

SAP on System z



Business Continuity for SAP on IBM System z - Draft

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 Practice Policy" for SAP	172
Overview of the resources	173
Description of the group structure	174
SAP system dependent groups	175
SAP infrastructure group	175
ABAP central services and enqueue replication server	177
Dependencies between the ABAP enqueue server and the enqueue replication server	179
Optional components of the ABAP central services	181
JAVA central services and enqueue replication server	181
DB2 policy	182
DB2 database server group (SAPHA1_DBX)	183
Classes	184
C_SAP_USS	184
SAP application servers	184
APAP-only application server	184
Double-stack (ABAP plus Java) application server	185
Java-only application server	187
SAP Application server groups	188
SAPHA1RAS	188
SAPHA1RASX	188
Overview of groups/applications	190
Adding entries to the Automation Table	191
Adding the definitions for extension DFS/SMB	192
Additions to the SA z/OS policy	192
Application	192
DFS_SMB	192
Application group	193
SMB_PLEX	193
Additions to the Automation Table for DFS/SMB	193

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 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

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 Practice Policy” for SAP” on page 172
- “Overview of groups/applications” on page 190
- “Adding entries to the Automation Table” on page 191
- “Adding the definitions for extension DFS/SMB” on page 192

Preparing SA z/OS for SAP high availability

This section describes 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

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 Practice Policy" for SAP

As a prerequisite, you must have installed and activated the z/OS SA APAR OA26776. This APAR includes the SAP “Best Practice Policy” version, which is assumed throughout this chapter. The SAP policy comes as a predefined add-on policy which must be customized and imported through 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, follow the naming conventions that are described in “Conventions used in the SA z/OS policy” on page 124.

Adapting the SAP “Best Practice 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. Do not forget to import the adapted sysplex and systems resources, because only then the import generates the linked resources.

- If you do not have an OSPF based (dynamic routing based) highly available network between application server and DB server, you must replace the HASPARENT relationship of the SAP and NFS server resources to the OMPROUTE application by a HASPARENT to TCPIP application. You must also change the ForceDown relationship of the SAP and NFS server resources to the OMPROUTE application by a ForceDown with condition WhenObservedAssumedDownOrStopping to the 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 Messages and User Data:
 - the messages for VIPAs and enqueue replication servers
 - Under Startup:
 - the Startup commands
 - the Poststart commands
 - Under Shutdown:
 - all phases which are not inherited through class C_SAP_USS
 - 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. Refer to Appendix E “Automation scripts” on page 332 for details and invocation syntax.

After the import and if you use our recommended setup of running the HA NFS server on z/OS, you must manually add the NFS server group (NFS_SERV_X), which is available in the USS “Best Practice” or add-on policy, to the SAP infrastructure group (SAPSYS_X).

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, stopjappsrv_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 might be needed for your specific environment. For example, for an ABAP-only SAP system you will *not* need the JAVA components. Therefore the first step in adapting the policy for a specific SAP system will be to determine 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.

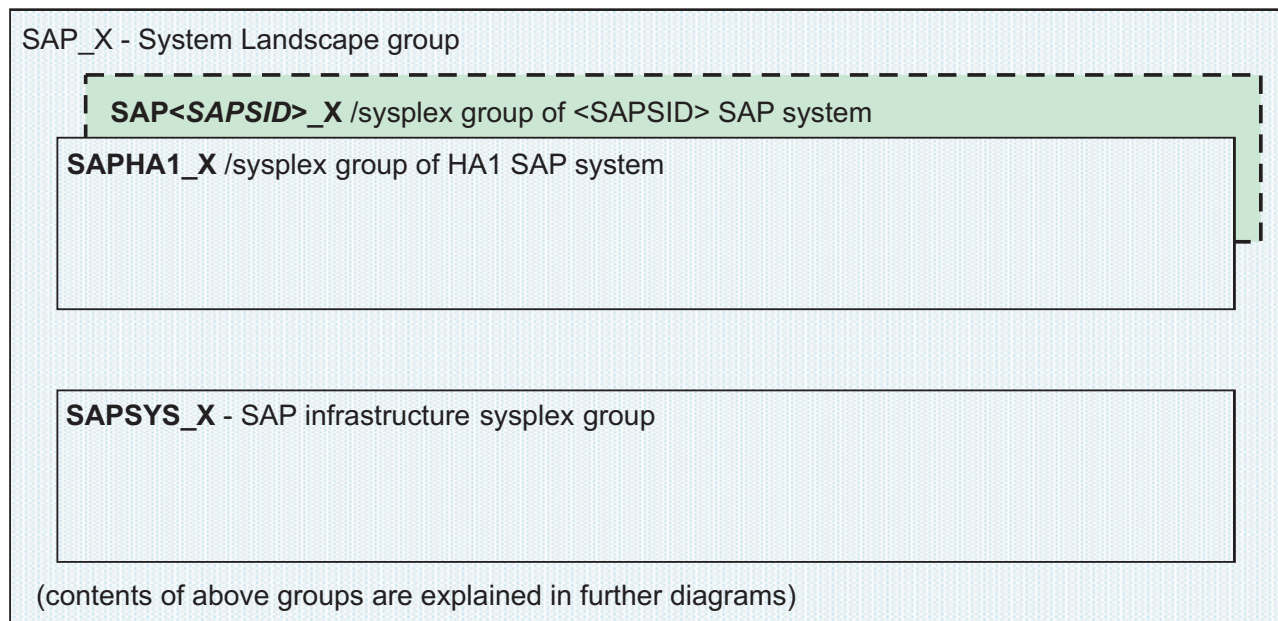
Customizing Tivoli System Automation for z/OS

- 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
 - The NFS server is not included in the SAP “Best Practice Policy”. It is available in the “USS” policy. You must manually add the NFS server group (NFS_SERV_X) to the SAP infrastructure group (SAPSYS_X).
 - 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 Practice 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 recommended general naming convention 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).

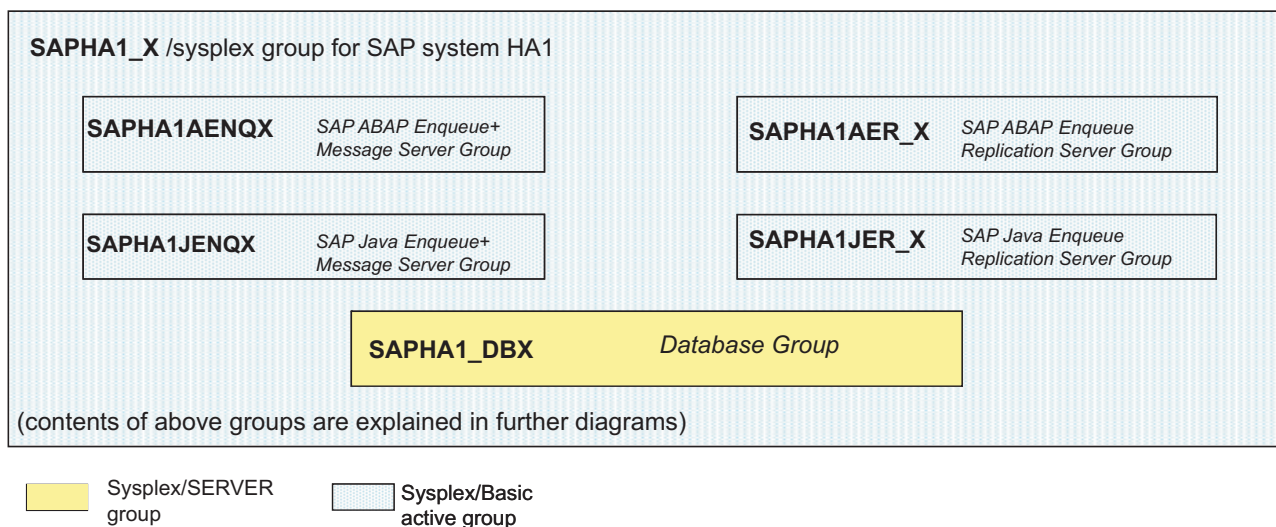


Figure 40. Group SAPHA1_X belonging to SAP system HA1

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 “Best Practice Policy” for a highly available NFS server is part of the SA z/OS “USS” policy. Adapt this part of the “USS” policy to your environment.

Customizing Tivoli System Automation for z/OS

Make sure you have setup the NFS server/clients according to the specifications described in: Chapter 9, "Preparing a high availability SAP solution," on page 119

Note: For optimum access performance of the NFS server exported file systems, the ownership of the file systems **must** be held by the LPAR, where the NFS server runs. If the NFS server is (re)started for whatever reason on another LPAR by System Automation, the ownership of the exported file systems must be moved exactly to that LPAR. This is done by adding a poststart command for each file system to the NFS server resource, like in the following example:

```
Poststart:
CMD=(,,'MVS SETOMVS FILESYS,FILESYSTEM='OMVS.ZFS.COHPLEX.SAPMNT',SYSNAME=COH1')
CMD=(,,'MVS SETOMVS FILESYS,FILESYSTEM='OMVS.ZFS.COHPLEX.TRANS',SYSNAME=COH1')
```

- The group required to perform SAP monitoring (SAPSYSOSC_X).
- An optional group for the SAP router (SAPSYSRTE_X).

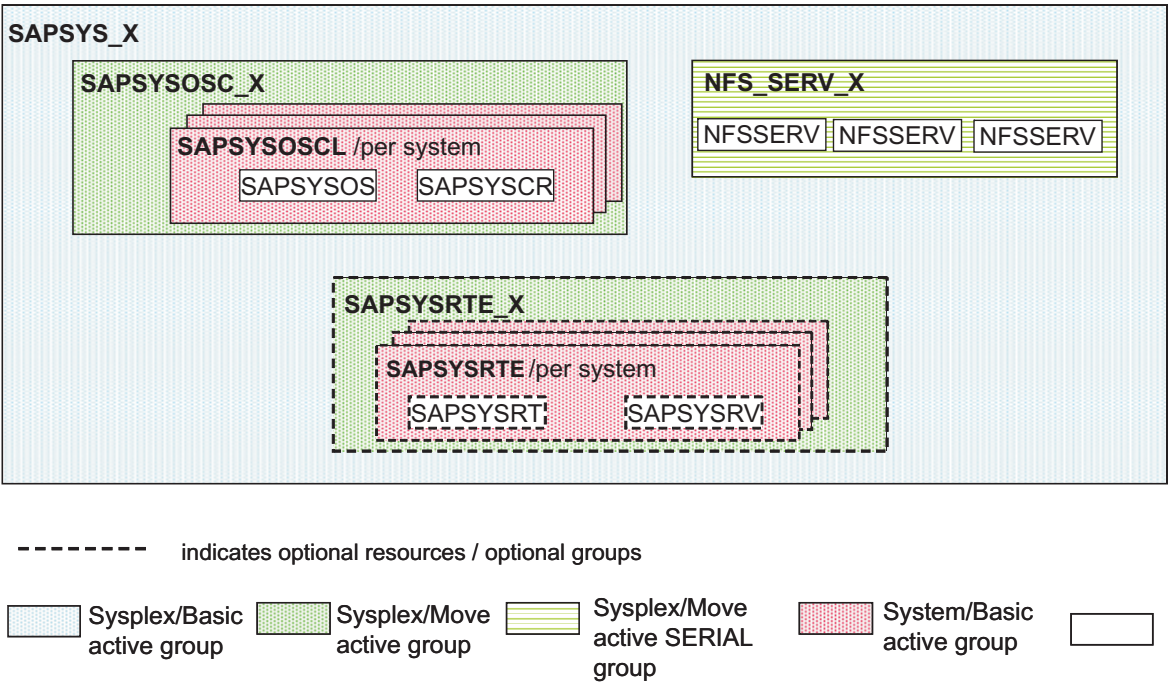


Figure 41. SAP-system-independent groups and resources

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.

Note: You must manually add the high level NFS server group to the high level SAP infrastructure group. For the samples here this means, that you must add the NFS_SERV_X group to the SAPSYS_X group via the SA z/OS customization dialog.

- To enable SAP monitoring on z/OS, we recommend that you have only *one* SAP operating system collector *saposc* 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 SAPSYSCR, 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, define the SYSPLEX MOVE group SAPSYSOSC_X containing the SAPSYSOSCL group as shown in Figure 41 on page 176.
- 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 SAPSYSRT and SAPSYSRV 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, define the SYSPLEX/MOVE group SAPSYSRTE_X containing SAPSYSRTE (as shown in Figure 41 on page 176).
- 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 Practice Policy" as a template.

ABAP central services and enqueue replication server

This section describes 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 181) are very similar to this one.

Figure 42 on page 178 shows the lowest level in the group structure of the ABAP central services and enqueue replication server.

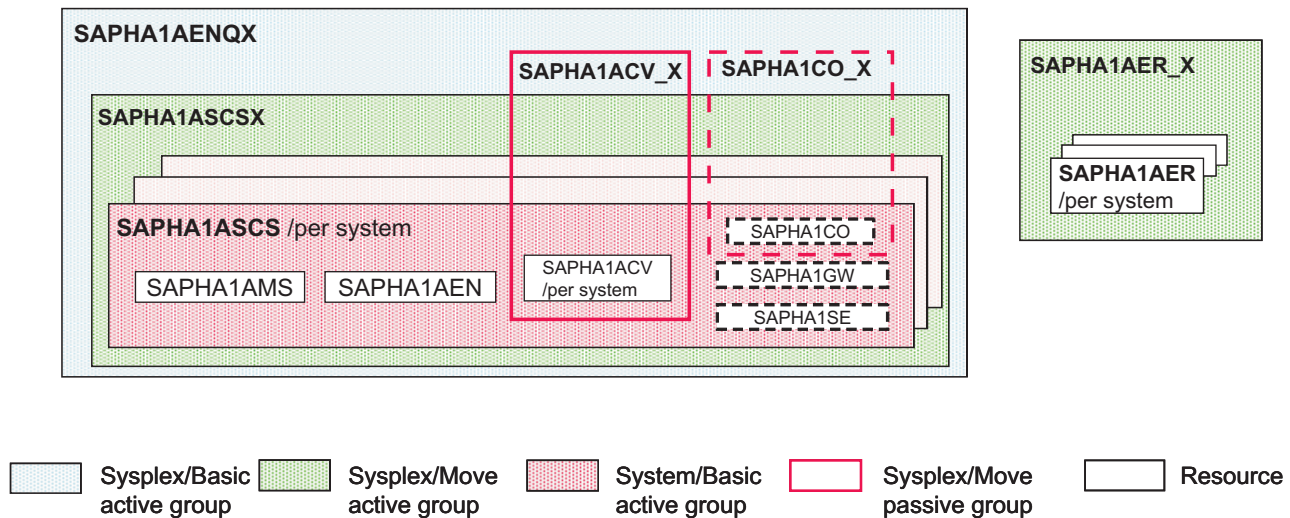


Figure 42. Lowest level in group structure of ABAP central services and enqueue replication server

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 180) 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 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

Customizing Tivoli System Automation for z/OS

PREFerence 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 “Messages and User data” section of the policy definition and in the “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 PREFerence value to 700 for the SAPHA1ASCS and the LPAR where the enqueue replication server (SAPHA1AER) is running. This requires, that the default PREFerence value of each SAPHA1ASCS group is set to 601 in the SAPHA1ASCSX SYSPLEX MOVE group.

Table 18. Startup section from the SAPHA1AER policy definition

Message ID	Command Text
ACORESTART	INGGROUP SAPHA1ASCSX/APG,ACTION=ADJUST,MEMBERS=SAPHA1ASCS/APG/ &SYSNAME.),PREF=(700)
POSTSTART	INGGROUP SAPHA1ASCSX/APG,ACTION=RESET INGGROUP SAPHA1ASCSX/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 Practice Policy”): In a two-LPAR environment, this might prevent the enqueue replication server from restarting at all. 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 the 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 might start to move back and forth if the it fails over and over again with the same error:

1. Enqueue server fails on LPAR1.
2. SA 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 179). 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 182 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.

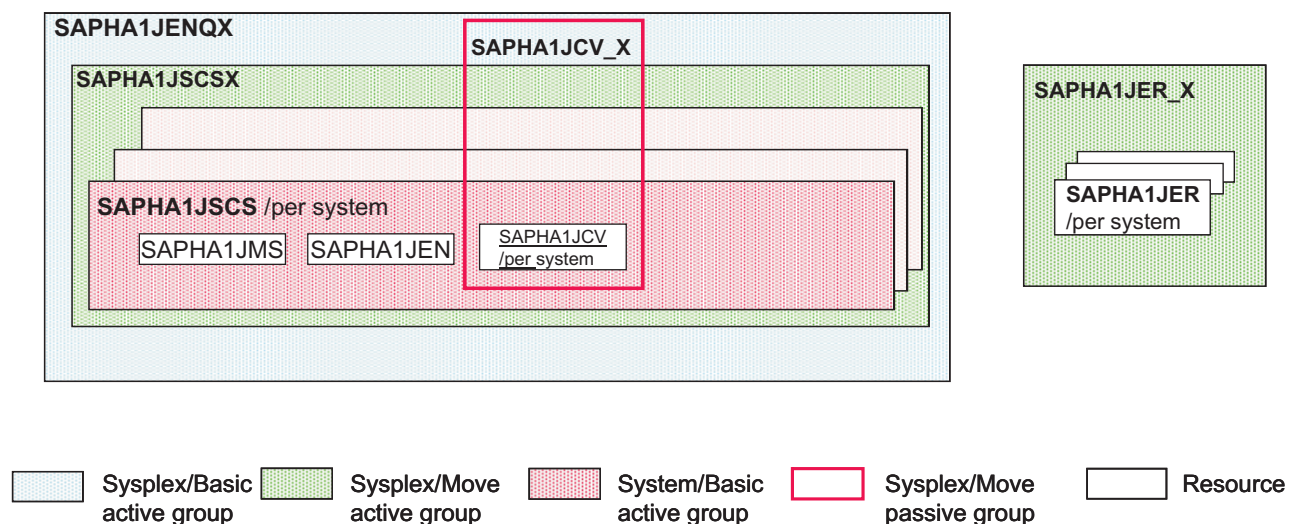


Figure 43. Lowest level in group structure of JAVA central services and enqueue replication groups

DB2 policy

As a prerequisite, 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 183 shows how IBM exploits SA's DB2 “Best Practice Policy” within the test environment.

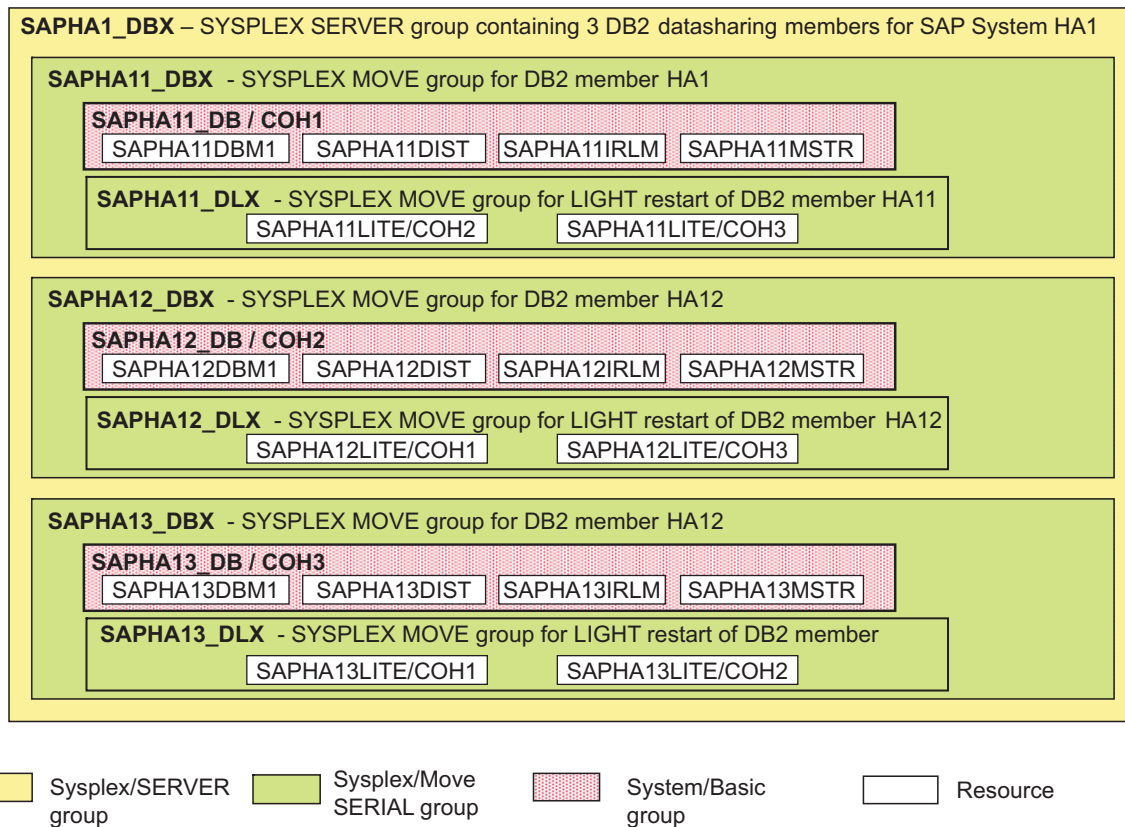


Figure 44. DB2 “Best Practice Policy” – adapted for SAP system HA1

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.

Note: You must manually add the high level DB2 group to the high level SAP group. For the samples here this means, that you must add the SAPHA1_DBX to the SAPHA1_X group via SA z/OS customization dialog.

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, one class only is used, 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. To recover SA z/OS from these situations, the restart option must be changed to ALWAYS.

SAP application servers

With the availability of SA MP, together with the end-to-end application manager, use SA MP and SA AM to control remote ABAP application servers running on Linux/AIX platforms. This section provides 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, *two* STOP commands are defined:

- 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

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

Startup STARTUP
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/haladm/startappsrv_v5 ihlscoh1
02 DVEBMGS02 SSH >& /u/haladm/startappsrv_02.ihlscoh1.log'

Startup POSTSTART
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/haladm/checkappsrv_v4 ihlscoh1
02 >& /u/haladm/checkappsrv_02.ihlscoh1.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/haladm/stopappsrv_v5 ihlscoh1 02 DVEBMGS02 SSH
>& /u/haladm/stopappsrv_02.ihlscoh1.log'

USS Control
User ID. . . . . HA1ADM
Command/Path . . . . ./rfcping_ihlscoh1_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 184).

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.

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, *two* STOP commands are defined:

- 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/haladm/startjappsrv ihlscoh2
94 JC94 2 SSH > /u/haladm/startjappsrv_94.ihlscoh2.log 2>&1'
```

Startup POSTSTART

```
INGUSS JOBNAME=&SUBSJOB,/bin/sh -L -c '/u/haladm/checkjappsrv ihlscoh2
94 > /u/haladm/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/haladm/stopappsrv_v5 ihlscoh2 94 JC94 SSH >&
/u/haladm/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.

The configuration contains only one remote double-stack AS, 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 there is 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 *MAKEAVAILABLE/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, such an SA policy extension is not recommended.

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 189 (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.

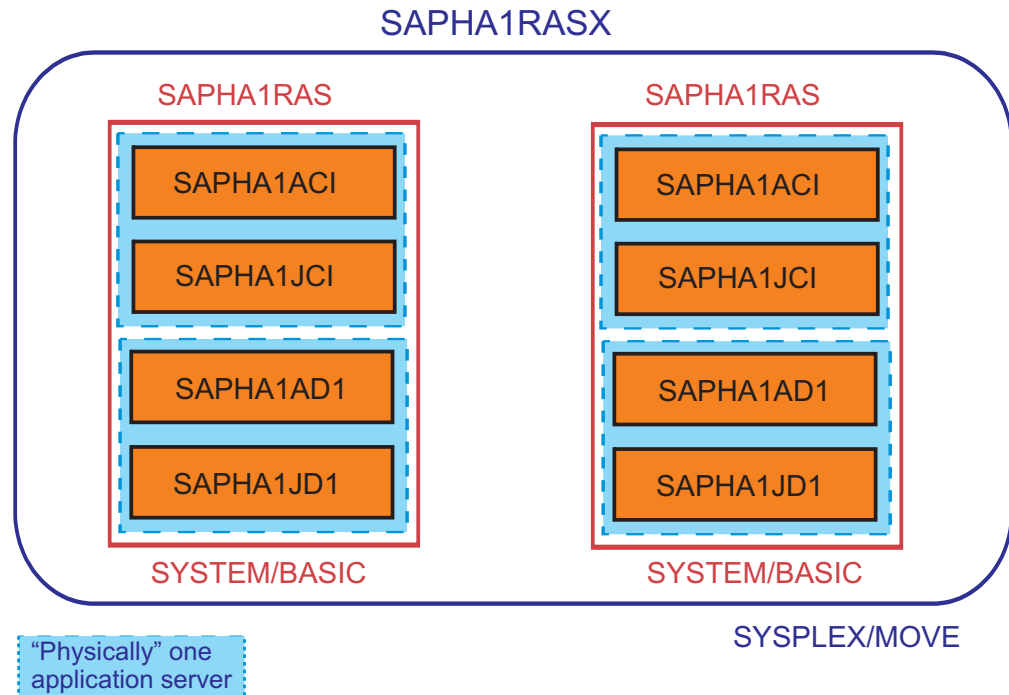


Figure 45. SAPHA1RASX application group

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 Practice Policy”. To improve readability, SYSTEM/Basic groups are only shown as *one box*, although *one such group per LPAR* exists in the SYSPLEX.

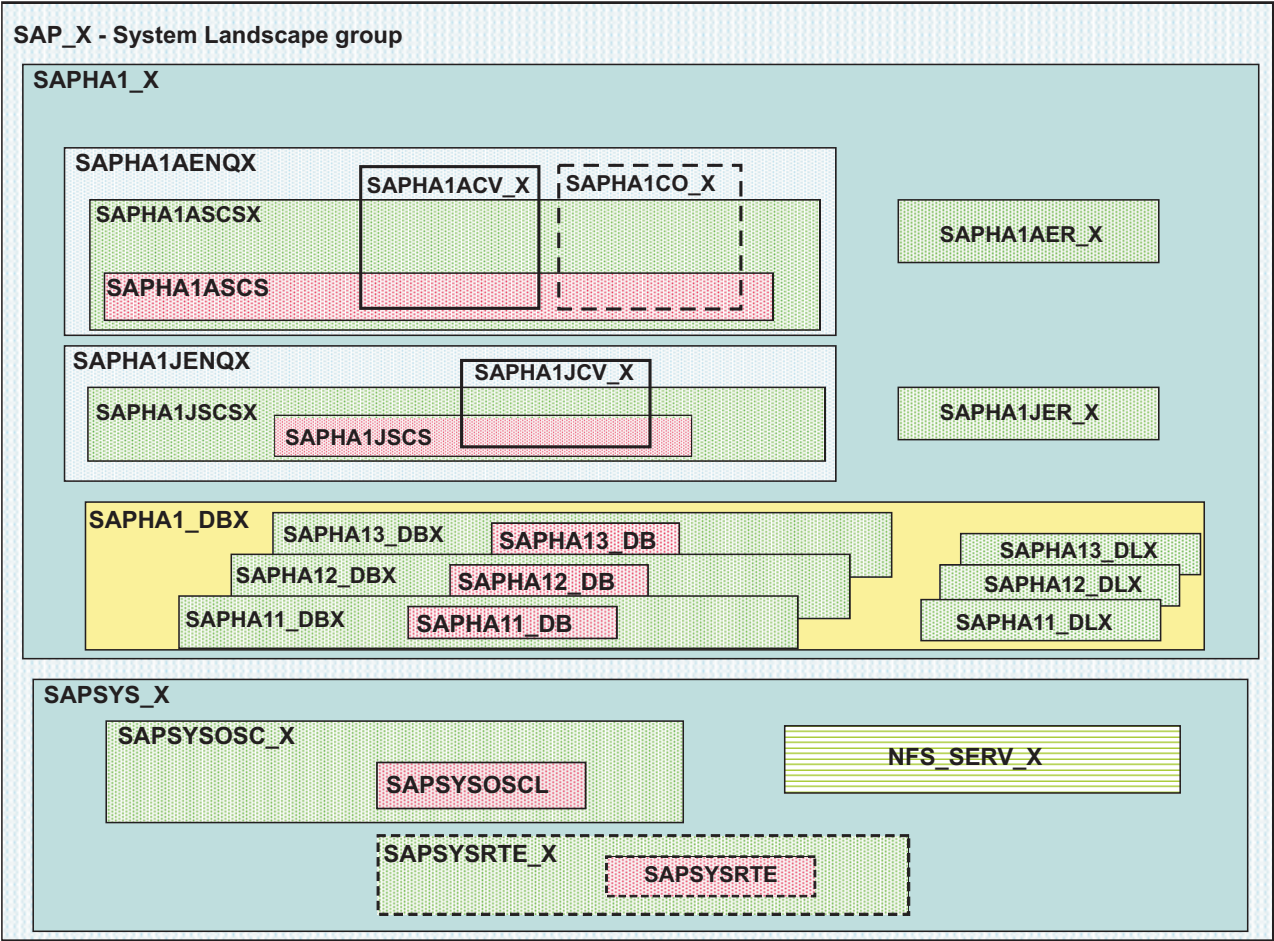


Figure 46. Overview of the resources

1 Figure 47 on page 191 provides an overview of the relationships of the SAP “Best
1 Practice Policy” that is provided for SAP NetWeaver 7.0.
1



1

1

Customizing Tivoli System Automation for z/OS

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('TERMSG 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 Practice Policy” for SAP” on page 172, describing 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, the additions to the SA z/OS policy are provided.

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.

```
MVS SETOMVS FILESYS,FILESYSTEM='SAPHA1.SHFS.TRANS',SYSNAME=&SYSNAME.
```

```
Shutdown NORM
1
MVS P &SUBSJ0B
4
MVS C &SUBSJ0B
```

Application group

Define one application group for DFS/SMB.

SMB_PLEX: DFS/SMB should run on one of the two systems at a time. Therefore, define a SYSPLEX/MOVE group with DFS/SMB, as shown in Figure 48 (active SA applications are represented as shaded boxes).

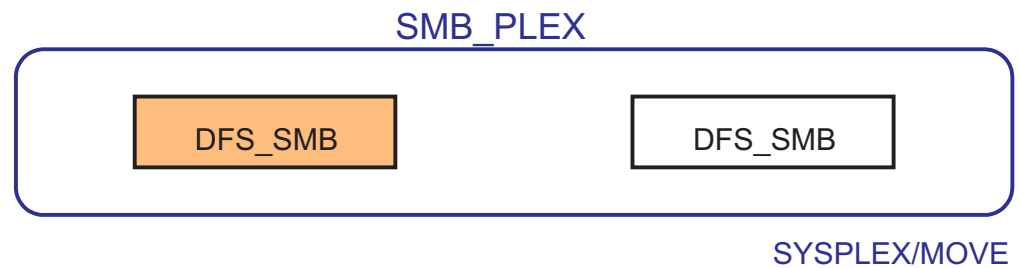


Figure 48. SMB_PLEX application group

```
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; therefore 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

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 %AOFPGSSOPER%);
*
    IF MSGID = 'IOEP01100I' .
        THEN EXEC(CMD('TERMMMSG FINAL=YES'))
            ROUTE(ONE %AOFPGSSOPER%);
*
    ALWAYS;
```