/tcpip/sbin/omproute'
//* 'ENVAR("_CEE_ENVFILE=DD:STDENV")/'
//*
//* PARM=('POSIX(ON)',
//* 'ENVAR("_CEE_ENVFILE=DD:STDENV")/-t1')
//*
//* Provide environment variables to run with the
//* desired stack and configuration. As an example,
//* the file specified by STDENV could have these
//* three lines in it:
//*
//* RESOLVER_CONFIG=//SYS1.TCPPARMS(TCPDATA2)'
//* OMPROUTE_FILE=/u/usernnn/config.tcpcs2
//* OMPROUTE_DEBUG_FILE=/tmp/logs/omproute.debug
//*
//* For information on the above environment variables,
//* refer to the IP CONFIGURATION GUIDE.
//*
//STDENV DD DSN=TCPIPA.&SYSNAME..OMPROUTA.ENNVARS,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*

```

ENVVARS - SC42:

```

RESOLVER_CONFIG=/etc/resolv.conf
OMPROUTE_FILE=/etc/omproute.conf
OMPROUTE_OPTIONS=hello_hi
OMPROUTE_DEBUG_FILE=/tmp/omproute.debug

```

Define the OMPROUTA 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 - SC42: 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.

Network setup

```
Area
  Area_Number=0.0.0.0
  Stub_Area=NO
  Authentication_Type=None
  Import_Summaries=YES
;
ROUTERID=10.101.4.132;
;
;AS_Boundary_Routing
;  Import_Static_Routes=Yes
;
Interface
  IP_Address=9.152.24.249
  Name=COH2L1
  Subnet_Mask=255.255.252.0
  MTU=1492
;
OSPF_Interface
  IP_Address=10.101.4.132
  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=10.101.5.132
  IP_Address=10.101.5.132
  Name=COHVS3
  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=10.101.4.68
  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
  IP_Address=10.101.5.68
  Name=HIPERE5
  Subnet_Mask=255.255.255.192
  Attaches_To_Area=0.0.0.0
  MTU=16384
  Cost0=10
  Router_Priority=11
  Parallel_OSPF=Primary
  Hello_Interval=10
  Dead_Router_Interval=40
  Retransmission_Interval=5
  Retransmission_Interval=5
  DB_Exchange_Interval=120
;
OSPF_Interface
  IP_Address=10.101.6.68
  Name=HIPERE6
  Subnet_Mask=255.255.255.192
  Attaches_To_Area=0.0.0.0
  MTU=32768
  Cost0=10
  Router_Priority=11
  Parallel_OSPF=Primary
  Hello_Interval=10
  Dead_Router_Interval=40
  Retransmission_Interval=5
  Retransmission_Interval=5
  DB_Exchange_Interval=120
;
OSPF_Interface
  IP_Address=10.101.7.68
  Name=HIPERE7
  Subnet_Mask=255.255.255.192
  Attaches_To_Area=0.0.0.0
  MTU=57344
  Cost0=10
  Router_Priority=11
  Parallel_OSPF=Primary
  Hello_Interval=10
  Dead_Router_Interval=40
  Retransmission_Interval=5
  Retransmission_Interval=5
  DB_Exchange_Interval=120
;
;OSPF_Interface statement for a virtual (VIPA) interface
;
OSPF_Interface
  IP_Address=10.101.4.212
  Name=VLINK1
  Subnet_Mask=255.255.255.255
  Attaches_To_Area=0.0.0.0
  Cost0=1
  Cost0=1
  MTU=8992
;
;OSPF_Interface statement for VIPARANGE
;
OSPF_Interface
  IP_Address=10.101.5.192
  Subnet_Mask=255.255.255.240
  Name=VRANGEIF
;

```

Linux on System z network settings for the test setup

In this section, we describe the network settings for Linux on System z.

Quagga setup - OSPF

```
hostname Ospfd
password quagga
enable password quagga
!
interface dummy0
ip ospf cost 1
!
interface eth1
ip ospf cost 15
!
interface eth2
ip ospf cost 15
!
interface hsi0
ip ospf cost 10
!
interface hsi1
ip ospf cost 10
!
interface hsi2
ip ospf cost 10
!
interface hsi3
ip ospf cost 10
!
router ospf
ospf router-id 9.152.27.138
network 10.101.4.216/32 area 0
network 10.101.4.128/26 area 0
network 10.101.5.128/26 area 0
network 10.101.4.64/26 area 0
network 10.101.5.64/26 area 0
network 10.101.6.64/26 area 0
network 10.101.7.64/26 area 0
!
line vty
!
log file /var/log/quagga/ospfd.log
```

Zebra setup - Zebra

```
hostname Router
password quagga
enable password quagga
ip route 0.0.0.0/0 9.152.24.1
route-map vipal permit 10
match ip address prefix-list DEST
set src 10.101.4.216
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

Sample /etc/ospf.conf containing the OSPF definitions for 'gated':

```
#####
# Config file of the gated daemon #
#####
routerid <IP address, VIPA address recommended>;
rip off
egp off;
bgp off;
ospf yes {
  backbone {
    networks {
      10.99.30.0 mask 255.255.255.0;
      10.99.31.0 mask 255.255.255.0;
      # here the entry for a VIP network of 10.99.2.0 with
      # network mask 255.255.255.0
      # 10.99.2.0 mask 255.255.255.0;
    };
    interface 10.99.30.54 cost 10 {
      enable;
      #
      # hellointerval 10;
      # routerdeadinterval 40;
      # retransmitinterval 5;
      # priority 20;
    };
    interface 10.99.31.54 cost 10 {
      enable;
      #
      # hellointerval 10;
      # routerdeadinterval 40;
      # retransmitinterval 5;
      # priority 20;
    };
    # here the entry for a VIPA of 10.99.2.1
    # interface 10.99.2.1 cost 10 {
    # enable;
    #
    # hellointerval 10;
    # routerdeadinterval 40;
    # retransmitinterval 5;
    # priority 20;
    # };
  };
};

;import proto ospfase { } ;
;export proto ospfase { } ;
#####
# End of file #
#####
```

Domain Name Server (DNS) definitions

Here are the DNS entries in flat format that we entered for our network.

```
sapred           A      172.20.10.1
sapnfs           A      172.20.10.2
saproute        A      172.20.10.3
vmlinux6        A      192.168.50.6
wtsc04a         A      172.21.10.1
wtsc42a         A      172.21.11.1

172.20.10.1     PTR    sapred
172.20.10.2     PTR    sapnfs
172.20.10.3     PTR    saproute
192.168.50.6    PTR    vmlinux6
172.210.10.1    PTR    wtsc04a
172.21.11.1     PTR    wtsc42a
```

Static VIPA definitions required for SLES-10

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"
```

Now create a definition for the dummy0 interface `/etc/sysconfig/network/ifcfg-dummy0`:

```
BOOTPROTO="none"
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 above new dummy0 interface can be started using ifup, and it will be 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"
```

The above will cause a script named setvipa to be run when ifup processes the local interfaces ifcfg-* file.

Here is an 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"
```

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"
```

Network setup

Appendix B. File system setup

This appendix includes the NFS server and client procedures with export and attribute files, and file system statements in the BPXPRM member in SYS1.PARMLIB. It also includes the Linux mount commands.

NFS server procedure

```
//MVSNFSSA PROC MODULE=GFSAMAIN,
//          SYSNFS=SYS1,NFSRFX=OS390NFS,
//          TCPIP=TCPIPA,
//          TCPDATA=TCPDATA
//*****
//*
/* NFS SERVER WITH VIPA FAILOVER SUPPORT
/* VIPA: SAPNFS = 172.20.10.2 ON STACK TCPIPA
/*
/*
//*****
//DEFVIPA EXEC PGM=MODDVIPA,REGION=OK,TIME=1440,
//          PARM='POSIX(ON) ALL31(ON) / -p TCPIPA -c 172.20.10.2'
//
//GFSAMAIN EXEC PGM=&MODULE,REGION=0M,TIME=1440,COND=(4,LT),
//          PARM=(,
//          'ENVAR("BPXK_SETIBMOPT_TRANSPORT=TCPIPA")/')
//SYSTCPD DD DISP=SHR,DSN=&TCPIP. .;&SYSNAME. .TCPARMS(&TCPDATA.)
//STEPLIB DD DISP=SHR,DSN=&SYSNFS. .NFSLIB
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//NFSATTR DD DISP=SHR,DSN=&NFSRFX. .SAPRED.PARMS(ATTRIB)
//EXPORTS DD DISP=SHR,DSN=&NFSRFX. .SAPRED.PARMS(EXPORTS)
//NFSLOG1 DD DISP=SHR,DSN=&NFSRFX. .SAPRED.SERVER.LOG1
//NFSLOG2 DD DISP=SHR,DSN=&NFSRFX. .SAPRED.SERVER.LOG2
//FHDBASE DD DISP=SHR,DSN=&NFSRFX. .SAPRED.FHDBASE1
//FHDBASE2 DD DISP=SHR,DSN=&NFSRFX. .SAPRED.FHDBASE2
//NFSXLAT DD DISP=SHR,DSN=&NFSRFX. .SAPRED.XLAT
```

NFS export file

Following is our export file content. Note that this new example uses the new <root> option, and also grants Read/Write access to the NFS Clients listed by their VIPA addresses:

File system setup

```
#####  
#  
# OS/390 Network File System Server EXPORTS #  
# #  
#####  
#  
/hfs/sapmnt/HA1/exe -rw=10.101.4.214<root>|\  
10.101.4.215<root>|\  
10.101.4.216<root>  
#  
/hfs/sapmnt/HA1/linux_s390x/exe -rw=10.101.4.214<root>|\  
10.101.4.215<root>|\  
10.101.4.216<root>  
#  
/hfs/sapmnt/HA1/profile -rw=10.101.4.214<root>|\  
10.101.4.215<root>|\  
10.101.4.216<root>  
#  
/hfs/sapmnt/HA1/global -rw=10.101.4.214<root>|\  
10.101.4.215<root>|\  
10.101.4.216<root>  
#  
/hfs/sapmnt/HA1/trans -rw=10.101.4.214<root>|\  
10.101.4.215<root>|\  
10.101.4.216<root>  
#
```

NFS attribute file

Following is our attribute file content:

```
space(100,10), blksize(0), lrecl(80)
norlse
recfm(fb), blksize(0), lrecl(80)
dsorg(ps)
dsntype(pds)
dir(25)
keys(64,0)
recordsize(512,4K)
nonspanned
shareoptions(3,3)
attrtimeout(120), readtimeout(90), writetimeout(30)
text
CRLF
blankstrip
mapleaddot
maplower
nlm
retrieve
nofastfilesize
setownerroot
executebitoff
xlat(oemvs311)
nofileextmap
sidefile(OS390NFS.SAPRED.NFS.MAPPING)
security(saf,exports,saf)
pcnfsd
leadswitch
mintimeout(1)
nomaxtimeout
logout(604800) # 60 * 60 * 24 * 7
nfstasks(8,16,8)
restimeout(48,0)
cachewindow(112)
hfs(/hfs)
logicalcache(16M)
bufhigh(32M)
percentsteal(20)
readaheadmax(16K)
maxrdfsleft(32)
smf(none)
sfmax(20)
nochecklist
fn_delimiter(,)
```

Mount commands

We have followed some of our customers by using the automount daemon which mounts NFS filesystems "On-Demand".

The automounted daemon is controlled via `/etc/init.d/autofs`, and can be started, stopped, or restarted via the service command.

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`

Here is our auto.master config file:

```
/sapmnt/HA1 auto.ha1.sapmnt
```

Here is the referenced file for sapmnt auto.ha1.sapmnt:

File system setup

```
exe -rw,hard sapnfsv:/hfs/sapmnt/HA1/linux_s390x/exe
profile -rw,hard sapnfsv:/hfs/sapmnt/HA1/profile,TEXT,cln_ccsid(819),srv_ccsid(1047)
global -rw,hard sapnfsv:/hfs/sapmnt/HA1/global,TEXT,cln_ccsid(819),srv_ccsid(1047)
trans -rw,hard sapnfsv:/hfs/sapmnt/HA1/trans,TEXT,cln_ccsid(819),srv_ccsid(1047)
```

Appendix C. Sample REXX sanity-check procedure

This appendix contains the sample REXX exec SANCHK.

Sample REXX procedure

SANCHK

This REXX procedure can be used to display and to clear EXCLUDEs and AVOIDs from Move Groups.

REXX sanity-check procedure

```

/* REXX SANCHK ----- */00001102
/* */00001202
/* FUNCTION : Display or CLEAR EXCLUDEs or AVOIDs from MOVE "groups */00001302
/* */00001402
/* */00001502
/* +--- DISPLAY -----+ */00001602
/* SYNTAX : sanchk -----+-----+ */00001702
/* +--- CLEAR -----+ */00001802
/* */00001902
/* -----+-----+ */00002502
Trace 0
00020000
00030000
Arg action .
00040000
00050000
/* Action is either CLEAR or DISPLAY */
00060000
If action = ' ' Then action = 'DISPLAY'
00070000
00080000
/* Issue processing message ... */
00090000
Address NetVAsis ,
00100000
"PIPE LIT /Gathering data step 1 .../" ,
00110000
"| CONS ONLY"
00120000
00130000

/* Find all groups via INGLIST */
00140000
"PIPE (STAGESEP | NAME INGLIST)" ,
00150000
"NETV INGLIST */APG,OUTMODE=LINE" , /* issue command */00160000
"| DROP FIRST 3 LINES" , /* remove header */00170000
"| DROP LAST 1 LINE" , /* remove trailer */00180000
"| SEPARATE" , /* split into single msgs */00190000
"| LOC 19.8 / /" , /* only sysplex groups */00200000
"| EDIT WORD 1.1 1 /\ N WORD 2.1 N" , /* create real name */00210000
"| STEM groups." /* set stem */00220000
00230000
/* Issue processing message ... */
00240000
Address NetVAsis ,
00250000
"PIPE LIT /Gathering data step 2 .../" ,
00260000
"| CONS ONLY"
00270000
00280000
cnt = 0
00290000
errcnt = 0
00300000
00310000
Do i = 1 to groups.0
00320000
group = groups.i
00330000
00340000
/* Get the group details via INGGROUP */
00350000
"PIPE (STAGESEP | NAME INGGROUP)" ,
00360000
"NETV INGGROUP "||group||",ACTION=MEMBERS,OUTMODE=LINE" ,
00370000
"| DROP FIRST 3 LINES" , /* remove header */00380000
"| TAKE FIRST 2 LINES" , /* get data */00390000
"| SEPARATE" , /* split into single msgs */00400000
"| EDIT WORD 3.* 1" , /* get system names */00410000
"| VAR excl avoid" /* set variable */00420000
00430000
If symbol('excl') = 'LIT' Then excl = ' '
00440000
If symbol('avoid') = 'LIT' Then avoid = ' '
00450000
00460000
If excl = ' ' & avoid = ' ' Then Iterate i
00470000
00480000
00490000
errcnt = errcnt + 1
00500000

```

```

errgroup.errcnt = group                                00510000
errdata.errcnt = strip(excl avoid)                   00520000
                                                       00530000
cnt = cnt + 1                                         00540000
outline.cnt = '-----'                             00550000
cnt = cnt + 1                                         00560000
outline.cnt = 'Group      = '||group                 00570000
cnt = cnt + 1                                         00580000
outline.cnt = ' Excluded = '||excl                  00590000
cnt = cnt + 1                                         00600000
outline.cnt = ' Avoided  = '||avoid                 00610000
                                                       00620000
End i                                                  00630000
                                                       00640000
If cnt = 0 Then Do                                    00650000
  If action = 'CLEAR' Then act = 'clear'             00660000
  Else act = 'display'                               00670000
  cnt = cnt + 1                                       00680000
  outline.cnt = 'Nothing to '||act||' ...'           00690000
End                                                    00700000
Else Do                                               00710000
  cnt = cnt + 1                                       00720000
  outline.cnt = '-----'                             00730000
  cnt = cnt + 1                                       00740000
  outline.cnt = 'End of Sanity Check'                00750000
End                                                    00760000
                                                       00770000
outline.0 = cnt                                       00780000
errgroup.0 = errcnt                                   00790000
errdata.0 = errcnt                                   00800000
                                                       00810000
Select                                                00820000
  When action = 'DISPLAY' Then Do                    00830000
    "PIPE (STAGESEP | NAME DISPLAY)" ,               00840000
    "STEM outline. COLLECT" ,                       00850000
    "| COLOR YELLOW" ,                               00860000
    "| CONS ONLY" ,                                  00870000
  End                                                 00880000
  When action = 'CLEAR' & errcnt = 0 Then Do         00890000
    "PIPE (STAGESEP | NAME DISPLAY)" ,               00900000
    "STEM outline. COLLECT" ,                       00910000
    "| COLOR YELLOW" ,                               00920000
    "| CONS ONLY" ,                                  00930000
  End                                                 00940000
  When action = 'CLEAR' Then Do                      00950000
    /* Issue processing message ... */               00960000
    Address NetVAsis ,                               00970000
    "PIPE LIT /Processing CLEAR .../" ,              00980000
    "| COLOR RED" ,                                  00990000
    "| CONS ONLY" ,                                  01000000
    01010000
    Do i = 1 to errgroup.0                            01020000
      /* Issue processing message ... */             01030000
      Address NetVAsis ,                              01040000
      "PIPE LIT \Processing CLEAR for "||errgroup.i||"\\" , 01050000
      "| COLOR RED" ,                                01060000
      "| CONS ONLY" ,                                01070000
      01080000
      "PIPE (STAGESEP | NAME INGGROUP)" ,            01090000
      "NETV INGGROUP "||errgroup.i||",ACTION=INCLUDE,"|| , 01100000

```

REXX sanity-check procedure

```
"SYSTEMS=(||errdata.i||)", "||" , 01110000
"OUTMODE=LINE" , 01120000
"| CONS ONLY" 01130000
End i 01140000
01150000
/* Issue processing message ... */ 01160000
Address NetVAsis , 01170000
"PIPE LIT /Finished CLEAR processing/" , 01180000
" | COLOR RED" , 01190000
" | CONS ONLY" 01200000
01210000
End 01220000
Otherwise Nop 01230000
End 01240000
01250000
Exit 01260000
```

Appendix D. Description of the z/OS high availability scripts

This appendix lists all scripts we used in our scenario. The scripts are invoked by SA z/OS.

Script availability

The scripts and related files described in this appendix were originally made available as part of the additional material associated with IBM Redbook *SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution Using System Automation*.

Since the Redbook was published, updated versions of the scripts have been provided for download via the SAP on DB2 Web site:

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

Select the "Downloads" function to obtain the current scripts.

Create a subdirectory (folder) on your workstation, and unzip the contents of the zip file into this folder.

The following files are provided within **SAP_v8.zip** at the time of this writing:

Table 36. Scripts for SA z/OS

Filename	Description
readme_v8	Supplementary and explanatory information for this version.
SA_AutomationTableAddition	Sample Automation Table entries for 'remote' SAP Application servers
sanchkv1.txt	Sample REXX program to check for and clear Move Group EXCLUDEs or AVOIDs.
saprfc.ini	Sample RFC definition file.
checkappsrv_v4	Sample script used to start the SAP monitor for remote application servers.
startappsrv_v5	Sample shell script used to start a remote application server instance.
sapctrl_em	Sample shell script used to start the: <ul style="list-style-type: none">• components of the ABAP SAP Central Services (ASCS).• Enqueue Replication server belonging to the ASCS.• components of the Java SAP Central Services (SCS).• Enqueue Replication server belonging to the Java SCS.
stopappsrv_v5	Sample shell script used to stop a remote application server instance (ABAP only, double-stack and Java only).
remote_checkdb_win01_90.bat	Sample Windows batch file to run the R3trans SAP executable. This file is executed by SSH to check the database availability on a remote SAP application server running under Windows.
startjappsrv	Sample script to start a remote Java application server
checkjappsrv	Sample script used to start the Java GetWebPage monitor for remote Java application servers.
StartABAP_EnqueueReplicationServer	Sample script to start ABAP enqueue replication server manually
StartJ2EE_EnqueueReplicationServer	Sample script to start Java enqueue replication server manually

z/OS high availability scripts

Table 36. Scripts for SA z/OS (continued)

Filename	Description
SAP-SAAM-zOS-08.xml	Sample end-to-end policy for the SAP HA1 system.

Script descriptions

All scripts (*except* the *sapctrl_em* script) are sample scripts which *must* be adapted to your SAP environment. The call syntax and the required tailoring are described below.

startappsrv_v5

This script is used to *start* a remote ABAP application server instance. It takes the host name, the instance number, the instance directory of the application server, and optionally the remote execution type, as parameters:

```
startappsrv_v5 <hostname> <instnr> <instancedir> [<via>]
```

<hostname>

is the name of the host where the SAP application server runs.

<instnr>

is the instance number of the SAP application server.

<instancedir>

is the instance directory of the remote application server. We run ASCS *and* an application server instance on the host. Therefore, this parameter is required. For example, if the application server uses <instnr> 66 and is a dialog instance only, then the instance directory is normally named D66.

<via> is an optional parameter. It identifies the remote execution type (REXEC or SSH) used to send commands to remote application servers (running under AIX, Linux on System z, or Windows). If a remote application server is started or stopped, then the default type is REXEC. However, if the remote application is controlled via SSH we highly recommend that you set type to SSH.

The line starting with `rfcping=` has to be edited to reflect the full path of the `rfcping` utility.

If remote execution is used, it must be set up to run without password prompt.

Note: To use SSH with Windows, the script must be adapted to use the .bat command files described below:

```
remote_startsap_<hostname>_<instance number>.bat
```

and

```
remote_checkdb_<hostname>_<instance number>.bat
```

To control a remote ABAP Windows application server via SA z/OS, we implemented an SSH based solution. See

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

Because `ssh` allows the execution of only one command on the remote Windows application server, we had to create 3 batch files with the following naming convention:

- `remote_startsap_<hostname>_<instance number>.bat`
- `remote_stopsap_<hostname>_<instance number>.bat`

- remote_checkdb_<hostname>_<instance number>.bat

The batch files contain the sequence of commands and/or calls to real SAP executables. We created directory `c:\sap\rcontrol`. This directory contains the above mentioned batch files. In order to be executable via ssh, the `c:\sap\rcontrol` directory must be added to the PATH of the <sapsid>adm user, here 'redadm'.

With ssh, we do not have a user specific environment. Therefore, in `remote_checkdb_<hostname>_<instance number>.bat`, we needed to add all SAP environment variables needed to run 'R3trans -d'.

Furthermore, the right to "log on as a service" needs to be granted to the user ID 'redadm'.

Command files:

- remote_startsap_win01_90.bat:

```
d:\usr\sap\RED\sys\exe\run\stopsap.exe name=RED nr=90 SAPDIAHOST=win01
d:\usr\sap\RED\sys\exe\run\startsap.exe name=RED nr=90 SAPDIAHOST=win01
```

- remote_stopsap_win01_90.bat:

```
d:\usr\sap\RED\sys\exe\run\stopsap.exe name=RED nr=90 SAPDIAHOST=win01
```

- remote_checkdb_win01_90.bat:

```
@echo off
rem -----
rem SAP env variables needed to run R3trans -d
set DBMS_TYPE=DB2
set SAPSYSTEMNAME=RED
set SAPDBHOST=wtsc42a
set SAPGLOBALHOST=sapred

set DIR_LIBRARY=d:\usr\sap\RED\sys\exe\run
rem Only needed if also set in Instance profile and env. of <sapsid>adm.
rem set R3_DB2_SSID=RED

rem Only needed if trusted connections are used.
d:\usr\sap\RED\sys\exe\run\R3trans -d
exit errorlevel
```

stopappsrv_v5

This script is used to *stop* a remote application server instance, an ABAP only, a double 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:

```
stopappsrv_v5 <hostname> <instnr> <instancedir> [<via>]
```

<hostname>

is the name of the host where the SAP application server runs.

<instnr>

is the instance number of the SAP application server.

<instancedir>

is the instance directory of the remote application server.

<via> is an optional parameter. It identifies the remote execution type (REXEC or SSH) used to send commands to remote application servers (running under AIX, Linux on System z, or Windows). If a remote application server is

z/OS high availability scripts

started or stopped, then the default type is REXEC. However, if the remote application is controlled via SSH we highly recommend that you set type to SSH.

If remote execution is used, it must be set up to run without password prompt.

Note: To use SSH with Windows, the script must be adapted to use the .bat command file described above:

```
remote_stopsap_<hostname>_<instance number>.bat
```

checkappsrv_v4

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.

The line starting with rfcping= has to be edited to reflect the full path of the rfcping utility. In addition, the *cd* command has to be adapted.

```
checkappsrv_v4 <hostname> <instnr>
```

<hostname>

is the name of the host where the SAP application server runs.

<instnr>

is the instance number of the SAP application server.

sapctrl_em

This script is used to manage the ABAP or Java SCS and their corresponding Enqueue Replication Servers. It can be used to start, stop, and monitor the resources of the ABAP and Java SCS instances.

The syntax for invoking sapctrl_em is:

```
./sapctrl_em <resource-prefix> <sapsid> <instance-dir> <host name>  
<exeU flag> <resource-identifier> <action>
```

Parameters:

<resource-prefix>

Prefix of the SCS resource names. If used *under z/OS USS*, this parameter is simply a dummy parameter. For example, use the abbreviation of your SAP solution (ECC / CRM).

<sapsid>

SAPSID to which the SCS belongs.

<instance-dir>

Instance directory of the SCS.

<host name>

Virtual host name of the SCS. The host name is used to identify the appropriate SAP profile for the instance. If the profile has no host name appended, you must use 'no_enq_host' as the value of <host name>.

<exeU flag>

For SAP 7.0x this must be set to 0.

<resource identifier>

The resource of SCS to be addressed (MS|ES|ERS|CO|SE|GW).

<action>

Action to be performed for the resource (START|STOP|CHECK).

startjappsrv

This script is used to start a Java application server instance. This is the syntax of the startjappsrv script:

```
startjappsrv <hostname> <instnr> <instance-dir> <InstType> [ <via> ]
```


<hostname>

is the name of the host where the Java application server instance runs.

<instnr>

is the instance number of the Java application server instance.

<instance-dir>

is the instance directory of the Java application server instance.

<InstType>

identifies if the instance is an Java only instance, or if it is part of a double stack instance. It must be '1' for double stack AS, or '2' for a Java-only application server instance.

<via> is an optional parameter. It identifies the remote execution type (REXEC or SSH) used to send commands to remote application servers (running under AIX, Linux on System z, or Windows). If a remote application server is started or stopped, then the default type is REXEC. However, if the remote application is controlled via SSH we highly recommend that you set type to SSH.

You must also adapt these two lines to your environment:

cd /u/ha1adm

Adapt to your home directory of <sapsid>adm.

Max_HC_Retries=24

Define the number of retries for the health checker (HC) call java GetWebPage. Because the script sleeps *10 seconds* between each HC call, it gives up *after 240 seconds* and then returns a STARTUP FAILED message. If starting the Java AppServer takes longer in your environment, you should adapt this value.

checkjappsrv

This script is used to start a Java application server monitor (GetWebPage). This is the syntax of the checkjappsrv script:

```
checkjappsrv <hostname> <instnr>
```

<hostname>

is the name of the host where the Java application server monitor runs.

<instnr>

is the instance number of the Java application server monitor.

You must also adapt this line to your environment:

cd /u/ha1adm

Adapt to your home directory of <sapsid>adm.

z/OS high availability scripts

Appendix E. Sample Tivoli System Automation for Multiplatforms high availability policy for SAP

This appendix explains why the SA MP HA policy for SAP (version 5.3) described in this document was defined as it is. It gives some basic background information to help you to understand what is happening and why. It is expected that all resources be created through the use of the supplied mksap script using the supplied configuration files as input.

Refer to Chapter 12, “Customizing the Tivoli System Automation for Multiplatforms (Base),” on page 195 for a general discussion on implementing this policy.

The ABAP SAP Central Services group (ASCS)

The ASCS group (for example SAPECC_ABAP_HA1_ENQ) contains *six* floating resources. The *five* SAP resources have a DependsOn relationship to the IP resource. On starting this group, these relationships cause the IP resource to be started first, and when it is online, the other components to be started in parallel. In addition, the DependsOn relationship has two more effects:

- All resources are started on the same node, but this would have happened anyway due to the collocated member location of the group.
- On failure of the IP resource, all other resources in this group are restarted.

Three resources are *mandatory group members* (IP, ES, and MS). The others (GW, CO, and SE) are *non-mandatory group members*, and are optional.

This is done to make sure the most critical applications are always running, and these are the ES, MS, and their IP addresses.

- If the ES or IP fails, no restart ‘in-place’ is attempted. Instead, the failover of the whole group is triggered.
- If the MS fails, a restart in-place is first attempted, and only if this restart fails does SA MP cause the whole group to be moved. Therefore, only the MS, ES and IP can trigger a failover.
- If one of the three other optional resources (GW, CO, or SE) fails and cannot be restarted, this does not trigger a group failover, and the resource stays down.

Note: Since the 6.40 kernel, the ASCS can run in two different character encoding schemes: ASCII and Unicode. The default is Unicode. If your SAP solution requires running the ASCS in ASCII mode, then you need to indicate this to the ‘mksap’ script when generating the SA MP resources.

In case you run the NFS server for SAP in the same SA MP domain as the SAP system, you must manually add ‘StartAfter’ relationships between the ASCS and the NFS server.

The ABAP ENQREP group for ASCS

The ABAP ENQREP group (SAPECC_ABAP_HA1_ENQREP) contains one mandatory floating resource, the ERS. The relationships of this group and its member to the ASCS group are described in “Interaction between ES and ERS” on page 328.

If the ASCS is run in ASCII mode, the ERS is also automatically created to run in ASCII mode.

The Java SAP Central Services group

The Java SCS group, for example SAPECC_J2EE_HA1_ENQ, contains three floating resources. The two SAP resources have a ‘DependsOn’ relationship to the IP resource. All three resources are mandatory group members: IP, ES, and MS. This means that ES, MS and IP can trigger a failover.

- If the ES or IP fails, no restart ‘in-place’ is attempted. Instead, the failover of the whole group is triggered.
- If the MS fails, a restart in-place is first attempted, and only if this restart fails does SA MP cause the whole group to be moved.

The Java SCS can run only in the Unicode character encoding scheme.

In case you run the NFS server for SAP in the same SA MP domain as the SAP system, you must manually add ‘StartAfter’ relationships between the Java SCS and the NFS server.

The ENQREP group for the Java SCS

The Java ENQREP group (SAPECC_J2EE_HA1_ENQREP) contains one mandatory floating resource, the ERS.

The relationships of this group and its members to the Java SCS group are described in “Interaction between ES and ERS” on page 328.

The application server groups

Each of the application server groups contains one fixed application server (AS) resource as a mandatory member. The application servers are in separate groups, because they should not affect each other in any way. All application server resources have ‘StartAfter’ relationships to their ES and MS.

These two relationships are established for the following reasons:

1. At the startup of an application server, the MS must be online, because the AS reads the license key from the MS. Otherwise, a logon to the AS is not possible, and monitoring of the AS will fail.
2. Without an ES online, AS startup would succeed, but no tasks could be processed in the AS.

There is no reason to restart an AS in case of a failure on ES or MS, because the AS reconnects to the MS automatically. Therefore, a StartAfter relationship is sufficient.

The sample HA policy for SAP (version 5.3) supports ABAP-only, Java-only and *double-stack* application servers.

- Although a double-stack application server is physically one instance, we separated it into its two logical parts, the ABAP application server and the Java application server.
- In other words, within an SA MP domain, one double-stack application server instance is automated as two logical application server instances, an ABAP application server instance and a Java application server instance. However, there is a close relationship between these two logical application servers:
 - The Java instance is always started after the ABAP instance. This 'StartAfter' relationship guarantees that starting the Java instance automatically triggers the prior start of the ABAP instance.
 - On the other hand, stopping the Java application server does not stop any of the double-stack server processes. It only stops the monitoring Java program 'GetWebPage', which does a primitive health check of Java application servers as the rfcping program does for ABAP application servers.

Notes:

1. The System z SAP solution is normally a heterogeneous environment, which means that the database server always runs under z/OS and the application servers run on a different operating system (such as Linux on System z) or even on a different hardware platform and operating system, such as System p and AIX.
2. For our System z SAP solution, we do not have a 'StartAfter' relationship between the application server and the database server. For an explanation of this, see "Application groups" in Chapter 11, "Customizing Tivoli System Automation for z/OS," on page 171.
3. If you use the sample HA policy for SAP (version 5.3) for an SAP solution running in a homogeneous environment, you should add a 'StartAfter' relation between each application server resource and the database server resource group.

The SAProuter / SAP Web Dispatcher group

The *SAProuter* program is an SAP utility controlling access to SAP ABAP systems. The SAProuter group (SAPECC_ABAP_SYS_ROUTER) contains two floating resources:

- The SAProuter (SAPROUTER),
- A service IP address (IP).

There is a DependsOn relationship defined from SAPROUTER to IP. On startup of the group, this causes the IP address to be started first. After the IP address is operational, the SAProuter starts. The SAP Web Dispatcher program is an SAP utility controlling Web access for SAP ABAP and Java systems.

The SAP Web Dispatcher group (SAPECC_ABAP_SYS_WEBDISP) also contains two floating resources:

- The SAP Web Dispatcher (SAPWEBDISP),
- A service IP address (IP).

There is a 'DependsOn' relationship defined from SAPWEBDISP to IP. On startup of the group, this causes the IP address to be started first. After the IP address is operational, the SAP Web Dispatcher starts

Interaction between ES and ERS

The most complex relationships are defined between the ES and the ERS. In the following, only the ES and the ERS are considered. Of course, a failover of the ES always causes all the other resources in the ES group to be moved. This description applies to ABAP SCS (ASCS) and Java SCS instances with *enqueue server replication enabled*.

1. Assume that currently everything is offline. Now the ES is started.
 - At ES startup, only one relationship must be honored. This is the 'Collocated/IfNotOffline' relationship. Of course, the 'DependsOn' relationship to the IP resource must be taken into consideration, too.
 - As noted, however, only ES and ERS are considered here. The relationship has a condition of 'IfNotOffline'. Currently, the ERS is offline, so the relationship is discarded.
 - This means that the ES can be started anywhere. SA MP will try to start the resources on the first node in the NodeNameList of the resources. If that is not possible, it will try the second node and so on.
2. Now the ERS is started. There are three relationships from the ERS to the ES that lead to the following behavior:

ERS AntiCollocated ES

The ERS is always started on a *different node* than the ES.

ERS StartAfter ES

This relationship makes sure that the ERS is started *after* the ES has become online.

ERS IsStartable ES

This relationship makes sure that the ERS is only started on a node *where the ES could potentially be started*. It makes no sense to start the ERS on a node where, for example, the ES cannot run.

Both the ES and the ERS are now online on different nodes.

In a failure situation where the ES terminates (or the node it is running on crashes), the scenario is different from the one previously described at the 'normal' startup of ES. The ES should be online and is obviously not.

At ES restart, the same thing happens as above. SA MP examines the relationships. The condition of the Collocated/IfNotOffline relationship matches, now that the ERS is online (not offline). This causes a start of the ES on the node where the ERS is already running. After the ES replicates the data, the ERS terminates by itself.

SA MP restarts the ERS on another node due to the AntiCollocated relationship from ERS to ES. Now, both resources, the ES and the ERS, are running on different nodes again.

Application server (AS) resources

We do not recommend moving AS instances from one node to another in the cluster as this means an "unacceptable downtime".

- The SAP architecture allows to run more than one AS. Run at least 2 AS on different hardware to get the necessary AS redundancy.
- AS resources are fixed resources, whereas utilities like SAProuter are floating resources and need their own virtual hostname which will move with those resources.

SA MP: High availability policy for SAP

- NFS server and the SAP Central Services are run on z/OS and USS, outside of the SA MP cluster.

ABAP-only SAP systems and Java-only SAP systems are very similar from the SA MP viewpoint. Each ABAP or Java Application server is modeled as a SA MP resource/group. This is different for SAP systems that support ABAP and Java stacks simultaneously. Such a so-called 'double-stack' system has Application servers, which runs both the ABAP and the Java stack. But such an AS is physically one instance.

Starting an AS means starting the ABAP and the Java stack. Although it is physically one instance, SA MP separates a *double-stack* AS into its two logical parts:

- the ABAP application server,
- the Java application server.

In other words, within an SA MP domain, one 'double-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 is always started after the ABAP instance. This StartAfter relationship guarantees that starting the Java instance automatically triggers the prior start of the ABAP instance.
- On the other hand, the SA MP implementation makes sure, that stopping the Java application server does not stop any of the 'double-stack' server processes but rather only the monitoring Java program 'java GetWebPage', which does a primitive health check of Java application servers. The HA policy for SAP (version 5.3) creates this 'StartAfter' relationship automatically if the corresponding ABAP application server resource has previously been created.

We want SA MP to call our monitoring script every 300 seconds and give it 10 seconds to return the status. For Java application servers, we explicitly set the start and stop command timeout to 200 seconds, because it can take that long to start or stop it.

- When SA MP starts a resource, it monitors the resource to find out when it goes 'online'.
- If the resource has been noted as 'offline' a certain number of times (3 by default), SA MP sets the resource to 'failed'. So in the case of the AS we have to live with the long monitoring interval.
- However, it is not acceptable to wait up to 5 minutes to find that an application server has died. Therefore, we need a different method of signaling the failure to Tivoli System Automation.

For an ABAP application server, we spawn a monitoring process (rfcping) from the start script after we start the AS. We do not now monitor the main process of the application server (disp+work) but rather the separate rfcping process. This process tests the health of the AS and stays alive as long as the AS and the connection to the database are in a healthy state.

- When rfcping dies, we invoke the action routine refreshOpState, which triggers monitoring immediately.
- If this monitor run finds the AS offline, a restart is initiated immediately.

SA MP: High availability policy for SAP

For a Java application server, we spawn a Java monitoring process (GetWebPage) from the start script after we start the AS. We do not now monitor the main process of the application server (jcontrol) but rather the separate Java GetWebPage process. This process tests the health of the AS and stays alive as long as the AS and the connection to the database are in a healthy state.

- When GetWebPage dies, we invoke the action routine refreshOpState, which triggers monitoring immediately.
- If this monitor run finds the AS offline, a restart is initiated immediately.

GetWebPage

For the monitoring case (CHECK action), it is important to handle the situation in which an application server hangs while its operating system processes are still active. Therefore, it is not enough to do just pid monitoring. We need a health checker.

- For ABAP application servers, SAP offers a program called *rfcping*, which does a dummy logon to the application server, sends a request to the database server, and waits for its response.
- You can decide whether rfcping does this a given number of times (for example, once) and supplies a return code or whether rfcping should do this in an infinite loop. In this case you can tell that, whenever the infinite rfcping stops, there might be a problem with the application server or the network connection to it.

For Java application servers, SAP currently does not offer such a simple health checker program. Therefore, we had to write one ourselves. This Java program is named *GetWebPage*. Its operation is very similar to the ABAP rfcping program:

1. It tries to connect to a Web page, in our case the index.html file of a Java application server.
2. If this page is reachable, it means that the Java application server is operational.
3. If not, it indicates that the Java application server is not working.

The GetWebPage program can be called to do the connection test to the Web page only once, supplying a return code, or infinitely at specifiable intervals. As with rfcping, you can tell that, whenever the infinite GetWebPage stops, there might be a problem with the Java application server or the network connection to it.

The program is called as follows:

```
java GetWebPage <resource identifier> <URL> <log-file> <interval>
```

where:

<resource-identifier>

is the prefix of the SCS resource names.

<URL>

is the URL to be queried (containing file index.html)

<log file name>

Log file (in HTML format). Contains the returned contents of the URL target file (index.html) if the connection is successful and <interval> is specified as '0'.

<interval>

If 0, the connection is tested once and the program delivers a return code

plus the contents of the URL target file if successful. Otherwise, the number specifies the interval (in seconds) at which a connection test is done within an infinite loop.

Sample invocation for GetWebPage:

```
java GetWebPage J2EE_AS00_on_siccps07 http://siccps07:50000/index.html
siccps07_Http_50000.html 30
```

Service IP addresses (or VIPAs)

The service IPs are logically mapped to resources of the class *IBM.ServiceIP*. They are related to a certain "service" and they must "move with it" when required.

Currently we have up to four service IPs:

- one for ABAP
- one for Java SCS
- one for SAPROUTER
- one for SAPWEBDISP.

To achieve the collocation, we put the service IPs into the corresponding groups, where we used member location=collocated.

In our sample test environment, we did not use dynamic routing for SAP Central Services, and so on. Therefore, in order to guarantee accessibility, all IP addresses must be in the *same subnet* as the interface to which they are defined as IP aliases.

From an HA point of view, it is better to have dynamic routing implemented and use addresses of different subnets. The sample policy contains the basis for this by creating a network equivalency between the service IP address and the primary IP address of the interface to which the service IP is aliased.

An even more "highly available" setup would be to use VIPAs instead of service IPs

Notes:

1. VIPA support is possible with Tivoli System Automation for Multiplatforms. However, it is not within the scope of the current version of the SAP HA sample policy to support virtual IP addresses (VIPAs). As described above, IP aliasing is used instead.
2. IP addresses are *critical in terms of SA MP*. Therefore, the default of attribute ProtectionMode=1 for the class *IBM.Service* is acceptable.
3. You require the *softdog kernel module* when using critical resources.

Setup scripts

We deliver the following scripts and executable configuration files, which can be used to set up, monitor and clean up the sample policy:

Table 37. Setup scripts for the SA MP high availability policy for SAP

Filename	Description
mksap	Creates the sample policy
rmsap	Removes the sample policy.
lssap	Reports the status of the groups and resources of the sample policy

SA MP: High availability policy for SAP

Table 37. Setup scripts for the SA MP high availability policy for SAP (continued)

Filename	Description
ABAP_instances.conf	Executable configuration file used by the above scripts to create ABAP resources for <ul style="list-style-type: none">• ABAP SAP Central Services• ABAP application server(s) and SAP system independent resources <ul style="list-style-type: none">• SAProuter• SAP Web Dispatcher
J2EE_instances.conf	Executable configuration file used by the above scripts to create Java resources for <ul style="list-style-type: none">• Java SAP Central Services• Java application server(s)

Important: All scripts must reside in the same directory.

Automation scripts

We deliver the following scripts that are used for starting, monitoring and stopping the resources of the sample policy:

Table 38. Delivered SA MP automation scripts

File	Description
sapctrl_pid	Monitors and stops a general operating system process. This script is used by the following scripts.
sapctrl_em	Manages the resources of ASCS (ABAP SCS) and SCS (Java SCS)
sapctrl_as	Manages the application server resources, which can be ABAP-only, Java-only, or double-stack application servers
sapctrl_sys	Manages the SAPSID-independent resources (currently SAPROUTER and/or SAPWEBDISP)

Notes:

1. All scripts *must reside in the same directory and must be executable by the <sapsid>adm.*
2. You can adjust the amount of information written to the syslog by specifying a different value for the variable DEBUG inside the script.

Removing the policy (rmsap)

If rmsap is called without any arguments, it performs the following actions to clean up the policy (the default configuration file is saphalinux.conf):

1. Call the configuration file (default: saphalinux.conf) to get the configuration information
2. First, ask the obligatory "Are you sure?" for <PREFIX>_<SAPSID>_% resources and wait for a response
3. If the answer is "yes", rmsap:
 - removes all relationships whose names start with the prefix specified by <PREFIX>_<SAPSID>

SA MP: High availability policy for SAP

- removes all resources of class IBM.ServiceIP whose names start with the prefix specified by <PREF>_<SAPSID>
 - removes all resources of class IBM.Application whose names start with the prefix specified by <PREF>_<SAPSID>
 - removes all groups whose names start with the prefix specified in <PREF>_<SAPSID>
 - removes all equivalencies between IP addresses and network interfaces whose names start with the prefix specified by <PREF>_<SAPSID>
4. Again, ask the obligatory "Are you sure?" for <PREF>_SYS_% resources and wait for a response.
 5. If the answer is "yes", rmsap:
 - removes all relationships whose names start with the prefix specified by <PREF>_<SYS>
 - removes all resources of class IBM.ServiceIP whose names start with the prefix specified by <PREF>_<SYS>
 - removes all resources of class IBM.Application whose names start with the prefix specified by <PREF>_<SYS>
 - removes all groups whose names start with the prefix specified in <PREF>_<SYS>
 - removes all equivalencies between IP addresses and network interfaces whose names start with the prefix specified by <PREF>_<SYS>

If you want to remove resources with different prefixes, you have to use different configuration files.

Note: All resources must be offline before you can remove them.

SA MP: High availability policy for SAP

List of abbreviations

The abbreviations used in this document are listed below. For more detailed explanations; refer to the glossary.

ABAP/4	Advanced Business Application Programming 4th Generation Language (SAP)
abend	Abnormal end of task
ACL	Access Control List
ADP	Automatic Data Processing
AIX	Advanced Interactive Executive (IBM implementation of UNIX)
ANSI	American National Standards Institute
APAR	Authorized Program Analysis Report
APF	Authorized Program Facility
API	Application Program Interface
APO	Advanced Planning and Optimization (SAP) (now known as SCM)
APPC	Advanced Program-to-Program Communication
AR	Address Register, Access Register
ARM	Automatic Restart Management
ARP	Address Resolution Protocol
AS	Application Server
ASCH	APPC/MVS Scheduler
ASCS	ABAP SAP Central Services (abbreviation used by SAP)
BAPI	Business Application Program Interface
BC	Basic Component
BSDS	Bootstrap Data Set
BSY	Busy
BTC	Batch (SAP process type)
BW	Business Information Warehouse
CAS	Catalog Address Space
CBU	Capacity Backup Upgrade
CCMS	Computing Center Management System
CCSID	Coded Character Set Identifier
CEC	Central Electronics Complex
CEE	Common Execution Environment
CF	Coupling Facility
CFRM	Coupling Facility Resource Manager
CI	Correlation ID (WLM work qualifier), Central Instance, Control Interval
CICS	Customer Information Control System
CIFS	Common Internet File System
CINET	Common AF_INET
CLI	Command Line Interface
CP	Control program
CPC	Central Processor Complex
CPU	Central Processing Unit
CRCR	Conditional Restart Control Record
CRLF	Carriage Return Line Feed
CRM	Customer Relationship Management
CSM	Communications Storage Manager
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
CSS	Customer Support System (SAP)
DASD	Direct Access Storage Device
DB	Database

Abbreviations

DB2	Database 2 (an IBM relational database management system)
DBA	Database Administrator
DBD	Data Base Descriptor
DBET	Database Exception Table
DBIF	Database Interface (SAP component)
DBMS	Database Management System
DBRM	Database Request Module
DBSL	Database Service Layer (functional interface within DBIF)
DDF	Distributed Data Facility
DDL	Data Description Language
DEC	Digital Equipment Corporation
DI	Dialog instance
DFS	Distributed File Service
DFSMS	Data Facility Storage Management Subsystem
DIA	Dialog (SAP process type)
DLL	Dynamic Linked Library
DNS	Domain Name Server
DPSI	Data-Partitioned Secondary Index
DRDA	Distributed Relational Database Architecture
DSN	Data Set Name
DSP	Dispatcher (SAP process type)
EBCDIC	Extended Binary Coded Decimal Interchange Code
EDM	Environment Descriptor Modules
EJB	Enterprise Java Beans
ENQ	Enqueue (SAP process type)
ERS	Enqueue Replication Server
ES	Enqueue Server
ESCD	ESCON Director
ESCON	Enterprise Systems Connection
ESS	Enterprise Storage Server
FAQ	Frequently Asked Questions
FLA	Fast Log Apply
FRR	Functional Recovery Routine
FTP	File Transfer Protocol
GAN	Group Attachment Name
GB	Gigabytes
GbE	Gigabit Ethernet
GBP	Group Buffer Pool
GEN	Generic (SAP process type)
GID	Group ID
GR	General Register
GRS	Global Resource Serialization
GSA	General Services Administration (U.S.)
GUI	Graphical User Interface
GWY	Gateway (SAP process type)
HA	High Availability
HACMP	High Availability Cluster Multiprocessing
HFS	Hierarchical File System
HLQ	High-Level Qualifier
HMC	Hardware Management Console
HPGRBRBA	High Page Recovery Base Relative Byte Address
HSC	Homogeneous System Copy
HSM	Hierarchical Storage Manager
I/O	Input / Output
IBM	International Business Machines Corporation

ICF	Integrated Catalog Facility
ICLI	Integrated Call Level Interface
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers (USA)
IFI	Instrumentation Facility Interface
IMS	Information Management System
INET	Integrated Network
IP	Internet Protocol
IPL	Initial Program Load
IRLM	Internal Resource Lock Manager
ISO	International Standards Organization
ISPF	Interactive System Productivity Facility
ISV	Independent Software Vendor
IT	Information Technology
ITSO	International Technical Support Organization
J2EE	Java 2 Enterprise Edition
JCL	Job Control Language
JCS	Job Control Statement
JES	Job Entry Subsystem
JES2	Job Entry Subsystem 2
KB	Kilobytes (1024 bytes)
LAN	Local Area Network
LLA	Library Lookaside
LPAR	Logical Partition
LPL	Logical Page List
LRSN	Log Record Sequence Number
LSA	Link State Advertisement
MB	Megabytes
MCOD	Multiple Components in One Database
MIPS	Million Instructions Per Second
MPC	Multi-Path Channel
MPF	Message Processing Facility
MS	Message Server
MSG	Message server (SAP process type)
MTU	Maximum Transmission Unit
MVS	Multiple Virtual Storage (component of z/OS)
MVT	Multiprocessing with a Variable Number of Tasks
NCCF	Network Communications Control Facility
NFS	Network File System
NIC	Network Interface Card
NIS	Network Information System
NLS	National Language Support
NMC	NetView Management Console
NPI	Non-Partitioned Index
OAM	Object Access Method
ODBC	Open Database Connectivity
OMVS	OpenEdition MVS
OPC	Operations Planning and Control
OS	Operating System
OS/390	Operating System/390
OSA	Open Systems Architecture
OSA-E	OSA-Express adapter
OSF	Open Software Foundation
OSS	Online Service System (SAP), now Customer Support System (CSS)
OSPF	Open Shortest Path First

Abbreviations

OTF	Output Text Format
PAM	Product Availability Matrix (SAP)
PC	Process name
PCI	Peripheral Component Interconnect
PDF	Portable Document Format
PDS	Partitioned Data Set
PID	Process ID
PKT	Packet
PMTU	Path MTU
PPRC	Peer-to-Peer Remote Copy
PR/SM	Processor Resource / Systems Manager
PSW	Program Status Word
PTF	Program Temporary Fix
QDIO	Queued Direct I/O
R/3	SAP R/3 System
RACF	Resource Access Control Facility
RAS	Reliability, Availability, Serviceability
RBA	Relative Byte Address
RBLP	Recovery Base Log Point
RC	Return Code
RDBMS	Relational Database Management System
RED	Redbook
REXX	Restructured Extended Executor Language
RFC	Remote Function Call
RISC	Reduced Instruction Set Computer
RMF	Resource Management Facility
RRAS	Routing and Remote Access Services
RRS	Recoverable Resource Services, or Resource Recovery Services
RRSAF	Recoverable Resource Management Services Attachment Facility
RS/6000	IBM RISC System/6000
RSCT	Reliable Scalable Cluster Technology
RTO	Retransmission Timeout
S/390	System/390
SA	System Automation
SA z/OS	Tivoli System Automation for z/OS
SA MP	Tivoli System Automation for Multiplatforms
SAF	System (or Security) Authorization Facility
SAP	Systems, Applications, Products in Data Processing (software vendor), System Assist Processor
SAPCCMSR	SAP CCMS agent for SAPOSCOL
SAPCL	SAP Collector
SAPOSCOL	SAP OS Collector
SAPSID	SAP system ID
SCA	Shared Communication Area
SCM	Supply Chain Management
SCS	SAP Central Services (also used by SAP to denote the Java SCS implementation)
SDF	Status Display Facility
SDSF	Spool Display and Search Facility
SID	System ID
SLA	Service Level Agreement
SLES	SUSE Linux Enterprise Server
SMB	Session Message Block
SMF	System Management Facility
SMIT	System Management Interface Tool
SMP	Symmetric Multiprocessor, System Maintenance Program

SMS	Storage Management Subsystem
SNA	Systems Network Architecture
SNMP	Simple Network Management Protocol
SPAS	Stored Procedure Address Space (DB2)
SPM	Subsystem Parameter (WLM work qualifier)
SPO	Spool (SAP process type)
SPOF	Single Point of Failure
SPUFI	SQL Processor Using File Input
SQL	Structured Query Language
SSH	Secure Shell
STP	Server Time Protocol
SVC	Service, Supervisor Call
SWA	Scheduler Work Area
SYSADM	System Administration Authority
Sysplex	Systems Complex
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TFS	Temporary File System
TNG	Transaction name group
TP	Transport Tool (SAP)
TSO	Time Sharing Option
UACC	Universal Access Authority
UDB	Universal Database
UDP	User Datagram Protocol
UID	User ID
UNIX	An operating system developed at Bell Laboratories
UP2	Update (SAP process type)
UPD	Update (SAP process type)
UR	Unit of Recovery
USS	UNIX System Services
VCAT	Volume Catalog
VIPA	Virtual IP Address
VLAN	Virtual LAN
VLF	Virtual Lookaside Facility
VM	Virtual Machine
VMCF	Virtual Machine Communication Facility
VS	Virtual Storage
VSAM	Virtual Storage Access Method
VSE	Virtual Storage Extended
VSWITCH	Virtual Switch
VTAM	Virtual Telecommunications Access Method
VTOC	Volume Table of Contents
WLM	Workload Manager, Workload Management
WP	Work Process
WWW	World Wide Web
XCF	Cross-System Coupling Facility
XRC	Extended Remote Copy
zFS	z/OS File System
z/OS	System z Operating System
z/VM	System z Virtual Machine

Abbreviations

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). IBM's licensed version of the UNIX operating system. The RISC System/6000 system, among others, runs on the AIX operating system.

application plan. The control structure produced during the bind process and used by DB2 to process SQL statements encountered during statement execution.

authorization ID. A string that can be verified for connection to DB2 and to which a set of privileges are allowed. It can represent an individual, an organizational group, or a function, but DB2 does not determine this representation.

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.

Authorized Program Facility (APF). A z/OS facility that permits identification of programs authorized to use restricted functions.

automatic bind. (more correctly, automatic rebind). A process by which SQL statements are bound automatically (without a user issuing a **BIND** command) when an application process begins execution and the bound application plan or package it requires is not valid.

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. See also 'automatic bind,' 'dynamic bind,' and 'static bind.'

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 tablespaces 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 component providing client access to a remote database within the Distributed Relational Database Architecture (DRDA).

default. An alternative value, attribute, or option that is assumed when none has been specified.

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 (DRDA). A connection protocol for distributed relational database processing that is used by IBM's 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.

dynamic bind. A process by which SQL statements are bound as they are entered.

Enterprise Systems Connection Architecture (ESCON). An architecture for an I/O interface that provides an optical-fiber communication link between channels and control units.

Ethernet. A 10- or 100-megabit baseband local area network that allows multiple stations to access the

Glossary

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).

Fast Ethernet. Fast Ethernet is an Ethernet networking standard capable of data transmission rates as high as 100 Mbps. Fast Ethernet networking requires a network interface card (NIC) capable of transmitting data at 100 Mbps. Fast 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. For example, a gateway converts the protocol for a Token Ring network to the protocol for SNA. 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. (1) Pertaining to a selection, choice, or condition that has 16 possible different values or states. (2) Pertaining to a fixed-radix numeration system, with radix of 16. (3) 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.

Integrated Call Level Interface (ICLI). No longer supported. A component used by the SAP DBIF interface. It consists of client and server components and allows AIX or Windows application servers to access a z/OS database server remotely across a network. The DBIF uses only a subset of data base functions and the ICLI delivers exactly that subset.

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.

Job Control Statement (JCS). A statement in a job that is used in identifying the job or describing its requirements to the operating system.

Job Entry Subsystem (JES). In OS/VS2 MVS, a system facility for spooling, job queuing, and managing the scheduler work area.

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.

Multiple Components in One Database. An SAP term that describes topologies in which more than one SAP system share one 'database'. In DB2 terminology, the SAP term 'database' is equivalent to a DB2 subsystem or a DB2 data sharing group. General information on MCOD is available at <http://service.sap.com/mcod>.

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.

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.

password. In computer security, a string of characters known to the computer system and a user, who must specify it to gain full or limited access to a system and to the data stored within it. In RACF, the password is used to verify the identity of the user.

Path MTU Discovery. A configuration option that requests TCP/IP to dynamically determine the *path MTU*, i.e., the minimum MTU for all hops in the path.

plan name. The name of an application plan.

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 *SAP Database Administration Guide*.

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.

Reduced Instruction Set Computer (RISC). A computer that uses a small, simplified set of frequently used instructions for rapid execution.

Relational Database Management System (RDBMS). A relational database manager that operates consistently across supported IBM systems.

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 (optional)
- Syslog collector (optional)
- Syslog sender (optional)

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 *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 ...
```

SQL Processor Using File Input (SPUFI). A facility of the TSO attachment subcomponent that enables the

Glossary

DB2I user to execute SQL statements without embedding them in an application program.

static bind. A process by which SQL statements are bound after they have been precompiled. All static SQL statements are prepared for execution at the same time. Contrast with dynamic bind.

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.

subsystem. A distinct instance of an RDBMS.

superuser. In OpenEdition MVS, 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 Authorization Facility (SAF). A z/OS component that provides a central point of control for security decisions. It either processes requests directly or works with RACF or another security product to process them.

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.

Systems Network Architecture (SNA). A widely used communications framework developed by IBM to define network functions and establish standards for enabling its different models of computers to exchange and process data. SNA is essentially a design philosophy that separates network communications into five layers.

table. A named data object consisting of a specific number of columns and some number of unordered rows. Synonymous with base table or temporary table.

Time-Sharing Option (TSO). An option of MVT and OS/VS MVS 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.

User Datagram Protocol (UDP). A packet-level protocol built directly on the Internet protocol layer. UDP is used for application-to-application communication between host systems.

Virtual IP Address (VIPA). 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). (1) 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. (2) 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 System z 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 System z.

System z. A range of IBM mainframe processors representing the successors to the S/390.

Glossary

Bibliography

IBM documents

The latest IBM documentation can be found at:

DB2 <http://www.ibm.com/software/data/db2/zos/library.html>

z/OS <http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>

Tivoli System Automation for z/OS

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

Tivoli System Automation for Multiplatforms

<http://www.ibm.com/software/tivoli/products/sys-auto-linux>

Table 39 lists the *IBM Tivoli System Automation for z/OS* publications that you require to implement your SA z/OS policy.

Tivoli System Automation for z/OS

Table 39. Selected IBM Tivoli System Automation for z/OS publications

IBM Documents	Order Number
Tivoli System Automation for z/OS (Release 3.1)	
<i>Tivoli System Automation: Customizing and Programming</i>	SC33-8260-04
<i>Tivoli System Automation: Planning and Installation</i>	SC33-8261-03
<i>Tivoli System Automation: Programmer's Reference</i>	SC33-8266-02
<i>Tivoli System Automation: User's Guide</i>	SC33-8263-03

Table 40 lists the *IBM Tivoli System Automation for Multiplatforms* publications you require to operate your SA MP policy.

Tivoli System Automation for Multiplatforms

Table 40. Selected IBM Tivoli System Automation for Multiplatforms publications

IBM Documents	Order Number
Tivoli System Automation for Multiplatforms (Release 3.1)	
<i>Administrator's and User's Guide</i>	SC33-8415-00
<i>Installation and Configuration Guide</i>	SC33-8416-00
<i>Reference</i>	SC33-8417-00

Table 41 on page 348 lists the IBM order numbers and SAP material numbers for earlier editions of this manual, as well as planning guides for earlier SAP releases.

Bibliography

Table 41. Other IBM reference documents

IBM Documents	Order Number
<i>High Availability for SAP on IBM System z Using Autonomic Computing Technologies</i>	SC33-8206-01
<i>SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th Edition</i>	SC33-7965-03
<i>SAP R/3 on DB2 for OS/390: Planning Guide 2nd Edition; SAP R/3 Release 4.6D</i>	SC33-7966-04
<i>Linux on System z Device Drivers, Features, and Commands (Linux kernel 2.6)</i>	SC33-8289

Table 42 lists the IBM Redbooks that you might find useful.

Table 42. IBM Redbooks and Redpapers covering related topics

IBM Redbooks/Redpapers (published by the IBM International Technical Support Organization, ITSO)	Order Number
<i>IBM System z Strengths and Values</i>	SG24-7333
<i>z/OS V1R8 DFSMS Technical Update</i>	SG24-7435
<i>DB2 9 for z/OS Performance Topics</i>	SG24-7473
<i>Enhancing SAP by Using DB2 9 for z/OS</i>	SG24-7239
<i>Best Practices for SAP BI using DB2 9 for z/OS</i>	SG24-6489
<i>SAP R/3 on DB2 for OS/390: OS/390 Application Server</i>	SG24-5840
<i>SAP R/3 on DB2 for OS/390: Disaster Recovery</i>	SG24-5343
<i>SAP on DB2 for z/OS and OS/390: DB2 System Cloning</i>	SG24-6287
<i>Open Source Software for z/OS and OS/390 UNIX</i>	SG24-5944
<i>Implementing SAP R/3 in an OS/390 Environment Using AIX Application Servers</i>	SG24-4945
<i>SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations [1]</i>	SG24-5690
<i>SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution Using System Automation [1]</i>	SG24-6836
<i>SAP on DB2 for z/OS and OS/390: High Availability and Performance Monitoring with Data Sharing [1]</i>	SG24-6950
<i>SAP on DB2 Universal Database for OS/390 and z/OS: Multiple Components in One Database (MCOB)</i>	SG24-6914
<i>DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There</i>	SG24-7088
<i>SAP on DB2 UDB for OS/390 and z/OS - Implementing Application Servers for Linux on zSeries</i>	SG24-6847
<i>IBM Systems for SAP Business Intelligence: 25 Terabyte Scalability Study</i>	REDP-4411
<i>DB2 9 for z/OS Technical Overview</i>	SG24-7330
<i>DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know , ... and More</i>	SG24-6079
<i>Distributed Functions of DB2 for z/OS and OS/390</i>	SG24-6952
<i>Optimizing Restore and Recovery Solutions with DB2 Recovery Expert for z/OS V2.1</i>	SG24-7606
<i>IBM System Storage DS8000 Architecture and Implementation</i>	SG24-6786
<i>IBM System Storage DS8000: Copy Services with IBM System z</i>	SG24-6787
<i>mySAP Business Suite Managed by IBM Tivoli System Automation for Linux (Redpaper) [1]</i>	REDP-3717
<i>Linux on IBM zSeries and S/390: VSWITCH and VLAN Features of z/VM 4.4 (Redpaper)</i>	REDP-3719
<i>GDPS Family - An Introduction to Concepts and Capabilities</i>	SG24-6374
[1] Information from these publications was updated and used as the basis for the current book.	

SAP documents

The latest SAP documentation can be found at:

SAP High Availability

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

SAP Installation Documentation

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

Table 43. SAP publications

SAP High Availability
<i>BC SAP High Availability</i>
(SAP online documentation is available in the SAP Library or at http://service.sap.com/ha)
SAP NetWeaver
Note: Publications are listed for ABAP-based application servers. Corresponding publications are also available for Java-based and mixed ABAP/Java-based (add-in) application servers.
<i>SAP NetWeaver 7.0 ABAP+Java on AIX: IBM DB2 for z/OS</i>
<i>SAP NetWeaver 7.0 ABAP+Java on Linux: IBM DB2 for z/OS</i>
<i>SAP Planning Guide for SAP, NetWeaver on IBM DB2 for z/OS SAP (NetWeaver 7.0)</i>
<i>SAP Database Administration Guide for SAP, NetWeaver on IBM DB2 for z/OS (SAP NetWeaver 7.0)</i>
<i>SAP Security Guide: IBM DB2 for z/OS</i>
<i>SAP System Copy Guide - System Copy for SAP Systems Based on SAP NetWeaver 7.0 ABAP+Java</i>
SAP Developer Network (SDN)
<ul style="list-style-type: none"> • Casebook - DB2 Backup, Recovery and Cloning for SAP Environments • Streamlining DB2 Connect for SAP • Unicode conversion • SAP for Banking on System z Reference Architecture • SAP for Insurance on System z Reference Architecture • DB2 for z/OS Specific Enhancements to the SAP Dictionary • Exploiting DB2 9 for z/OS Features: Partition-by-Growth with Universal Table Spaces • Best Practice for Installing or Migration to DB2 V8 • Best Practice for Installing or Migration to DB2 V9

SAP Notes

This section lists selected SAP Notes that are referenced in this publication and/or are useful in constructing and maintaining a high availability SAP system on the System z platform. It should serve as a reference list to assist you in your availability planning.

SAP Notes can be found in the SAP Service Marketplace.

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

Table 44. Relevant SAP Notes

SAP Note	Title
353529	Minimizing downtime when switching from and to Daylight Saving Time
496904	Data sharing optimizations for SAP banking applications
81737	APAR List
98051	Database Reconnect: Architecture and function

Bibliography

Table 44. Relevant SAP Notes (continued)

SAP Note	Title
407325	Released operating systems SAP kernel 6.x DB2/390
538081	High-availability SAPLICENSE
569996	High-availability and automation solution for DB2 on z/OS
728743	zSeries: Release of DB2 V8 for SAP Components
757692	Changing the hostname for J2EE Engine 6.40/7.0 installation
803018	Central note for NetWeaver04 High Availability capabilities
821904	Separating SCS instances for ABAP and Java
853510	Release Restr.: Usage Type PI of SAP NetWeaver 2004s
858969	SAP NetWeaver 7.0 (2004s) Installation: IBM DB2 for z/OS
915482	DB2-z/OS: Automating DB failover
951910	NW2004s High Availability Usage Type PI
953653	Rolling Kernel Switch
1085521	DB2/390: HA for DB outage for java stack (2)
1100775	SAP NetWeaver 7.0 (2004s) SR3 Installation IBM DB2 on z/OS
1032273	Configuring DB2 V9
1031213	Customizing WebAS 6.10 - 7.00 for V9 CLI Driver
1068654	Customizing V9 CLI driver for ICLI & WebAS 6.20
1178661	High Availability for SAP Business Intelligence Accelerator
1239127	Data sharing optimizations for SAP Business Intelligence
1260453	Data sharing optimizations for SAP PI

APARs

For an up-to-date list of all relevant APARs for SAP on DB2 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 Entwicklung GmbH
Department 3248
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 System z. Use of these materials is at your own risk.

Trademarks and service marks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
AIX 5L
CICS
Database 2
DB2
DB2 Connect
DB2 Universal Database

Notices

DFS
DFSMSdss
DFSMSHsm
Distributed Relational Database Architecture
DRDA
Enterprise Storage Server
Enterprise Systems Connection Architecture
ESCON
eServer
FlashCopy
GDPS
Geographically Dispersed Parallel Sysplex
HACMP
HiperSockets
IBM
IMS
Language Environment
MQSeries
MVS/DFP
MVS
Netfinity
NetView
OS/390
Parallel Sysplex
PR/SM
pSeries
RACF
Redbooks
RISC System/6000
RMF
RS/6000
S/390
SecureWay
Sysplex Timer
System/390
System p
System z
System z9
Tivoli
VTAM
WebSphere
xSeries
z9
z/OS
z/VM
zSeries

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

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

Microsoft, Windows, Windows NT, Windows 2000, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

Index

A

- ABAP application server
 - checkappsrv script 160
 - configuring for SA z/OS 159
 - shell scripts 159
 - startappsrv script 159
 - stopappsrv script 160
- ABAP SAP Central Services 325
 - configuring and starting 157
- ABAP_instances.conf file 215
- abbreviations 335
- active connections
 - checking 284
- AIX
 - application server timeout
 - behavior 86
 - Source VIPA 78, 80
- APARs
 - list of 350
- application server
 - APAP-only 183
 - checkappsrv script 322
 - checkjappsrv script 323
 - double-stack (ABAP plus Java) 184
 - Java-only 185
 - multiple DB2 members in same LPAR failover support 110
 - on AIX
 - timeout behavior 86
 - on Linux on System z
 - timeout behavior 88
 - on Windows
 - timeout behavior 89
 - remote 109
 - remote control under Windows 161
 - rfcping 160
 - sapctrl_em script 322
 - startappsrv script 320
 - startjappsrv script 322
 - stopappsrv script 321
- application server (SAP)
 - as SAP resource 183
- application server group
 - as SAP resource 187
- applications
 - checking for problems 281
 - health check 9
- architecture
 - database server 108
 - file system 106
 - network 102
 - of high availability solution 95
 - options and trade-offs 45
- ARP takeover function 83
- Automatic Restart Management (ARM) 110
- automation
 - objectives for SAP 8
- Automation Table
 - additions for DFS/SMB 192
 - additions to 190

- autonomic computing
 - self-managing systems 3, 27
- availability
 - test scenarios 286
 - with data-sharing configuration 109
 - with non-data-sharing configuration 108
- availability features
 - DB2 data sharing 35
 - DB2 for z/OS 29
 - Parallel Sysplex 28
 - System z hardware 27
 - z/OS 27

B

- backup and recovery
 - with data sharing 57
- BACKUP SYSTEM utility
 - DB2 for z/OS 62
- bibliography
 - IBM documents 347
 - SAP documents 349

C

- central instance
 - DB2 connection failover 114
 - double network 114
 - replaced by SAP Central Services 99
 - with data sharing 114
 - without data sharing 112
- change management
 - DB2 245
 - DB2 Connect 244
 - SAP kernel 241
 - z/OS 245
- checkjappsrv
 - start a Java application server monitor (GetWebPage) 163
- client
 - connection timeout 88
 - AIX 86
 - Windows 89
 - idle timeout
 - AIX 87
 - Linux on System z 88
 - Windows 90
 - transmission timeout 88
 - TCP/IP on AIX 86
 - Windows 89
- configuration structure
 - DB2 connection failover 109
- connection failover 41
- connection status
 - DB2 for z/OS 285
 - SAP 285
- connection timeout
 - Linux on System z client 88

- connections
 - checking 284

D

- data sets
 - PROFILE.TCPIP 90, 91
- data sharing groups
 - determining number of 51
- data sharing members
 - determining number of 52
- database server
 - architecture for high availability 108
 - idle timeout 91
 - primary 110
 - standby 109, 110
 - transmission timeout 90
- DB connection failover 72
- DB2
 - connection failover architecture 41
- DB2 Connect
 - rolling update 244
- DB2 connection failover 109
 - central instance 114
- DB2 data sharing
 - architecture 40
 - availability considerations 109
 - availability features 35
 - availability scenarios 37
 - backup and recovery architecture 57
 - central instance 114
 - central instance without 112
 - considerations for disaster recovery 57
 - design options for SAP 45
 - failover design 54
 - groups 110
 - homogeneous system copy 64
 - members 110
 - on Parallel Sysplex 39
 - SAP benefits 37
- DB2 database server group (SAPHA1_DBX)
 - as SAP resource 182
- DB2 exception events
 - deadlocks 92
- DB2 for z/OS
 - "light" restart 36
 - availability features 29
 - BACKUP SYSTEM utility 62
 - Best Practise Policy 125
 - check if running 285
 - checking connection status 285
 - data sharing 35, 109
 - duplexing of SCA and lock structures 36
 - group buffer pool duplexing 36
 - improvements in recent releases 36
 - multiple DB2 members in same LPAR 110
 - non-data-sharing 108

- DB2 for z/OS (*continued*)
 - non-disruptive software changes 29, 35
 - planning information 125
 - updating 245
- DB2 members
 - in same LPAR 110
- DB2 - Best Practise Policy 125
- DDF
 - keep-alive interval times 91
- deadlock detection interval 92
- DFS/SMB
 - additions to Automation Table 192
 - extensions for 191
- disaster recovery
 - data sharing considerations for 57
 - GDPS infrastructure for 60
 - tracker site 60
- Domain Name Server (DNS)
 - settings 307
- dynamic VIPA 283

E

- end-to-end automation management 227
- end-to-end component
 - of SA MP 227
- enqueue replication server
 - failure scenario 116
- enqueue server
 - failure of 264

F

- failover
 - multiple DB2 members in same LPAR 110
 - of NFS server 107
 - of SAP Central Services 101
- failover scenarios
 - Linux on System z 289
 - SA z/OS policy 249
- failure
 - of an LPAR 273
 - of enqueue server 264
 - of message server 267
 - of NFS server 269
 - of TCP/IP stack 270
- failure scenarios
 - impact on SAP system 111
- file system
 - architecture 106
 - NFS 284
 - planning information 125
 - setup 311
 - shared HFS 284

G

- GDPS
 - infrastructure for disaster recovery 60
- GetWebPage 9
- GetWebPage java program 330
- GetWebPage java utility 163
- glossary 341

- group buffer pool
 - duplexing 36

H

- health check
 - applications 9
- high availability
 - definitions 5
 - objectives for SAP 8
 - recommended setup
 - OSPF 79
 - recovery attributes 82
 - recommended setup for client/server connections 79
 - SAP sysplex failover 75
- high availability policy for SAP (SA MP)
 - customizing 215
 - installing 199
- high availability scripts
 - for Tivoli System Automation for z/OS 319
- high availability solution for SAP
 - architecture 95
 - automation 10
 - overview 3, 9
 - planning 13
 - planning and preparing for 119
 - software prerequisites 120
- HiperSockets 73
- homogeneous system copy (HSC)
 - from data sharing to data sharing 67
 - from data sharing to
 - non-data-sharing 68
 - in data sharing 64
 - in non-data-sharing 65
 - offline copy 68
 - online copy 67

I

- idle timeout
 - database server 91
 - Linux on System z client 88
- Internal Resource Lock Manager (IRLM) 91
- iptables
 - as alternative to Source VIPA 81

J

- Java SAP Central Services 326
- Java_instances.conf file 215

K

- keep-alive
 - DDF 91
 - probes 87, 90, 91

L

- Link State Advertisements 76

- Linux on System z
 - application server timeout
 - behavior 88
 - failover scenarios 289
 - mount commands 313
 - multiple guests under z/VM 74
 - network settings 306
 - verification 289
 - Zebra setup 306
- load balancing
 - OSPF 76
- lock structures
 - duplexing 36
- LPAR
 - failure of 273
 - shutdown and restart 260
- LPAR-to-LPAR communication 73

M

- Message Processing Facility (z/OS) 277
- message server
 - failure of 267

N

- naming conventions
 - Tivoli System Automation for Multiplatforms 124
 - Tivoli System Automation for z/OS 121
- NetView
 - netlog 276
 - planning information 132
 - region size 172
 - setup 315
- network
 - architecture considerations 102
 - central instance 114
 - hardware 297
 - problem determination 282
 - setup 297
 - setup recommendations 72
- network attributes
 - AIX 86, 87
 - Linux on System z 88
 - tcp_keepalive_interval 88
 - tcp_keepalive_probes 88
 - tcp_keepalive_time 88
 - tcp_retries2 88
 - Linux on System z application server 88
 - tcp_syn_retries 88
- network failures
 - impact levels 71
- network setup
 - DNS settings 307
 - Linux on System z 306
 - z/OS settings 298
- NFS
 - attribute file 313
 - checking status 284
 - export file 311
 - high availability with SA MP 208
 - NFS failover 116

- NFS server
 - failover 107
 - failure of 269
 - on z/OS 128
 - setup procedure 311
- NIC
 - failure recovery 77
- NIC failure recovery
 - subnet configuration 79
 - VIPA 79
- non-data-sharing
 - availability considerations 108
- non-disruptive software changes
 - DB2 for z/OS 29, 35

O

- Open Shortest Path First (OSPF)
 - as recovery mechanism 72, 76
 - configuration aspects 79
 - dead router interval 83
 - gated daemon sample definition 307
 - implementation 76
 - load balancing 76
 - tables 283
- outages
 - planned 6
 - types of 6
 - unplanned 6

P

- Parallel Sysplex
 - architecture 39
 - availability features 28
 - availability scenarios 37
 - DB2 data sharing on 39
 - SAP benefits 37
- parameters
 - AIX
 - rto_high 86
 - rto_length 86, 87
 - rto_limit 86
 - rto_low 86
 - tcp_keepidle 87
 - tcp_keepinit 86
 - tcp_keepintvl 87
 - SAP profile parameters
 - rdisp/max_wprun_time 90
 - TCP/IP on Windows parameters 89
 - Windows
 - KeepAliveInterval 90
 - KeepAliveTime 90
 - TcpMaxConnect
 - Retransmissions 89
 - TcpMaxDataRetransmissions 89
- path MTU discovery 77
- problem determination
 - Linux/AIX 289
 - Tivoli System Automation for
 - z/OS 276, 278
 - z/OS 249

Q

- quorum, SA MP 225

R

- recovery
 - of SAP Central Services 101
 - remote using archive logs 58
- recovery mechanisms
 - DB connection failover 72
 - dynamic routing (OSPF) 72
 - on Windows 83
 - OSPF 76
 - SAP sysplex failover 75
 - Virtual IP Addresses (VIPAs) 72, 77
- recovery site
 - configuring 57
- registry values
 - Windows 89, 90
- remote application server
 - and DB2 connection failover 109
- remote execution
 - of scripts 160
- remote site recovery 58
- resource timeout 91
- RFC connections
 - setup for SAP 165
- rfcping
 - application server check 160
- rfcping program 330
- rfcping utility 9
- rolling update
 - of DB2 Connect 244
- rolling upgrade
 - of SAP kernel 243
- routing tables 283

S

- SA MP (Base)
 - of Tivoli System Automation for
 - Multiplatforms 195
- SA MP high availability policy for SAP
 - GetWebPage java program 330
 - service IP addresses (VIPAs) 331
 - setup scripts 331
- SA MP quorums 225
- SANCHK 315
- SAP
 - checking connection status 285
 - checking database connections 285
 - command summaries for Tivoli
 - System Automation 166
 - configuring for Tivoli System
 - Automation for z/OS 157
 - customizing for high availability 135
 - directory definitions 127
 - high availability and automation
 - objectives 8
 - license considerations 133
 - logon groups 133
 - setup for RFC connections 165
- SAP availability
 - System z 27
- SAP benefits
 - DB2 data sharing 37
 - Parallel Sysplex 37
- SAP Central Services
 - failover 101
 - failure scenario 116
 - SAP Central Services (*continued*)
 - recovery 101
 - replacement for central instance 99
- SAP installation
 - planning information 132
- SAP kernel
 - rolling upgrade 243
- SAP Notes
 - list of 349
- SAP resources
 - classes 183
 - DB2 database server group
 - (SAPHA1_DBX) 182
 - defining in Tivoli System Automation
 - for z/OS 172
- SAP system
 - failure scenarios 111
- SAP transactions
 - maximum time 90
 - rollbacks 109
- SAP work processes 109
- sapccmsr
 - registering 163
- SAPCCMSR
 - directory 128
- SAPHA1ACI application server 183
- saposcol
 - starting/stopping via System
 - Automation 163
- SAPOSCOL
 - directory 128
- SAProuter 166
- SCA
 - duplexing 36
 - self-managing systems 3
 - service IP addresses (VIPAs) 331
- Shared HFS
 - checking status 284
- Source VIPA
 - iptables as alternative 81
 - on AIX 78, 80
 - on remote application servers 80
- standalone enqueue server 100
- startjappsvr
 - start a Java application server
 - instance 162
- syslogd, USS syslog daemon 172
- sysplex failover
 - as recovery mechanism 75
- sysplexes
 - determining number of 52
- System Automation
 - checkjappsvr (start a Java application
 - server monitor, GetWebPage) 163
 - registering sapccmsr 163
 - self-healing technologies for
 - autonomic computing 7
 - starting/stopping saposcol 163
 - startjappsvr (start a Java application
 - server instance) 162
- System z
 - availability features 27
 - Parallel Sysplex features and
 - benefits 28
 - SAP availability benefits 27

T

- TCP/IP
 - failure of 270
- test scenarios
 - for availability 286
 - planned outages 258
 - unplanned outages 264
- tie breaker 224
- timeout behavior
 - Linux on System z application server 88
 - of AIX application server 86
 - of database server 90
 - of Windows application server 89
- Tivoli System Automation
 - planning information 132
- Tivoli System Automation for Multiplatforms
 - ABAP_instances.conf file 215
 - automation scripts 332
 - customizing high availability policy 215
 - customizing the SA AM (E2E) 227
 - customizing the SA MP (Base) 195
 - documentation 347
 - end-to-end high-availability policy for SAP 237
 - high availability policy (details) 325
 - installing 198
 - installing high availability policy 199
 - Java_instances.conf file 215
 - naming conventions 124
 - NFS high availability with 208
 - overview 196
 - removing policy 223
 - removing SAP policy 332
 - SAP setup 197
 - setting up to manage SAP resources 199
 - setup 132
 - using a tie breaker 224
 - verifying 223
- Tivoli System Automation for z/OS
 - adapting the best practices policy for SAP 172
 - checkappsrv script 322
 - checkjappsrv script 323
 - configuring SAP for 157
 - customizing 171
 - documentation 347
 - high availability benefits 10
 - high availability scripts 319
 - initialization exit (AOFEXDEF) 171
 - naming conventions 121
 - planning information 132
 - preparing for high availability 171
 - problem determination 276, 278
 - SANCHK 315
 - sapctrl_em script 322
 - setup 315
 - startappsrv script 320
 - startjappsrv script 322
 - stopappsrv script 321
- Tivoli System Automation for z/OS policy
 - failover scenarios 249
 - verification 249

- tracker site
 - for disaster recovery 60
- transmission timeout
 - database server 90
 - Linux on System z client 88

U

- UNIX messages
 - sending to NetView 280
 - sending to syslog 172, 280
- UNIX System Services
 - setup 298
- USS PARMLIB member BPXPRMxx 128
- USS zFS, filesystem type 126

V

- verification
 - Linux on System z 289
 - Linux/AIX 289
 - SA z/OS policy 249
 - z/OS 249
- VIPA
 - as recovery mechanism 72, 77
 - dynamic 78, 85
 - Source VIPA on AIX 81
 - Source VIPA on remote application servers 80
 - static 78, 85
 - z/OS 85, 298
- VIPAs (service IP addresses) 331
- Virtual IP Address
 - See VIPA 77
- Virtual Switch (VSWITCH) 74

W

- Windows
 - registry values for timeout 89
 - remote control of application servers 161

Z

- z/OS
 - availability features 27
 - failure of 110
 - Message Processing Facility 277
 - networking software 73
 - NFS server on 128
 - non-disruptive software changes 29
 - syslog 277, 280
 - updating 245
 - VIPA 85, 298
- z/VM
 - multiple Linux on System z guests 74
 - Virtual Switch (VSWITCH) 74
- Zebra setup 306
- zFS, filesystem type (USS) 126

Readers' Comments — We'd Like to Hear from You

SAP on System z
Business Continuity for SAP on IBM System z

Publication No. SC33-8206-02

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 a fax to the following number: FAX (Germany): 07031+16-3456
FAX (Other Countries): (+49)+7031-16-3456
- Send your comments via e-mail 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.

E-mail address



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Research and Development GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany
71032-0000



Fold and Tape

Please do not staple

Fold and Tape



Printed in USA

SC33-8206-02

