SAP on System z

IBM

High Availability for SAP on IBM System z Using Autonomic Computing Technologies

SC33-8206-01

SAP on System z

High Availability for SAP on IBM System z Using Autonomic Computing Technologies

SC33-8206-01

Note:

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 375.

Second Edition (June 2006)

This edition is a major update of *SAP on zSeries: High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33-8206-00. It applies to the following software components:

- SAP R/3 release 4.6D
- mySAP: SAP Web Application Server 6.20
- SAP NetWeaver '04
- SAP NetWeaver 2004s
- z/OS Release 1.4 (5694-A01)
- AIX Release 5.1 (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 OS/390 and z/OS Version 7 (5675-DB2) and DB2 Universal Database for z/OS Version 8 (5625-DB2)
- IBM Tivoli System Automation for z/OS V2.3
- IBM Tivoli System Automation for Multiplatforms V1.2.0.6
- Windows Server 2003

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 Entwicklung 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.

Contents

| Chapter 11. Customizing the Tivoli | System Automation for Multiplatforms | base component 199

| Chapter 12. Customizing the Tivoli | System Automation for Multiplatforms | end-to-end component 221

Part 5. Verification and problem determination 235

Chapter 13. Verification and problem determination on z/OS 237

Figures

Tables

Summary of changes

Summary of changes
to SC33–8206–00
SAP on zSeries:
High Availability for SAP on zSeries®
Using Autonomic Computing Technologies

New information:

- Description of version 4.2 of the sample High Availability Policy for SAP for Tivoli System Automation for Multiplatforms, which:
  - supports NetWeaver '04 J2EE-only installations such as EP 6.0 and thus the J2EE SCS (SAP Central Services), J2EE application servers and the SAP Web Dispatcher
  - supports NetWeaver '04 'add-in' installations such as XI 3.0 and thus ABAP and J2EE in parallel
  - supports more than one SAP system in one SA MP domain
  - when stopping application servers, reflects "pending offline" as long as ’stopsap’ is running
  - allows policy parts to be created and removed independently (CentralServices, SAPCommon, AppServer)
  - contains and uses a small J2EE Java™ application server health-check program
  - contains sample commands to create relationships from SAP policy resources to a high-availability database and NFS server resources
  - supports AIX

  See
  - Chapter 8, “Customizing SAP for high availability,” on page 129
  - Chapter 11, “Customizing the Tivoli System Automation for Multiplatforms base component,” on page 199
  - Chapter 12, “Customizing the Tivoli System Automation for Multiplatforms end-to-end component,” on page 221
  - Appendix F, “Detailed description of the sample Tivoli System Automation for Multiplatforms high availability policy for SAP,” on page 327
- Description of version 7 of the sample high availability scripts for z/OS® (see Chapter 10, “Customizing Tivoli System Automation for z/OS,” on page 161 and Appendix E, “Detailed description of the z/OS high availability scripts,” on page 315).
- Description of new DB2® tools (see “IBM DB2 Toolkits for z/OS: SAP Editions” on page 15).
- The use of `iptables` as an alternative to Source VIPA, which is not supported under certain conditions (see “VIPA and Source VIPA functions on remote application servers” on page 74)
- Information on the Server Time Protocol (STP) in a sysplex environment. See “Sysplex Timer® and Server Time Protocol (STP)” on page 6.

Information that has been removed or moved to other documentation:

- DB2 V6 and System Automation for OS/390® V2.2 are no longer supported.
- Deletions or relocations of text passages are not indicated by change bars.

Summary of changes

Changed information:

- The followingIBM® brand changes have been reflected:

Table 1.

Old trademark	**New trademark**
IBM eServer™	IBM Systems
zSeries	System z™
pSeries®	System p™
xSeries®	System x

Some product and publication titles have changed accordingly.

- References to ICLI interfaces have been updated to reflect the required conversion to DB2 DDF and DB2 Connect™ as of DB2 V8, which is now assumed as the standard case.
- SAP Central Services (SCS) is now differentiated by SAP as ABAP SCS (SAP abbreviation: *ASCS*) or J2EE SCS (SAP abbreviation: *SCS*). In the interest of clarity, the abbreviation 'SCS' in this book continues to refer to the SAP Central Services concept in general, and 'J2EE SCS' is used to designate the J2EE-based SCS. See “SAP Central Services” on page 91.
- A vertical bar (|) in the left margin indicates modifications and additions to the text with respect to the previous edition (SC33-8206-00 dated August 2004). Changes attributable solely to rebranding (of *zSeries* to *System z*, for example) are not reflected by change bars.
- Changes to figures are not indicated.

About this document

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®, 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.

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.

Important remarks

Unless otherwise stated, for SAP NetWeaver '04 (SAP Web Application Server 6.40) and above, references in this document to the term ICLI server[1] should be understood as applying to the DB2's Distributed Data Facility (DDF address space), and references to the ICLI client should be understood as applying to DB2 Connect.

Information applying only to the ICLI environment is enclosed in brackets as follows:

For ICLI only

End of For ICLI only

The described configuration applies to SAP R/3 4.6 and SAP Web Application Server 6.20. The concept applies to higher SAP releases as well, although policies and scripts for SA z/OS will need to be adapted.

[1] As of SAP NetWeaver '04 and DB2 V8, the ICLI client and server are no longer supported. Instead, on the z/OS side, the standard DB2 DDF address space is supported, with DB2 Connect used on remote application servers. Since the DDF address space is part of the standard z/OS System Automation policy for DB2, ICLI server resources are no longer needed.

Conventions and terminology used in this document

In this document, the following naming conventions apply:

- IBM DB2 Universal Database™ for z/OS (or OS/390) is usually referred to as DB2.
- The SAP on DB2 UDB 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 (64-bit) is 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).
- 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 V7 and V8. Refer to “Bibliography” on page 369 for specific information.
- The term *Planning Guide* encompasses separate documents pertaining to the respective SAP kernel release:
  - IBM document *SAP R/3 on DB2 for OS/390: Planning Guide, 2nd Edition, SAP R/3 Release 4.6D* for supported versions of SAP R/3
  - IBM document *SAP on DB2 for OS/390 and z/OS: Planning Guide, 2nd Edition, SAP Web Application Server 6.20* for SAP Web Application server versions up to and including 6.20 (including use of the 6.40 downward-compatible kernel with release 6.20), with ICLI as the remote SQL interface
  - SAP document *Planning Guide: z/OS Configuration for SAP on IBM DB2 Universal Database for z/OS*, applying to SAP Web Application server version 6.40 (SAP NetWeaver ’04) and higher, which requires DB2 UDB Version 8 and DB2 Connect as a replacement for ICLI.

  Unless otherwise stated, the term *Planning Guide* refers to the version for the system under discussion. To ensure clarity, the SAP version is indicated when appropriate (for example, *6.40 Planning Guide*). Full titles for this publications (and numbers where applicable) are provided in “SAP documents” on page 372.
- The IBM documentation *SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th Edition* is usually referred to as the *Connectivity Guide*..
- 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 372 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 372).
- SAP has designated SAP Central Services for ABAP as *ASCS* (ABAP SAP Central Services) and now applies the abbreviation *SCS* to the J2EE-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 *J2EE SCS* to designate the environment-dependent instances when required. See “SAP Central Services” on page 91.

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:

```
►►──Command──────────────────────────────────────────────────────────►◄
```

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

```
►►──┬────────┬───────────────────────────────────────────────────────►◄
    └─Option─┘
```

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 369 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:

“Introducing high availability and automation for SAP” on page xxi

Provides general information on high availability in an SAP environment.

Part 1, "Database considerations for high availability," on page 1

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

Part 2, "Network considerations for high availability," on page 63

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 3, "Application server considerations for high availability," on page 89

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 sysplex failover support. 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. It then describes what is needed to adapt the SAP system to the high availability solution, including configuring SAP for SCS and for Tivoli System Automation. Finally, it discusses issues in updating and upgrading the system components.

Part 4, "Autonomic operation of the high availability solution for SAP," on page 159

Discusses the customization of SA z/OS and the base and end-to-end management automation components of SA MP.

Part 5, "Verification and problem determination," on page 235

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.

″**Appendixes**″

Provide setup details for networking, file systems, Automatic Restart Management, NetView, 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 357

Contains a list of important abbreviations appearing in this document.

"Glossary" on page 363

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

"Bibliography" on page 369

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

Introducing high availability and automation for SAP

The solution documented in this book uses autonomic computing technologies of IBM Systems products to provide automation and high availability for SAP systems. The availability of a production SAP system is a critical business factor and therefore requires the highest level of availability. Continuous availability combines the characteristics of high availability (the ability to avoid unplanned outages by eliminating single points of failure) and continuous operation (the ability to avoid planned outages, such as for administrative or maintenance work) in order to keep the SAP system running as close to 24x365 as possible.

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:

Self-
Configuring
Self-
Healing
Self-
Optimizing
Self-
Protecting

Figure 1. The concept of autonomic computing

Since the initial announcement of SAP on DB2 UDB for OS/390 and z/OS[2] , we have used DB2 Parallel Sysplex® data sharing combined with the SAP sysplex failover feature to remove the database server as a single point of failure. This also gave customers the ability to avoid planned and unplanned outages of the database server. See "Remote application server and sysplex failover support" on page 105.

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

[2] Unless otherwise noted, the term "z/OS" also applies to its predecessor, OS/390.

server, which normally resides on a different system. The enqueue replication server stores the replication data in a shadow enqueue table that resides in shared memory. For a more detailed description of the enqueue server and replication server, see "SAP Central Services" on page 91 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'.

The SAP installation program SAPInst under NetWeaver '04 supports the installation of a J2EE SCS for J2EE-only systems. Under NetWeaver 2004s, it supports the installation of an ABAP SCS (ASCS) for ABAP-only systems and both variants for 'add-in' 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 6, "Architecture for a highly available solution for SAP," on page 91.

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 UDB for OS/390 and z/OS that provides for near continuous availability.

The solution is applicable to a homogeneous z/OS environment as well as to a heterogeneous environment. 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.

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

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 sysplex failover is an example of how this is accomplished in an SAP environment. Part 1, "Database considerations for high availability," on page 1 describes how a number of planned outages can be avoided by taking advantage of DB2 data sharing and SAP sysplex failover.

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

Continuous availability

Continuous availability combines the characteristics of high availability and continuous operation to provide the ability to keep the SAP system running as close to 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 xxv). 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.

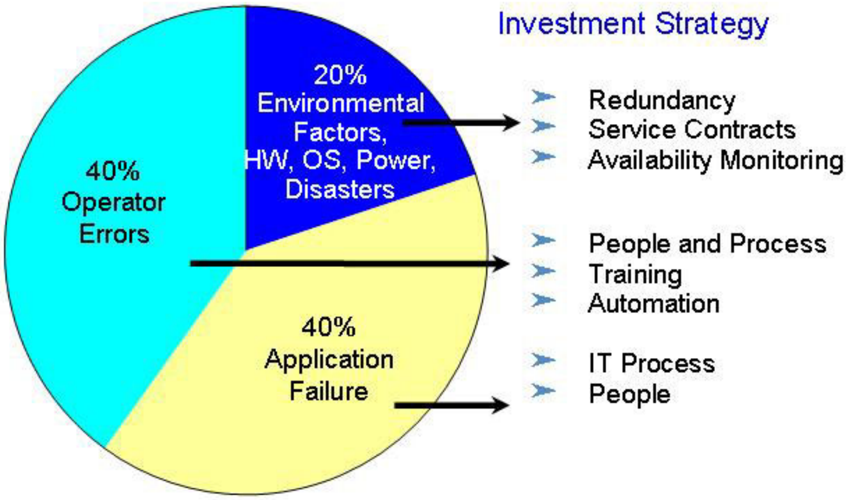


Figure 2. Causes of application downtime and appropriate response

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:

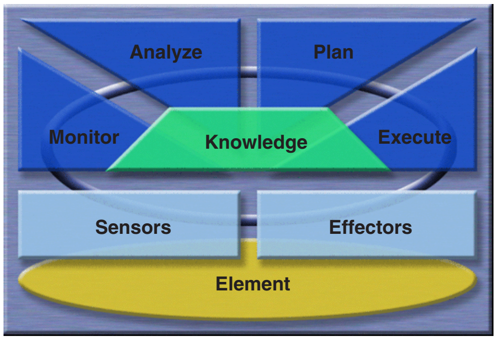


Figure 3. The closed loop of automation

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 SAP sysplex failover to dynamically move remote application server instances to standby database servers. The procedures for doing this are documented in Part 1, “Database considerations for high availability,” on page 1.

SAP release 4.6 added the capability to switch an application server instance between the primary and secondary database server using transactions DB2 or

ST04. Prior to this the only way to switch database servers was to stop the active ICLI. See SAP Note 509529 for further details.

SAP release 6.10 further extended sysplex failover by adding the capability to define connections to multiple database servers. The application server instance cycles through the defined database servers in the event of a failure, or when using transaction DB2 or ST04. For details, see the section ″Sysplex Failover and Connection Profile″ in the SAP installation guide.

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 SAP sysplex 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 8, “Customizing SAP for high availability,” on page 129 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 application servers. The high availability solution for SAP uses SA to invoke the monitoring task at regular intervals to check the status of application server instances. The monitoring task issues an RFC call to the application server and waits for a response. If a response is received, then the monitor ends. If a response is not received, the monitor signals SA that the application server instance is down. For a more detailed description of rfcping, see “rfcping” on page 143.

Overview of the high availability solution for SAP

High availability of an SAP system

As described in “High availability” on page xxiv, 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, ICLI servers, local and remote application server instances, enqueue server, message server, syslog collector and sender, gateway server, enqueue replication server, TCP/IP, and NFS server. By automating all the SAP components, the SAP system can be started, stopped, and monitored as a single resource. This provides for the highest level of availability by reducing operator commands, thus reducing the chance for operator errors.

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

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-linux
```

Part 1. Database considerations for high availability

Chapter 1. SAP availability benefits provided by System z

The IBM System z platform incorporates a variety of advanced autonomic computing capabilities. As discussed in "Introducing high availability and automation for SAP" on page xxi, 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 UDB for z/OS.

Features of the System z hardware architecture

Many of the System z RAS features were developed at a time when the failure of hardware elements was more frequent. Such failure is rare today, but these hardware availability features remain just as valuable. Most System z hardware elements have built-in redundancy or can be circumvented if they fail. The following table summarizes some of these features. For the features listed, the table shows which apply to the frequency, duration, and scope of an outage. It further explains whether this feature helps eliminate planned or unplanned outages, or both. For a complete and detailed explanation of the features, see the IBM publication *zSeries 900, System Overview*, SA22-1027.

Table 2. Selected System z availability features matrix

Availability feature	Reduces outage frequency	Reduces outage duration	Reduces outage scope	Planned outage	Unplanned outage
Alternate support element		X	X		X
Processing unit sparing	X				X

Table 2. Selected System z availability features matrix (continued)

Availability feature	Reduces outage frequency	Reduces outage duration	Reduces outage scope	Planned outage	Unplanned outage
System Assist Processor reassignment	X				X
Error correction code	X				X
Memory scrubbing	X				X
Dynamic memory sparing	X				X
LPAR dynamic storage reconfiguration	X			X	
Dynamic I/O configuration	X			X	
Concurrent channel upgrade	X			X	
Dual power feeds	X				X
Redundant power supply technology	X				X
Concurrent hardware maintenance	X			X	
Capacity upgrade on demand	X			X	
Concurrent licensed internal code patch	X			X	
Internal battery feature	X				X
X = applies					

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 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. It 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.

List of z/OS availability features

Following are some of the availability features provided by z/OS:

- **System integrity**

  z/OS system software has a total commitment to system integrity. Data and system functions are protected from unauthorized access, whether accidentally or deliberately with malicious intent. IBM provides an integrity warranty for z/OS that is unique in the industry. Without system integrity, there can be no assurance of security, and therefore reliability.
- **Dynamic operating system customization**

  To provide maximum continuous operation, many functions and major subsystems in z/OS are designed to be dynamically operated, configured, and tuned.
- **Storage key protection**

  This mechanism allocates a key to each program and piece of storage. It then ensures a match between the key of a program and the storage it is accessing. This protects the operating system from the applications it is running. It also protects the kernel of the operating system from the outer subsystems, and those subsystems from each other. Storage key protection also protects against storage overlays due to erroneous I/O operation. Without storage key protection, it would be impossible for IBM to provide the z/OS integrity warranty.
- **Low address protection**

  z/OS stores its most critical status and control information at the low end of each address space. This feature protects these vital areas from erroneous modification, even by the operating system kernel itself. A wide variety of potential system hangs are thus prevented.
- **Functional recovery routines**

  A major philosophical design guideline of z/OS is that every part of the operating system should include recovery facilities. These facilities handle both hardware and software errors by utilizing a specific Functional Recovery Routine (FRR) of the component of z/OS in control at the time of the error. The signal takes the form of an interrupt to a pre-specified FRR. The job of each FRR is to assess any damage, generate diagnostic information, and either repair the problem or remove the offending work unit from the system. Particularly serious problems beyond the scope of a specific FRR can percolate up to a higher level FRR, but a major attempt is made to isolate the error for minimal impact. If an error in z/OS itself occurs, therefore, FRRs will take over control, diagnose, and repair the failed parts of z/OS or isolate them from other parts of z/OS and other work in the z/OS system.
- **Isolation from outboard errors**

  z/OS has defenses against outboard errors that can impact the system. One example is called hot I/O detection. I/O devices can malfunction in such a way as to harass the server with extremely frequent unsolicited interruptions. When that happens, it can stop the system from processing any useful work. z/OS protects against hot I/O by fencing the device that is causing the problem.

Availability features and benefits with System z Parallel Sysplex

Sysplex Timer® and Server Time Protocol (STP)

A Sysplex Timer is used to synchronize all clocks in all System z processors connected to a System z Parallel Sysplex. It is a single point of control external to all processors that makes sure that all sysplex members use the same time source. This is necessary in order to guarantee the correct order of database updates. If the timer fails, the sysplex fails. The Expanded Availability Feature provides a second timer.

A new time synchronization feature, Server Time Protocol (STP), which is designed to provide the capability for multiple System z servers to maintain time synchronization with each other, is the follow-on to the Sysplex Timer. STP is designed for servers that have been configured to be in a Parallel Sysplex or a sysplex (without a Coupling Facility) as well as servers that are not in a sysplex but need to be time synchronized. It is designed to support a multisite timing network of up to 100 km (62 miles) over fiber-optics cabling.

Coupling Facility

The function of a Coupling Facility can be performed by a standalone specialized System z processor, or it can be operated within an LPAR of a standard System z processor. This Internal Coupling Facility allows the same functions to be performed within an LPAR, therefore reducing cost. However, at least two CFs are required for full redundancy. Splitting the work between two coupling facilities also allows a faster takeover, because only half the contents need to be rebuilt, not the complete contents.

Coupling Facility Link

A Coupling Facility Link is the hardware that forms the communication path between the coupling facility and the System z processor. It is essential that sufficient links be included for redundancy in this area.

Because the System z Parallel Sysplex is designed for continuous operation, you should make sure that your configuration follows the principle of avoiding single points of failure (SPOFs) to indeed achieve your availability goal. The key design point is that the configuration should tolerate a failure in any single major component. The same design concept of redundancy that applied to single System z systems therefore also applies to System z Parallel Sysplex. If at least two instances of a resource exist, a failure of one allows the application to continue. Sysplex Timer, Coupling Facility, and Coupling Facility Links are elements that should be duplicated.

List of System z Parallel Sysplex availability features

The following table summarizes the features which are implemented in the design of DB2 UDB 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 3. 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

Table 3. Parallel Sysplex availability features matrix (continued)

Availability feature	Reduces outage frequency	Reduces outage duration	Reduces outage scope	Planned outage	Unplanned outage
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 UDB for z/OS availability features with data sharing" on page 13
- **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 UDB 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 156.

  For details on DB2 UDB for z/OS, see "Features of DB2 UDB for z/OS" and in particular "List of DB2 UDB for z/OS availability features with data sharing" on page 13.
- **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 UDB for z/OS

DB2 UDB 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 UDB for z/OS to improve high availability and continuous operation of the database.

List of DB2 UDB for z/OS availability features

In this section we only discuss DB2 UDB 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 UDB for z/OS availability features with data sharing” on page 13.

The following table summarizes the features which are implemented in the design of DB2 UDB 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 4. DB2 UDB 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	
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
Fast log-only recovery due to more frequent HPGRBRBA updates		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

Table 4. DB2 UDB for z/OS availability features matrix (continued)

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

The main DB2 UDB 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. This is accomplished by identifying the minimum number of objects that need to be involved in the recovery process, which in turn reduces the time needed to restore the data and minimizes the amount of log data that needs to be applied. For DB2 systems that serve SAP systems and more than 30,000 tables, this enhancement significantly improves the data recovery time. Moreover, the BACKUP SYSTEM utility allows taking system-wide backups with unrestricted read and write activity of the SAP workload.

- **Suspend log write activity**

  The capability to suspend log write activity by issuing the command SET LOG SUSPEND has been available before DB2 V8. It allows taking fast volume-based backups outside the control of DB2. These backups can be the basis for a fast system-wide recovery.

- **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.

- **Online backup with SHRLEVEL CHANGE option**

  The use of the SHRLEVEL CHANGE option produces a fuzzy image copy during concurrent SAP workload. To recover to a point of consistency, DB2 applies the necessary log records. An important aspect of the online backup is the “incremental” online backup. This is a copy of only those tablespace data pages that have been changed since the last backup. Except for a small processor and DASD overhead, the online backup has no impact on the concurrent SAP activities.

- **CONCURRENT option in COPY**

If you perform an offline backup of a tablespace, concurrent write activity on this particular tablespace is not allowed. The usage of the option CONCURRENT, however, can significantly reduce the time the tablespace is unavailable for write activity. The database activity will be quiesced and made available again automatically. This method does not need the separate quiesce and restart steps.

- **CHANGELIMIT option in COPY**

  The CHANGELIMIT option in COPY allows DB2 to determine whether to take a full or incremental image copy, depending on the number of pages changed since the last image copy. With this option, you can avoid running image copy jobs when very few or no pages of a tablespace have changed since the last image copy was taken. The savings in time can be used to maximize the use of batch windows.
- **COPYDDN and RECOVERYDDN**

  The options COPYDDN and RECOVERYDDN allow you to create up to four identical copies of the tablespace.
- **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, which restores the image copy and applies log records.
- **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.
- **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.
- **Parallel COPY and Parallel RECOVER**

  DB2 allows you to specify a list of tablespaces and index spaces that you can copy and recover in parallel using the PARALLEL option of COPY and RECOVER. This feature enables a faster object-based backup and restore of DB2 because it reduces the elapsed time of these jobs.
- **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. The is controlled by the FASTSWITCH option of REORG.
  4. 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 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.
- **Partition independence**

  A key availability feature of DB2 UDB 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.

List of DB2 UDB 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 the scheduled removal of a member for service or a similar action. In each case, the service provided by the Parallel Sysplex is unaffected. SAP implements failover mechanisms that 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 UDB 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. 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 156.

DB2 UDB for z/OS improvements and new products

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 UDB 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.

To utilize light restart, an ARM policy needs to be put in place for DB2 that will specify the LIGHT(YES) keyword.

IBM DB2 Toolkits for z/OS: SAP Editions

IBM offers special DB2 toolkits that consist of sets of DB2 tools at a reduced price. These tools can be used to optimize the management of your DB2 subsystems and ensure high-availability characteristics. The following toolkits are available exclusively for SAP customers:

IBM DB2 Automation Toolkit for z/OS: SAP Edition

This toolkit consists of DB2 Automation Tool, a tool which advances the goal of enterprise autonomic computing and offers enhancements to ERP environments by providing capabilities such as automating the execution of copy, reorg, and other utilities to provide better management of system resources.

IBM DB2 Performance Toolkit for z/OS: SAP Edition

This toolkit consists of DB2 Performance Expert and DB2 Query Monitor, two tools designed to help keep DB2 running at peak levels even under heavy demand.

IBM DB2 Administration Toolkit for z/OS: SAP Edition

This toolkit consists of the DB2 Administration Tool and DB2 Object Comparison Tool, which work together to form a function-rich object management solution, providing quality assurance in an SAP ERP environment by comparing elements in SAP ERP database objects before they are transported from test to production.

For details on these special toolkits, see

`http://www.ibm.com/software/data/db2imstools/db2tools/db2sap/index.html`

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
- ICLI failure
- DB2 Connect instance failure

The IBM Redbook *SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations*, SG24–5690, describes tests for some of the failure scenarios listed above.

Chapter 2. DB2 data sharing on System z Parallel Sysplex

This chapter discusses the motivations for implementing an SAP-DB2 data sharing solution and 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 “Introducing high availability and automation for SAP” on page xxi.
- 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 tool 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 4 on page 18 introduces a high-level picture of the elements of a System z Parallel Sysplex environment.

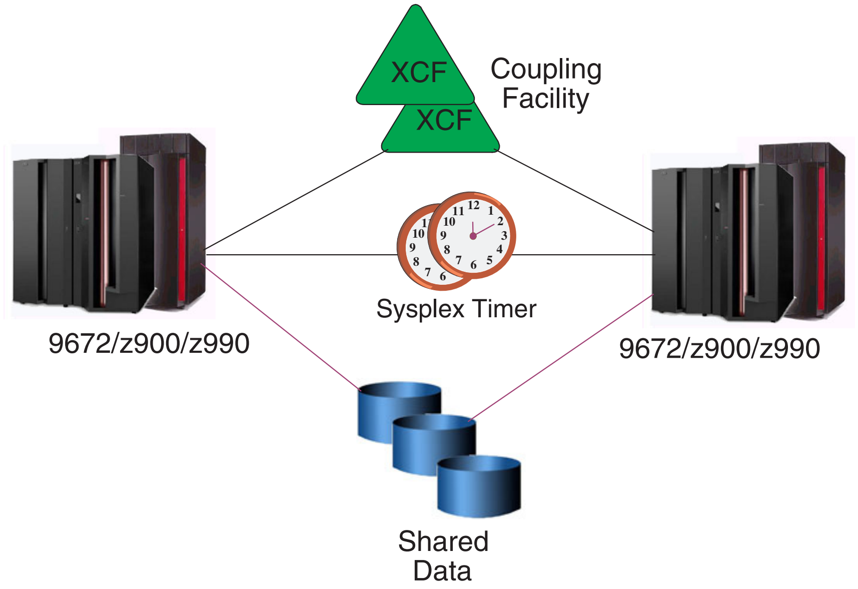


Figure 4. System z Parallel Sysplex architecture elements

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 special purpose System z computers known as 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, sysplex-wide shared data, and multiple high-speed, duplexed links connecting the components. This implementation employs hardware, software, and microcode.

DB2 data sharing architecture

The following figure completes the picture by laying multiple DB2 data sharing members on top of the Parallel Sysplex infrastructure.

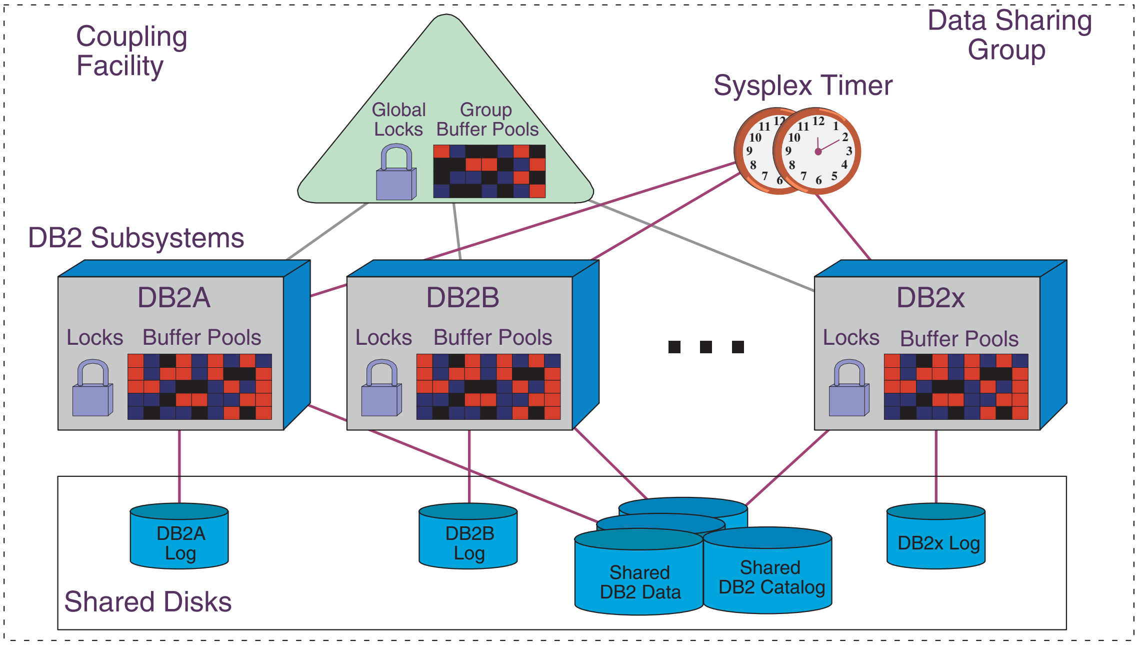


Figure 5. DB2 data sharing in a Parallel Sysplex

In this picture we see up to 31 DB2 subsystems (DB2 members) making up a DB2 data sharing group. There is one DB2 data sharing group for each production SAP system (an SAP system instance is also known as an SAP ID or SID). 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 UDB for z/OS are explained in more detail in the
| DB2 document *Data Sharing: Planning and Administration*.

SAP sysplex failover architecture

We complete the DB2 data sharing infrastructure picture with an SAP customization known as SAP sysplex failover. The following figure introduces the basic building blocks of an SAP sysplex failover architecture.

DB2 data sharing

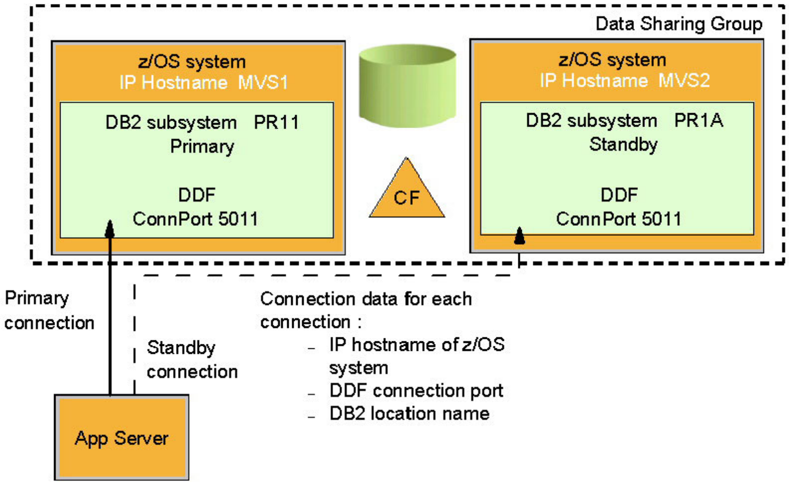


Figure 6. SAP sysplex failover configuration: Option 0 example

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 3, "Architecture options and trade-offs," on page 23.

The three-character SAP instance name (SID name) is illustrated in the figure. We use the SID name (for example, PR1) 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, PR11), and the standby DB2 member names will be the DB2 group name plus a letter (PR1A).

This figure illustrates a three-tiered implementation in which each application server has a primary DB2 member in one LPAR (for example, MVS1) and a standby DB2 member in a standby LPAR (MVS2). DDF is an integral part of a DB2 subsystem. With older releases that still use ICLI, an ICLI server needs to be available for each DB2 member.

Each application server has an SAP control file known as its *instance profile*, which contains the z/OS LPAR IP host name, connection port, DB2 member name (PR11) of the primary DB2 member, as well as the same information for the standby DB2. In the event of a planned or unplanned incident, the SAP Database Services Layer (DBSL) in the application server recognizes the need to fail over, looks for standby information in the instance profile, and connects the application server to the standby DB2 member. Starting with SAP 6.10, this information is no longer included in the instance profile but instead in a general file for all application servers, the `connect.ini` file. For details on connect.ini, see the *SAP Database Administration Guide.*

Chapter 3. Architecture options and trade-offs

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
- ICLI design

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 SAP application servers to another DB2 member at a specific time.

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

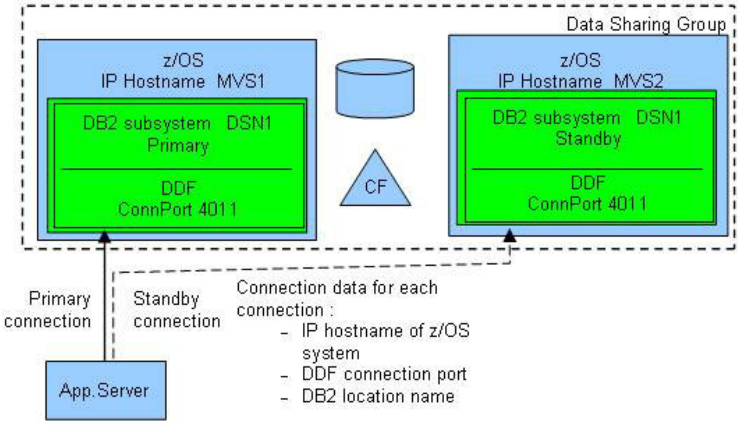


Figure 7. Option 0: Single DB2 member with passive (inactive) standby member

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.

Under normal conditions (with every component working properly), the passive DB2 member (and associated ICLI server for older SAP releases) 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 transaction ST04 provides. 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.

For ICLI only

You can also look for the CPU activity on the standby ICLI server address spaces. Another method that is more passive is to direct ICLI server messages to the z/OS system console. This is accomplished by setting the environment variable ICLI_WRITE_TO_SYSLOG to 1. In the event of a failure, the application servers connect to standby ICLI servers and ICLI server message ICLS1300I is sent to the z/OS console. The database attach name is included in the ICLI server message or in the SAP connection profile `connect.ini`. For this method to work, it is important to identify the DB2 member name in SAP profile parameter dbs/db2/ssid.

End of For ICLI only

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. In order to use this functionality, ICLI must not be used to collect DB2 performance data via the DB2 IFI interface. Instead, RFCOSCOL or SAPCL must be used, depending on the SAP release. See SAP Notes 426863 and 509529 for details.

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

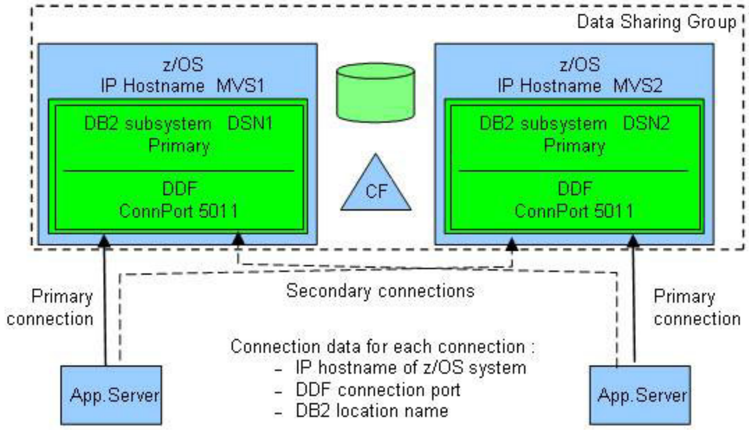


Figure 8. Option 1: Two active DB2 members in active-standby mode

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, SAP sysplex 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.

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 OS/390 and 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 services 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.

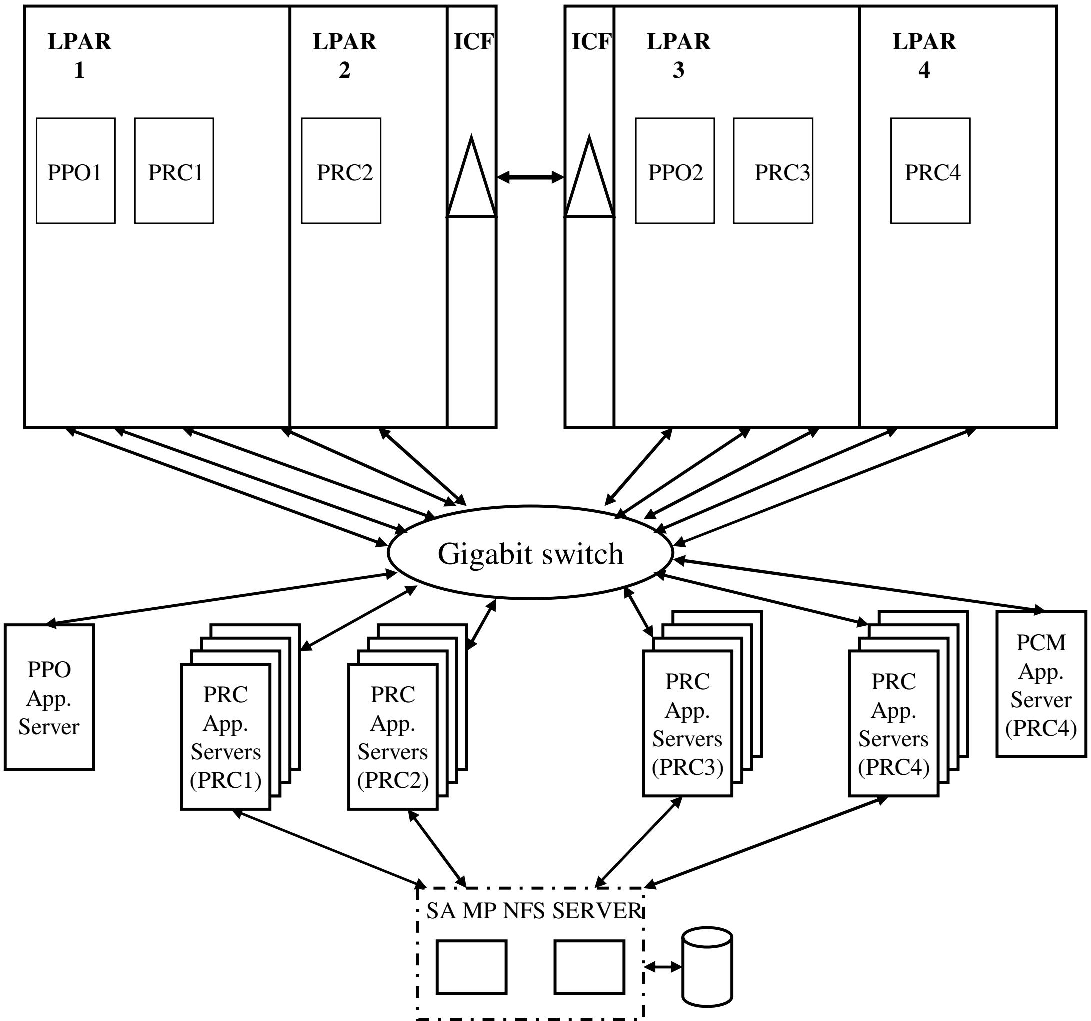


Figure 9. Large company using architecture options 0 and 1

Architecture options

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:

- R/3 Enterprise 4.7 (<sapsid> PRC)
- SCM Release 5.0 (<sapsid> PP0)
- CRM Release 4.0 (<sapsid> PCM)

R/3 and CRM are in an MCOD environment.

R/3 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.

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 R/3, 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

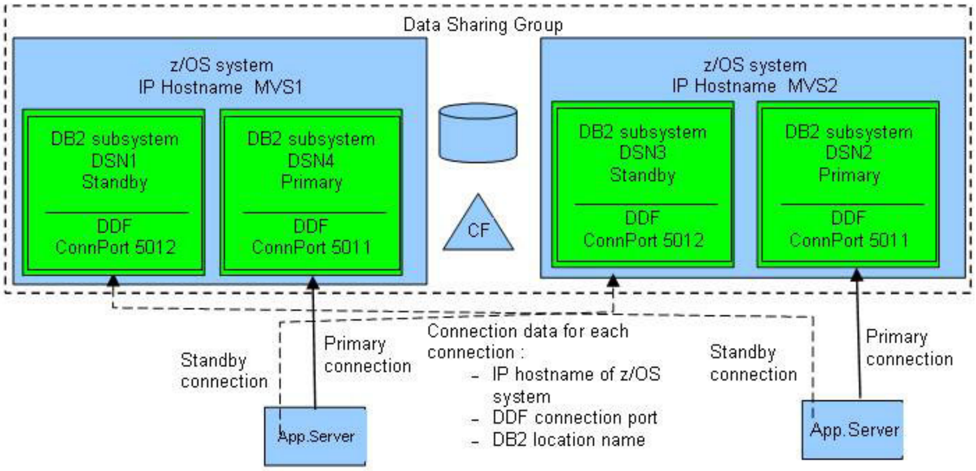


Figure 10. Option 2: Two active DB2 members, each with a passive standby member in the same LPAR

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.

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

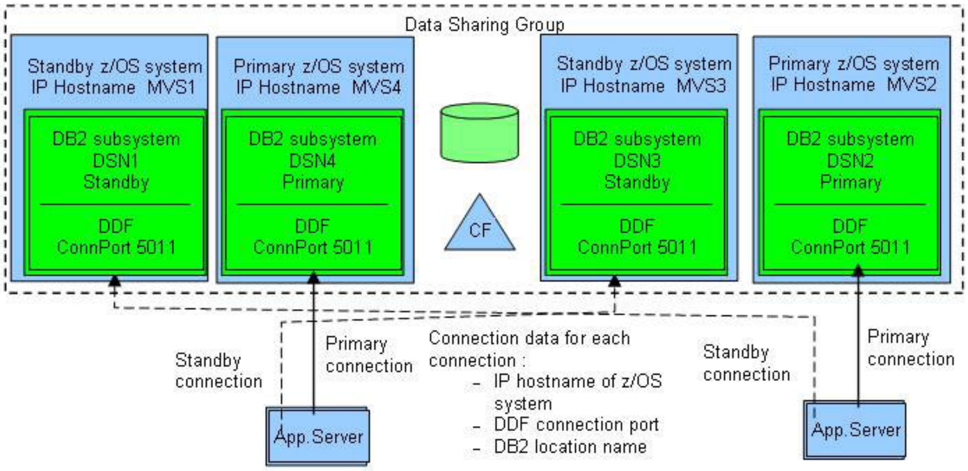


Figure 11. Option 3: Two active DB2 members, each with a passive standby member in an independent LPAR

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 z900 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 Information Warehouse (BW), 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. In DB2 V7 and previous releases, keep in mind that there is a 2 GB limit on the addressable memory in the DBM1 address space. In DB2 V8, some of the storage structure, such as thread storage, still resides 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 UDB 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). 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) and 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 5. 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.

High availability considerations with DB2 Connect and DDF

With DB2 V8 and SAP Web Application Server 6.40, standard DRDA-based database connectivity replaces ICLI. DB2 Connect replaces the ICLI client. The DB2 DDF address space implements the server role. There is no need for a separate component like the previously used ICLI server.

To ensure high availability of the DB2 Connect instance that a SAP application server employs, it automatically starts the instance if it is not already running. The counterpart of DB2 Connect, the DDF address space, is an integral part of the DB2 subsystem. Therefore, it is sufficient to have a single DDF address space per DB2 subsystem.

If you run multiple SAP application servers on the same operating system image, it is recommended to have a dedicated DB2 Connect instance for each application server.

For SAP Web AS 6.20 running SAP Kernel 6.40 and DB2 V8, ICLI is replaced by DB2 Connect.

For migration to DB2 V8 and prerequisites, refer to SAP Note 728743. Via this note, SAP will keep you informed whenever additional SAP technology is enabled to run on DB2 V8. For further information about possible upgrade paths involving DB2 V8, see *SAP on IBM DB2 DB for OS/390 and z/OS: Best Practice for Installing or Migrating to DB2 V8*, available in the SAP Service Marketplace under:

`http://service.sap.com/solutionmanagerbp`

ICLI design

For ICLI only

If you still run the ICLI client/server product in your SAP HA environment, you should read the following in order to create an ICLI server setup suitable for your HA requirement.

The Integrated Call Level Interface (ICLI) is an IBM feature shipped with z/OS that provides connectivity between the SAP application server and the DB2 database server. It can be described as a black box that delivers requests from the SAP application servers to the DB2 database server and returns responses from the DB2 database server to the SAP application server. All communications are initiated on the application server side.

ICLI is used only when the SAP application server runs on AIX, various versions of Windows or Linux on System z. Said another way, it is used for all application servers that are remote with respect to the DB2 database server. The application server that runs on z/OS UNIX System Services (USS) uses cross-memory services and ODBC to communicate with the DB2 database server.

The ICLI is made up of the ICLI client and the ICLI server. The ICLI client runs where the SAP kernel and Database Services Layer (DBSL) run (such as AIX) and the ICLI server runs in the LPAR where the DB2 database server is running. Each application server instance in your system landscape runs its own instance of the ICLI client. Each of these ICLI client instances can communicate with a single ICLI server instance or multiple ICLI server instances. There is no limitation on the number of ICLI server instances that can be started in one LPAR. The only requirement is that each ICLI server must have its own port number to distinguish it from the others.

The SAP kernel directs the ICLI client via the DBSL to connect to a particular ICLI server. The kernel does this by passing the host name and port number where the ICLI server is running. So an SAP kernel can communicate with any ICLI server anywhere as long as it can be reached using TCP/IP. The other parameter that is sent in the connect request is the DB2 subsystem ID.

The SAP kernel has an additional function that enables it to provide two or more sets of parameters so that it can communicate with more than one ICLI server. This connectivity flexibility known as sysplex failover is ideally suited for configuring high availability environments. Note that one ICLI client can talk to only one ICLI server at a time.

How many ICLI servers?

So how do we best configure our ICLI servers to take advantage of the connectivity function and use this sysplex failover function to implement a highly available SAP system and not impact system performance?

For the high availability requirement: As with most high availability systems, the key is to eliminate single points of failure. We can accomplish this with ICLI servers by starting more than one ICLI server per SAP system in order to divide the workload.

To examine how this would work with data sharing option 1, assume that we have two application servers connected to our data sharing system and that each one supports a different SAP module, such as SD and FI. We could configure both application servers to connect through a single ICLI server, but this makes the single ICLI server a single point of failure. To enable the SD users to work independently of the FI users, it would be better to configure two ICLI servers so that a failure of one ICLI server will not affect the connections established through the other ICLI server. This way either the SD users or FI users would still be productive if only one ICLI server fails.

Suppose that the number of SD and FI users is large enough that each module requires more than one application server to support the respective workload. We could configure the two SD application server instances to connect through the one ICLI server assigned to handle SD workload, and we could configure the two FI application server instances to connect through the ICLI server assigned to handle FI workload. It is still true that the SD users are isolated from the FI users, but now you have more of each. Even if only one ICLI server fails, fewer users are sitting idle until the ICLI server problem is resolved and it is restarted.

There is no real limit to the number of ICLI servers that can run in one LPAR, so we start one ICLI server per application server instance. Now the failure of one ICLI server would temporarily affect a smaller percentage of the user community and all business functions could still be operated. All of the users who were forced off during the failure would log on again, and the SAP logon load balancing function would direct the logons to the surviving application server instance that is capable of connectivity to the database server.

End of For ICLI only

Chapter 4. Backup and recovery architecture in data sharing

In this chapter, we discuss the backup and recovery issues a DB2 installation must consider when moving from a non-data sharing to a data sharing environment. 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 also discuss Disaster Recovery and Homogeneous System Copy in an SAP environment.

This chapter includes the following sections:

- Data sharing backup/recovery considerations
- Disaster recovery considerations
- Homogeneous system copy considerations

Data sharing backup/recovery considerations

In this section we start with a brief description of the recovery environment introduced by DB2 data sharing. We cover how DB2 data sharing recovers data, we analyze the modifications to the current models of backup and recovery procedures that apply to SAP databases when moving from non-data sharing to data sharing, and we describe the possibilities of DB2 Version 8 that address the SAP recovery requirements in a data sharing environment.

DB2 data sharing introduces features that enable database recovery from failures across multiple DB2 data sharing members:

- Log record sequence number (LRSN)
- Logical page list (LPL), which is also valid for a non-data sharing environment
- Group buffer pool recovery pending (GRECP)
- SCA structure
- LOCK structure
- Damage assessment processing (DAP)

These features are introduced to cover data sharing requirements. DB2 uses all of these features at recovery time, and they have changed the way DB2 performs recovery from different failures in a data sharing environment.

Data sharing recovery environment

In a data sharing environment, the member subsystems maintain separate recovery logs. Each manages its own active and archive log data sets and records those in its own bootstrap data set (BSDS). The shared communications area (SCA) in the coupling facility contains information about all members' BSDSs and log data sets. In addition, every member's BSDS also contains information about other members' BSDS and log data sets in case the SCA is not available.

In accordance with other operational procedures, the changes introduced by DB2 data sharing to database recovery are mostly internal to DB2, and they have little impact on the existing tablespace recovery procedures. However, the scope of the recovery process is at a data sharing group level, and updates made by all members must now be considered. Consequently, DB2 has to process logs from all members, and it must be able to sequence updates to a single page across all members in the DB2 data sharing group. The recovery process can be performed on any member.

Therefore, it is clear that relative byte addresses (RBAs) cannot be used to sequence the log records from multiple members of the data sharing group. Each DB2 member has its own log. The RBA of a member has no relationship with the RBAs of other members. The rate at which log RBAs are advanced is related to the intensity of updates occurring at the individual member. For example, some members in a data sharing group can only be used for special purpose, such as parallel batch runs or query support. Their log RBAs are likely to lag behind the RBAs of other members, where updates occur all the time.

The LRSN is used to sequence log events such as page set updates from different members. RBAs are still used within a single member. The LRSN is based on the time of day, which is obtained using the STORE CLOCK instruction. It is a 6-byte value that is equal to or greater than the time-of-day time stamp value truncated to 6 bytes. When a page is updated in a data sharing environment, the LRSN is stored in the page header. Because members generally run on different machines, they must have synchronized time and therefore consistent LRSN values. This function is provided by the Sysplex Timer or STP.

It is important to keep in mind that a unit of recovery (UR) can execute on a single member. All log records related to one UR are written to the log of the member where the UR executed. A single UR cannot write some of its updates to one member's log and other updates to another member's log, although different URs, executing on different members, can concurrently update the same page set or partition.

In a data sharing environment, the DB2 catalog and directory are shared among all members. This means that every new DDL or bind originated by one member is immediately visible for the rest of the members. Any tablespace or partition can be recovered from any member. However, the RECOVER utility executes on a single member.

As in a non-data-sharing environment, key information for recovery is stored in the DB2 catalog and directory, in the bootstrap data sets, and on the log. The DB2 catalog and directory are expanded to track member-specific information and to achieve log synchronization. In a data sharing group each member maintains its own active and archive logs. A member must have read access to other members' BSDSs and logs, but they cannot write to them.

An important aspect to consider when enabling data sharing in a DB2 environment is the media used to store archive logs. Recovery processes that require archive logs from multiple members must allocate these archive data sets simultaneously. This means, in case those data sets are stored on tape, that we need the same number of tape units available to perform the recovery. This can be a source of problems in recovery situations.

Note: For data sharing it is recommended, not only for availability but also for performance reasons, to avoid using tape archive logs for data recovery.

For more about the DB2 data sharing recovery environment, see the DB2 publication *Data Sharing: Planning and Administration*.

Next, we discuss two kinds of data recovery situations that can happen with data sharing:

- Tablespace recovery
- Recovering pages in the logical page list (LPL)

Tablespace recovery

The procedures for data recovery are fundamentally the same for data sharing as for non-data-sharing. Data sharing involves one catalog, but there are now many logs and BSDSs. In addition to disk and cache controllers, a new medium is introduced: the coupling facility. This adds a possible point of failure and requires appropriate recovery procedures. In planning for data sharing, it is important to consider having more than one coupling facility. Should a structure failure occur, recovery for the SCA and LOCK structure can proceed automatically if a second coupling facility is available.

As in non-data-sharing, full image copy is used as the base for the tablespace recovery. When a member performs recovery of an object, it will review all SYSLGRNX entries from all members for the object being recovered. Working in data sharing mode, SYSLGRNX contains starting and ending LRSN values in addition to RBA values for each member and the member ID to which each range belongs.

DB2 can access the logs from other DB2 systems in the group and merge them in sequence. The log record sequence number (LRSN) uniquely identifies the log records of a data sharing member. The LRSN is always incremented for log records that pertain to the same page. There are never duplicate LRSNs for the same page, but LRSNs may be duplicated in log records of members on different pages.

The following figure illustrates how database recovery works in a data sharing group. Each member's log participates in the recovery process. The member that performs the utility gets access to the group log environment.

Backup and recovery

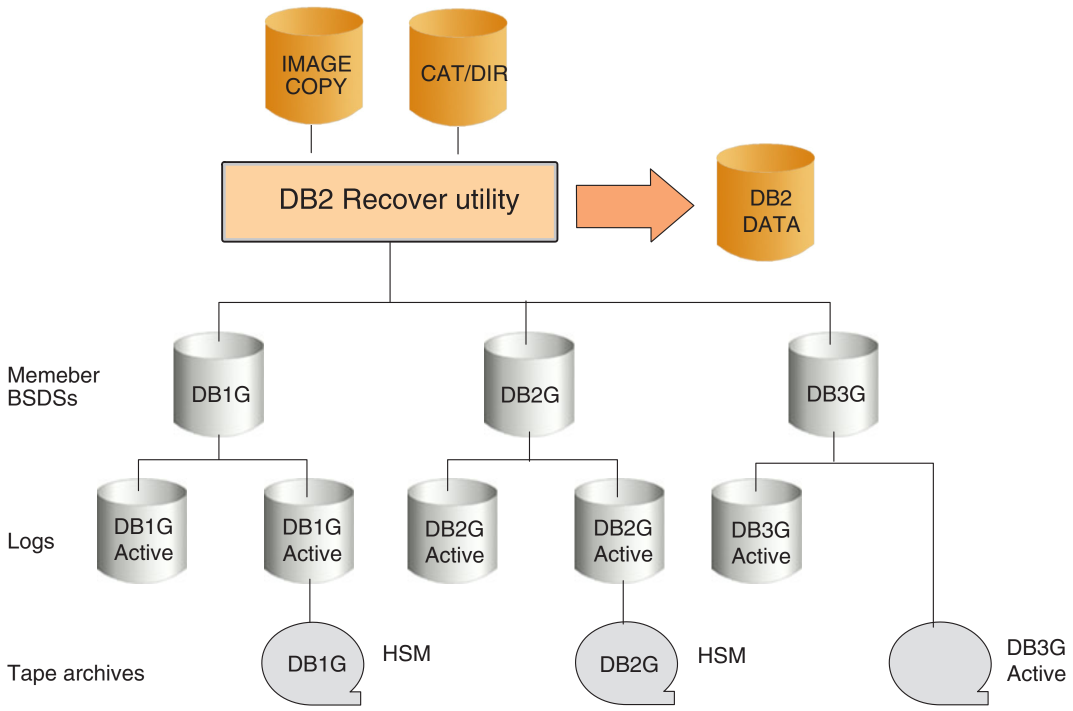


Figure 12. Database recovery in a data sharing group

The efficiency of the log apply process can be greatly enhanced using the fast log apply (FLA) feature. This feature appeared in Version 6, and it is also used in data sharing during DB2 restart and during START DATABASE for LPL and GRECP recovery. The process is able to sort log records so that pages that are to be applied to the same page or same set of pages are together. Then, using several log apply tasks, it can apply those records in parallel.

In order to enable fast log apply you must provide enough storage using the LOG APPLY STORAGE field of installation panel DSNTIPL. This storage is allocated in the DBM1 address space. If virtual storage is not a problem in the DBM1 address space, it is recommended during recovery to increase the storage available to fast log apply by setting the ZPARM parameter LOGAPSTG to the maximum of 100. In DB2 V8, the default value of LOGAPSTG is 100 MB, and it is recommended not to change this value.

When operating in a data sharing group, ability to recover can be hindered by a failed member holding retained locks. You must remove the retained locks by restarting the failed member before you can proceed with page set recovery. New restrictive states such as LPL and GRECP can have an impact if you perform a logical partition recovery on a nonpartitioning index, so you must remove them first. The recovery process relies on applying changes to a page in the same sequence as they originally occurred. Over a period of time, the same page is likely to be updated by URs running on different members. All changes are externalized to the member’s log at commit time. In support of data sharing, log records are expanded with new fields in their headers.

More information about data recovery in data sharing and log considerations can be found in the DB2 V8 document *DB2 UDB for z/OS Data Sharing: Planning and Administration*.

Recovering pages on the logical page list

DB2 responds to transient disk read and write problems by placing pages in the LPL. If DB2 cannot determine the reason for a page read or write error, the page is recorded in the LPL. In a data sharing environment, the LPL also contains pages that could not be read or written for must-complete operations because of some problem with the group buffer pool.

Typically only write problems result in LPL pages. Read problems typically result in resource unavailable conditions. The only time a read problem can result in LPL pages is when the read fails during must-complete processing. The read or write operation mentioned can be to DASD or to the group buffer pool.

The LPL is maintained for each data set of a page set; indexes or partitions have separate lists. The LPL is kept in the database exception table (DBET) in the SCA. For fast reference, the DBET is also cached by each member. When adding pages to the LPL, they are logged as non-UR-related DBET REDO log records.

Some common situations that result in adding entries to the LPL are:

- Error for must-complete operations

  DB2 finds errors when reading a page during restart or rollback processing.
- Force at commit write failure

  After a commit, DB2 must write updated pages to the group buffer pool. In case of problems the page is placed in LPL.
- Group buffer pool castout read failure

  The group buffer pool castout process reads updated pages from the group buffer pool and writes them to DASD. If the read request fails, the requested pages are added to the LPL.
- Restart with DEFER option

  When a member fails while holding pages pending to write to DASD or unresolved units of recovery, DB2 needs access to the data set to apply the changes during restart. If you restart this failing member, and the restart option for the page set indicates the DEFER option, the pending pages are converted to LPL entries.

DSNB250E is the message that DB2 issues when adding a page to the LPL. This is an important message that should be caught in the system and analyzed. Apart from taking the appropriate action to resolve, it can reveal other important problems.

In some cases, DB2 can automatically recover pages on the logical page list when group buffer pools are defined with AUTOREC(YES), the default. However, there are many situations where pages are put on the LPL that require you to do manual recovery. There are several ways to do this:

- Start the object with access (RW) or (RO). This is the most common method of recovery, and in most cases all that is required.
- Run the RECOVER utility on the object. This method should be used when simply starting the object does not successfully recover the LPL pages.

- Run the LOAD utility with the REPLACE option on the object. This assumes an acceptable copy of the object exists that is current and consistent with the other data objects in regard to application referential integrity.
- Issue an SQL DROP statement for the object. This assumes that the object is no longer needed or can be recreated.
- Use the utility REPAIR SET with NORCVRPEND. This can leave your data in an inconsistent state.
- Use START DATABASE ACCESS(FORCE). This can leave your data in an inconsistent state.

DB2 V8 brings two enhancements for performing LPL recovery:

- Automatic recovery of LPL pages: To avoid manual intervention for LPL recovery through the START DATABASE command or the RECOVER utility, in most cases DB2 automatically initiates an LPL recovery processor to recover pages as they are added to the LPL.
- Less-disruptive LPL recovery: The LPL recovery processor (by way of the START DATABASE command or the automatic LPL recovery feature), makes a write claim instead of a drain on the object that is being recovered. As a result, good pages in the object are available to SQL users, and performance is improved because the claim is less disruptive than a drain. In Version 7, the whole tablespace is inaccessible during the recovery.

None of the items in this list works if there are retained locks held on the object. You must restart any failed DB2 that is holding those locks.

Data sharing impact on SAP recovery procedures

We must consider some modifications in order to prepare SAP database recovery procedures to run in a data sharing environment. You will find a good summary of the required changes for DB2 V7 in SAP Note 83000, *DB2/390: Backup and Recovery Options*.

This note describes the backup and recovery options that should be implemented in the SAP on DB2 for z/OS environment with DB2 V7. The backup and recovery options with DB2 V8 are described in the *SAP Database Administration Guide* for SAP NetWeaver '04 (SAP Web Application Server 6.40) or higher. The note and the latter description are valid for both the data sharing mode and the DB2 normal mode.

It is not in the scope of this book to discuss extensively all possible backup and recovery scenarios for SAP on DB2 environments. In the following scenarios we discuss only the main issues affecting backup and recovery as you move your SAP system to data sharing:

- Object-based backup: online and offline
- Volume-based backup: online
- Establishing a group level point of consistency
- Recovery to any prior point in time

Object-based backup: online and offline

This option has no specific data sharing considerations. Full or incremental image copies with SHRLEVEL CHANGE for backup online (concurrent read/write access to the data), or SHRLEVEL REFERENCE for backup offline. Read access to the data can be scheduled by any member of the group during the offline backup, which meets the requirements of the installation.

From an availability point of view, it is recommended that you schedule DB2 administration utility jobs on multiple members. If all of the DB2 administration utility jobs run on one member, the catalog table SYSUTILX will never get inter-DB2 R/W interest and, as a consequence, this member will get an exclusive P-lock at a page set level. If this member fails, this exclusive P-lock will be retained by this member until it is restarted. This will cause a resource unavailable condition on other active members.

For this reason, the Group Attachment Name (GAN) support for generic access to the DB2 members becomes invaluable. Combining GAN support with a Workload Manager Batch Scheduling environment could be established to distribute the DB2 administration utility jobs (for example, jobs generated using SAP transaction DB13) on the most available DB2 member, according to LPAR resources availability and performance objectives.

As a reminder, we offer a list of other considerations that apply to running utilities in data sharing environments:

- DISPLAY UTILITY is a group scope command.
- A running utility can only be terminated on the same running MVS™ image. A stopped utility can be terminated from any active member in the data sharing group.
- A stopped utility can be restarted in any member of the group.

Online volume-based backup without the BACKUP SYSTEM utility

Online volume-based backups require availability of a disk subsystem capable of generating very fast volumes copies. Many options are available from different disk vendors. One of the options is FlashCopy® with the IBM Enterprise Storage Server® (ESS). Prior to DB2 Version 8, these backups are not registered in DB2. In order to obtain a consistent copy of all volumes containing DB2 system or data objects, the copy must be taken after DB2 update activity has been suspended. This is accomplished with the DB2 command SET LOG SUSPEND, which suspends the log.

In data sharing terms, this command is only member scope, which means that the command only effects the one member on which the command has been issued. Therefore, to achieve a real suspension of the update activity across the whole data sharing group, it is necessary to issue this command on all active members of the group.

Prior to taking the volume-level copies, it is important to ensure that particular statuses and activities do not exist in the DB2 data sharing group in order to avoid delays at restart time following a recovery action of the database. In particular, no utilities should be active, no page sets should be in a restricted status, and no long-running units of recovery (batch without frequent commits) should be running during the backup process. In the case of a running utility or a page set in a restricted status, there may be a recovery of one or more objects required after system restart. If a long-running unit of recovery was running during the volume-level copy, then backout processing may extend the system restart time. The installation parameters URCHKTH and URLGWTH, used in conjunction with LOGLOAD or CHKFREQ, detect long-running units of recovery and issue warnings of a workload not committing in the established period of time for your installation.

DB2 infrequently updates the HPGRBRBA (high page recovery base relative byte address), which is the starting point of object-based log-only recovery, of certain DB2 catalog and directory objects. The update process that is controlled by parameter DLDFREQ does not apply to these objects. These tablespaces are DSNDB01.SYSUTILX, DSNDB01.DBD01, DSNDB01.SYSLGRNX, DSNDB01.SCT02, DSNDB01.SPT01, DSNDB06.SYSCOPY, DSNDB06.SYSGROUP, and their associated IBM defined indexes with attribute COPY YES. Moreover, these objects do not have entries in directory table SYSIBM.SYSLGRNX, which limits the log range that needs to be scanned during recovery.

To ensure that the HPGRBRBA of these objects can be updated more frequently, DB2 V7 APAR PQ79387 introduces the following enhancement to the COPY utility. When taking a copy of the special catalog and directory objects listed above using the option SHRLEVEL(CHANGE), DB2 updates their HPGRBRBA. Therefore, in DB2 V7, if you intend to recover these objects based on a volume-based backup with option LOGONLY recovery, you should take image copies with the option SHRLEVEL(CHANGES) for these objects prior to taking the volume-based backup. This does not apply if you recover a system using the RESTORE SYSTEM utility introduced in DB2 V8.

An alternative way to advance the HPGRBRBA of the special catalog and directory objects is to submit the QUIESCE utility for each of these seven objects before you issue -SET LOG SUSPEND. There is special code in QUIESCE that updates the HPGRBRBA of these objects. The job should be composed of seven steps, because its purpose is not to develop a common recovery point but rather only to update HPGRBRBA of the objects. If each object is quiesced in its own step, there is a high probability of success in obtaining a DRAIN lock, which then allows the QUIESCE utility to advance the HPGRBRBA. This method provides for the minimum log scan if it is run just prior to the volume dumps.

Issues with HPGRBRBA can be circumvented by recovering the special catalog and directory objects based on image copies. This is usually much slower, however, and can involve mounting tapes.

In a non-data-sharing environment, issuing SET LOG SUSPEND would trigger a DB2 system checkpoint. This is not the case in data sharing. In order to force a DB2 member in the data sharing group to perform checkpoint processing, the command SET LOG LOGLOAD(0) must be issued. It is important that checkpoint processing take place prior to issuing the SET LOG SUSPEND in order to externalize the DB2 data buffers to DASD. This is especially important for page sets with 32K pages because it takes multiple physical I/Os to externalize their pages if their page size does not match their CI size. This lowers the probability of a 32K page write only being partially complete during the volume-level copy. Again, this reduces an eventual restart delay. The commands SET LOG LOGLOAD(0) and SET LOG SUSPEND are not group-level commands so they must be issued to each member of the group. There will be pending writes in group buffer pools that will not be externalized to DASD. This is not a problem and will be handled during DB2 group restart.

DB2 V8 allows page sets with 32 KB pages to have 32 KB CIs. This eliminates intra-page inconsistencies with 32 KB pages.

Following are the actions that DB2 initiates with the SET LOG SUSPEND command in a data sharing environment:

1. Force out log buffers.

2. Update the high-written RBA in the BSDS.
3. Hold the log-write latch to suspend updates to the log output buffers.
4. As of DB2 V8, the following additional actions are taken:
   a. Record the recovery base log point (RBLP) in DBD01 (this enables backups taken during log suspension to be used for RESTORE SYSTEM).
   b. Quiesce 32 KB page writes for objects with a CI size of 4 KB.
   c. Quiesce data set extends.
5. Echo back high-written RBA and last system checkpoint RBA in a DSNJ372I message.

Whenever a SET LOG SUSPEND is issued on a DB2 system, upon successful completion the following message is written to the LPAR syslog, DB2 MSTR message log, and the console:

```
*DSNJ372I  -DB7X DSNJC09A UPDATE ACTIVITY HAS BEEN  606
 SUSPENDED FOR DB7X AT RBA 0008E300CBD5, LRSN 0008E300CBD5, PRIOR
 CHECKPOINT RBA 0008E2EEA6A6
 DSN9022I  -DB7X DSNJC001 '-SET LOG' NORMAL COMPLETION
```

Keep in mind that volume-level backups are of no use unless it is certain that update activity throughout the DB2 data sharing group has been suspended. Therefore, it is recommended that an automated procedure be put in place to guarantee that this message has appeared for all active members in the group before starting the backup.

We recommend setting up an automated process for the whole backup procedure in order to follow these steps in all of the active members of the group. The specific IBM product for this implementation is Tivoli System Automation for z/OS.

After successful execution of the command, in each member:

- Shared reads are allowed.
- Updates are not allowed.
- Buffer pool contents are not flushed.
- Group buffer pools (GBP) are not flushed.
- Write I/Os and castouts are still allowed.

An SAP end user will notice after triggering this command that saving data takes more time than expected, but querying data proceeds as usual.

At all times we can verify the log activity in DB2 with the DIS LOG command. If log activity is suspended, the following output appears.

```
-dis log
DSNJ370I =DBK4 DSNJC00A LOG DISPLAY
CURRENT COPY1 LOG = DB2V610K.DBK4.LOGCOPY1.DS01 IS 9% FULL
CURRENT COPY2 LOG = DB2V610K.DBK4.LOGCOPY2.DS01 IS 9% FULL
H/W RBA = 000000A0D662, LOGLOAD = 100000
FULL LOGS TO OFFLOAD = 0 OF 6, OFFLOAD TASK IS (AVAILABLE)
DSNJ371I =DBK4 DB2 RESTARTED 09:37:59 APR 21, 2000
RESTART RBA 00000001D000
DSNJ372I =DBK4 DSNJC00A UPDATE ACTIVITY HAS BEEN SUSPENDED FOR DBK4
AT RBA 000000A0D662
DSN9022I =DBK4 DSNJC001 '-DIS LOG' NORMAL COMPLETION
```

Now, the fast volume copy must be invoked from one of the systems. In case of FlashCopy, you can use DFSMSdss™, TSO, Web interface, or script.

For this kind of backup, it is very important to have volume independence between all of the components of the DB2 subsystem. In other words, the active logs and BSDSs of all DB2 members should be on volumes separate from any DB2 directory and catalog objects (VSAM data sets) or SAP DB2 objects (VSAM data sets). Different ICF catalogs should be created for the DB2 system data sets and objects and the SAP DB2 objects. These ICF catalogs should be on one of the volumes with either the DB2 directory and catalog objects or the SAP DB2 objects. This automatically includes them in any volume-level copies. This is even more important if you want to implement PIT recovery without restoring the LOG and BSDS data sets from the volume-level backup.

After the volume-level copy has finished, the updating activity must be resumed in each member of the group using the SET LOG RESUME command.

SET LOG RESUME will:

1. Resume logging and update activity
2. Release log-write latch.
3. Issue the DSNJ373I message.
4. Delete the DSNJ372I message from the console.

Establishing a group-level point of consistency

Getting a point of consistency in the database is becoming less important due to the fact that point in time recovery must be set at a system level (the whole database is restarted at an specified LRSN and in this way DB2 gets consistency) and because a point of consistency implies a degree of unavailability. Even so, for an installation that can afford this cost, getting a point in which the database is consistent may help in certain situations. Dealing with full image copies obtained with share level change (in which case you cannot recover TOCOPY), it is good to have a daily point where you know that all of your data is committed.

In a data sharing environment, it is even more difficult to get a point of consistency because with all members sharing the same database, this must be a coordinated situation.

There are different ways of getting the database quiesced:

- ARCHIVE LOG MODE(QUIESCE) TIME(n)
- QUIESCE utility
- START DATABASE ACCESS(RO)
- STOP DB2 MODE(QUIESCE) in all members

Any of these methods also work with data sharing. The first method is preferred. Issuing this command from one of the members provokes all of them to start waiting for all transactions or jobs to commit and draining new units of recovery in the group.

As in non-data sharing, it is recommended that you set the TIME parameter of this command just below the IRLM timeout parameter to avoid cancelling transactions.

Recovery to the current state

Normally, we recover to the current state after some set of data has been damaged, leaving a number of tablespaces in an unavailable state. Typical examples are DASD problems or a failed reorganization. The RECOVER utility discussed earlier in this chapter is usually used. If you are not recovering to current and you need to recover to a prior point in time, you cannot specify TORBA for a PIT recovery. In that case you have to look for an LRSN and specify TOLOGPOINT in the utility control statement.

Most often this will not be the case because recovering a subset of tablespace to a previous point in time, leaving the rest of the database in the current state, goes against the SAP requirement of considering the whole database as a consistency unit of recovery.

Recovery to a previous point in time using DB2 V7

When planning for this kind of recovery after moving to data sharing, there are some important points to consider.

The main consideration applies when enforcing a point of consistency to the database with a conditional restart to an arbitrary prior point in the current log or when restoring the whole volume-based backup of the DB2 environment including logs and BSDSs.

First, be consistent within the group. All members of the data sharing group must be restarted to the same point in time to ensure that the database is left in a truly consistent state.

In data sharing, use an LRSN as a common point of conditional restart for all the members instead of an RBA. The DSNJU003 utility enables creation of a conditional restart control record (CRCR) that specifies an LRSN as a parameter. A CRCR should be defined for each member as in:

```
CRESTART CREATE ENDLRSN=0008E300CBD
```

In order to find a valid LRSN, in most installations it is possible to convert a time stamp into STCK format. In some cases enabling data sharing introduces a delay in the LRSN, so it does not match the time stamp. In any case, it is always possible to use DSNJU004 to print BSDS information. From the checkpoint queue section we can match an LRSN with a target point in time.

```
CHECKPOINT QUEUE
                       14:28:31 OCTOBER 15, 2002
0      TIME OF CHECKPOINT       14:20:33 OCTOBER 15, 2002
       BEGIN CHECKPOINT RBA            0A05351409BA
       END CHECKPOINT RBA              0A05354CC6EF
       END CHECKPOINT LRSN             B861C340ABA0
0      TIME OF CHECKPOINT       14:19:28 OCTOBER 15, 2002
       BEGIN CHECKPOINT RBA            0A0533939C8D
       END CHECKPOINT RBA              0A053396939C
       END CHECKPOINT LRSN             B861C30286E9
```

Using the DB2 Version 7 parameter CHKFREQ in minutes ensures that for a given number of minutes there will be a checkpoint that may serve as a reference.

One of the major components of DB2 data sharing is the coupling facility. There are a number of structures in the coupling facility that provide for data integrity across the members of the data sharing group. These structures include the

Systems Communication Area (SCA), the LOCK structure, and all of the group buffer pools (GBP). When all of the members of a data sharing group are shut down, the SCA and the LOCK structure remain allocated, but the GBPs are de-allocated. In certain failure situations, a GBP may remain allocated with a failed-persistent connection. Also, there is information about the GBPs stored in the BSDSs of the data sharing members.

There will be information in the BSDSs, logs, and coupling facility structures that represent the state of the system at the time the members were stopped or abended.

So, when using conditional restart, the current state of the data sharing group as recorded in the BSDSs, logs, and coupling facility structures will not match the state of the system at the conditional restart point. In order to be consistent, this information must be rebuilt at restart time by deleting the structures before restarting the members with the MVS command SETXCF FORCE. Then DB2 can perform a group restart.

Before you can force the deallocation of the LOCK structure, all connections must be forced out first. If DB2 abnormally terminated and a GBP is retained in the coupling facility with failed persistent connections, these connections must be forced out as well. In the case of GBP, when a failed persistent connection is forced, it automatically deallocates the GBP structure. If these structures are not purged before restarting, when using a conditional restart the pages resident in the structure could be considered valid for the DB2 members. This could lead to data inconsistencies.

Important

In situations when a conditional restart is performed or data and logs are restored from a previous system backup, it is important to delete the DB2 structures in the coupling facility and let DB2 perform a group restart.

As specified in the *SAP Database Administration Guide*, one recovery option is to recover the whole data sharing environment to the time when a volume-based backup was obtained. In this case a conditional restart of the members is not necessary. The information for restarting the various DB2 members is stored in their BSDSs and logs. If the volume-level copy was taken when the DB2 members were suspended, the HIGHEST WRITTEN RBA for each member is equal to the suspend RBA for each member. Be sure to restore all that is needed for restarting: DB2 and SAP databases, logs from all members, BSDSs from all members, and ICF catalogs. After the structures in the coupling facility are deleted, a group restart brings the DB2 data sharing group to a consistent state based on the volume-level backup. This method is valid under the assumption that you can afford to lose the activity since the backup.

If a volume-level copy is being used in a point-in-time recovery, consider:

- Volume independence was established.
- Only the volumes containing the DB2 directory and catalog and the SAP DB2 objects are restored.
- All of the logs created between the volume-level copy point and the restart point are registered in the BSDS as either active or archive logs.
- The active logs on DASD are the ones registered in the BSDS.

- Some certain number of archives prior to the volume-level copy point are available.
- The image copies of the DB2 directory and catalog taken prior to the volume-level copy are available.
- Image copies of all SAP DB2 tablespaces taken prior to the volume-level copy are available.
- After identifying which SAP DB2 objects need recovering, use LOGONLY recovery. Prior to DB2 V7, there was no way to guarantee that the HPGRBRBA in the SAP tablespaces was current. Potentially, the LOGONLY recovery would require an extremely old archive log, once again requiring the recovery to be performed using an image copy. With DB2 V7, by using the CHKFREQ ZPARM parameter combined with the DLDFREQ parameter, you can ensure that the HPGRBRBAs for all SAP DB2 tablespaces and indexes are current. Then, LOGONLY recovery will not require archive logs that may have expired.

  The HPGRBRBA for each updated object will be updated on every *n*th checkpoint. The value *n* is based on the DLDFREQ value. If DLDFREQ=5, the HPGRBRBA should be updated every hour for each object being updated. Not all HPGRBRBAs are updated during the same checkpoint. Instead only a percentage is updated at each checkpoint. That percentage is established by (1/DLDFREQ)*100. So, if DLDFREQ=5, then 20% of the updated objects will have their HPGRBRBAs updated during one checkpoint. If CHKFREQ is set at 10 minutes, then all objects being updated should have their HPGRBRBAs updated once an hour.
- If you want to recover the following DB2 catalog and directory objects with the option LOGONLY, ensure that you have either taken an image copy of these objects using the option SHRLEVEL(CHANGE) or have run the QUIESCE utility for each of these objects. Otherwise, LOGONLY recovery may fail because HPGRBRBA is too old. These tablespaces are DSNDB01.SYSUTILX, DSNDB01.DBD01, DSNDB01.SYSLGRNX, DSNDB01.SCT02, DSNDB01.SPT01, DSNDB06.SYSCOPY, DSNDB06.SYSGROUP, and their associated IBM defined indexes with attribute COPY YES.

In summary, to recover to an arbitrary prior point in time using conditional restart, follow the next steps in a data sharing environment:

1. Create a list of tablespaces and indexes that need to be recovered.

   When using object-based backup, these tablespaces are those that changed since the target LRSN. The rest have not changed since the target point and therefore they do not need to be recovered.

   When using volume-based backup, only those page sets that have been modified between the backup and the target point should be recovered with LOGONLY. The rest are already at the target point.

   Refer to the *SAP Database Administration Guide* for a detailed explanation and for using DSN1LOGP as the basis to prepare the list. There are also considerations concerning tables dropped or created in between and REORG LOG(NO).
2. Stop all data sharing group members.
3. Copy BSDSs and LOGs that contain LRSNs larger than the target recovery point.
4. Look for a target LRSN at which all members will conditionally restart.
5. Use DSNJU003 to create a conditional restart record ENDLRSN for all members.
6. Delete all data sharing group structures in the coupling facilities.

7. If restoring from volume-based backup, flashback just the volumes that contain databases. If you are recovering from image copies, be sure to have all image copy data sets ready.
8. Update each member's system parameters and specify DEFER ALL.
9. Restart all members (group restart). New structures must be allocated.
10. Working from one member, recover the DB2 catalog and directory to the current point in time following the specific instructions for this type of database. See the section "Recover," in the DB2 publication *Utility Guide and Reference*.
11. Recover all tablespaces identified in the first step to the current point in time.
12. Recover or rebuild the indexes on recovered tablespaces.
13. Reinstate RESTART ALL in members' system parameters.
14. Start SAP and perform verifications.
15. Take a new full database backup.

Refer to the *SAP Database Administration Guide* for a detailed description that is not limited to data sharing.

DB2 utilities for online backup and point-in-time recovery

DB2 UDB for z/OS Version 8 has introduced an easier and less disruptive way for fast volume-level backup and recovery. This utility greatly simplifies backing up systems such as SAP in which the number of database objects in use, as well as recovery requirements, make volume-based backups the most efficient option.

The total solution provided by this utility is dependent on the DFSMShsm™ in z/OS V1.5 and a disk system that provides hardware-assisted volume-level copy. In order for the backup to be registered, the disk system has to support the DFSMShsm fast replication services. IBM ESS disk systems using FlashCopy takes full advantage of this solution. Even so, it is possible to take advantage of some of its features with other disk models and fast copy solutions.

One of the challenges of the online volume backup solutions prior to DB2 V8 is the need for coordination between DB2, using the SET LOG SUSPEND command, and the mechanism for triggering the FlashCopy. Before DB2 V8, the physical copy is not registered in DB2 and is thus out of DB2's control for later use as a recovery point or as a registered copy to be used in a point-in-time recovery. The procedure for obtaining a system-level copy using FlashCopy must ensure that all SAP system volumes are included and data consistency can be enforced. The procedure for recovering an SAP system using the FlashCopy-produced backup must ensure that:

- All volumes are correctly restored.
- There is a process to identify which SAP DB2 objects (page sets) require recovery.
- There is a process for generating the recovery jobs.

In DB2 V8 new utilities have been developed integrating DB2 and the fast volume copy capability. Now system-level backups using the fast volume-level copy are managed by DB2 and DFSMShsm, which work together to support a system-level point-in-time recovery. Thus, suspending the DB2 log will no longer be necessary.

The *SAP Database Administration Guide* (as of 6.40) describes procedures that accomplish system-level backup and recovery based on the DB2 utilities BACKUP SYSTEM and RESTORE SYSTEM. The procedures apply to both data sharing and non-data-sharing.

Note: The RESTORE SYSTEM utility can be used even if volume-based backups are taken outside the control of DB2, i.e. not using the BACKUP SYSTEM utility.

Data sharing considerations for disaster recovery

Another important aspect to consider, after deciding to enable data sharing for SAP on DB2 on OS/390 or 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/diskdrls/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.

  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, large CPC such as an IBM z800 or later or System z processor. 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,CONNAME=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.

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, 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 (PPRC, XRC, 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 (currently not recommended more than 20 km). 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 two data mirroring technologies:

1. Peer to peer remote copy (PPRC) in which:
   - The mirroring is synchronous.
   - GDPS manages secondary data consistency and therefore no, or limited, data is lost in failover.
   - The production site performs exception condition monitoring. GDPS initiates and executes failover.
   - Distance between sites up to 40 km (fiber).
   - Provides for both: Continuous availability and Disaster recovery solution.
2. Extended remote copy (XRC) with:

- Asynchronous data mirroring.
- Limited data loss is to be expected in unplanned failover.
- XRC manages secondary data consistency.
- GDPS executes Parallel Sysplex restart.
- Supports any distance.
- Provides only a disaster recovery solution.

The following is an example of multi-functional disaster recovery infrastructure using GDPS and PPRC 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, and the need for SET LOG SUSPEND is avoided even before DB2 V8 by using the command to freeze GDPS. 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 ESS disk subsystems with PPRC/XRC and FlashCopy functionality, and SAP/IBM replication server for application high availability.

The following figure shows the GDPS solution landscape.

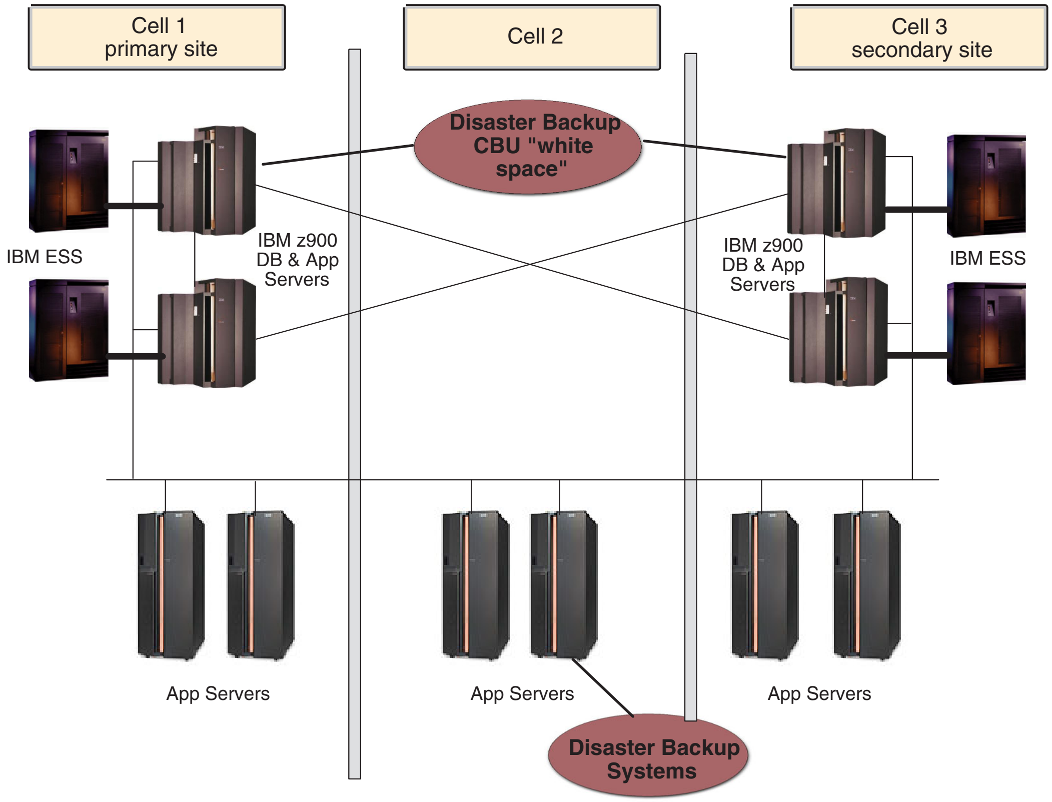


Figure 13. Example of high availability with GDPS configuration

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.

ESS primary and active set of disks is located on the primary site and, using PPRC, they are mirrored to the secondary site. If the BACKUP SYSTEM utility is employed to copy the data, the ESS FlashCopy activity takes place at the primary site. Otherwise all ESS activity takes place at the secondary site. The design keeps symmetry between both sites, having the same ESS disk capacity on each site. Therefore, if one site is not available (disaster, maintenance), the other is able to provide an alternate backup process.

In DB2 V7, 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.

The following figure illustrates the process of obtaining a non-disruptive volume backup before the availability of the DB2 V8 BACKUP SYSTEM utility.

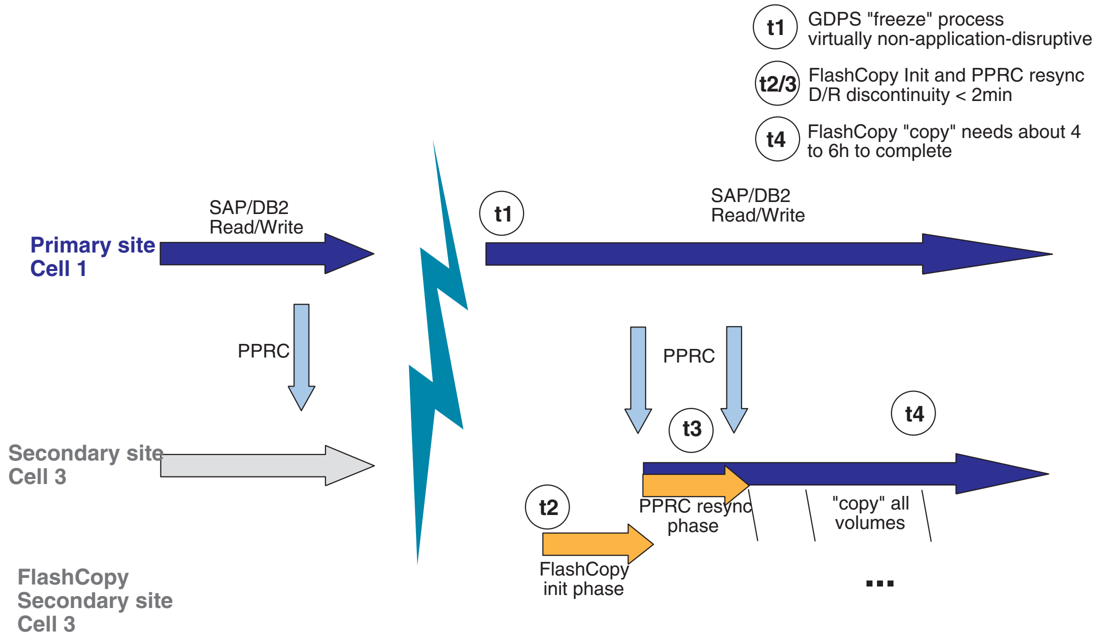


Figure 14. Process for obtaining a non-disruptive volume backup without the BACKUP SYSTEM utility of DB2 V8

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 PPRC 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), PPRC synchronizes the volumes on both sites.

Important

Even with this configuration, there is a possibility of having 32 KB page sets in recovery pending status after restoring from backup (see SAP Note 363189). In this case a recovery from image copy is still needed to reset status. This exposure can be avoided by converting all page sets to a control interval size that matches the page size of page sets in DB2 V8.

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 PPRC relationship 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 PPRC 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 PPRC.

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 Homogeneous System Copy* documentation (see “SAP documents” on page 372). 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.

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 ESS 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 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. 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.

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 2. Network considerations for high availability

Chapter 5. 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, three different recovery mechanisms for network component outages 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 289. 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 for Windows, z/OS VIPA usage, and timeout behavior.

Introduction

A network can be subdivided into a physical layer and a communication software 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.

Planned or unplanned outages of a network result in interruptions of the communication path between the remote SAP 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.

SAP offers its own recovery mechanism, **SAP sysplex failover**. If set up correctly, all network outages can be recovered with it. However, SAP sysplex 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, Linux on System z, and Windows 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) (except Windows)

If both mechanisms are set up appropriately in addition to SAP sysplex failover, you can achieve recoveries that are transparent to the end user in most failure scenarios. This chapter gives you some hints and recommendations on how to set up and utilize such features for communication between a remote SAP application server and the z/OS database server.

After some general recommendations, all three recovery mechanisms (SAP sysplex failover, OSPF, and VIPA) are explained in detail.

General recommendations

Hardware considerations

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.

To get optimum network performance for remote SAP application servers connected via a LAN only (for example, AIX SAP application servers), use switched OSA-Express Gigabit Ethernet and exploit jumbo frames with an MTU of 8992. This has superior latency and capacity. To connect such remote SAP application servers to the sysplex in a high availability configuration, you need to duplicate LAN hardware such as adapters, cables, and switches.

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.

Note: 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.

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.

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 4.4 (or later), we recommend setting up a virtual LAN for the guests that is based on z/VM Virtual Switch (VSWITCH) technology.

Under z/VM 4.4, the IEEE 802.1Q VLAN support is available based on z/VM Virtual Switch. One benefit of this configuration is that you do not need a router stack to route to the real network as with Guest LAN configurations. A Guest LAN allows you to create multiple virtual LAN segments within a z/VM environment.

Note: While the structures and simulated devices related to the Guest LAN under z/VM are ’virtual’, IBM uses the term Guest LAN rather than Virtual LAN, because the term Virtual LAN (VLAN) has a different meaning in the networking world.

Another benefit is security. A VSWITCH can securely control which guest uses which VLANID(s).

z/VM VSWITCH is a z/VM networking function introduced with z/VM 4.4. It is designed to improve the interaction between guests running under z/VM and the physical network connected to the System z processor. You do not need to dedicate a real OSA device directly to each guest. Also, traffic between the guests connected to the VSWITCH does not go through the OSA adapter.

The following figure depicts the use of VSWITCH:

Network considerations

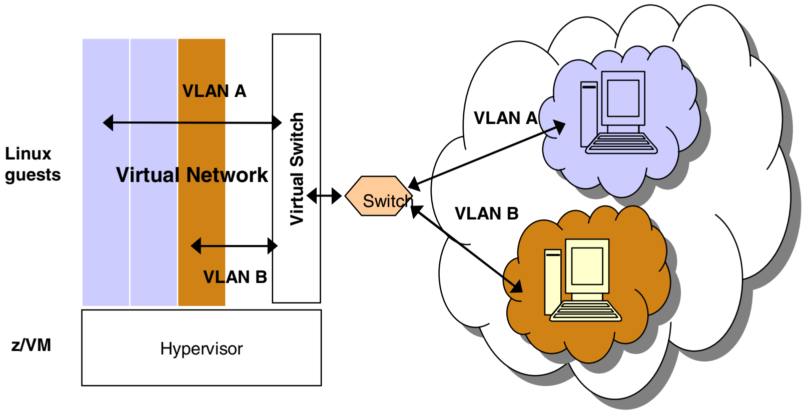


Figure 15. Sample VSWITCH utilization

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.

SAP sysplex failover recovery mechanism

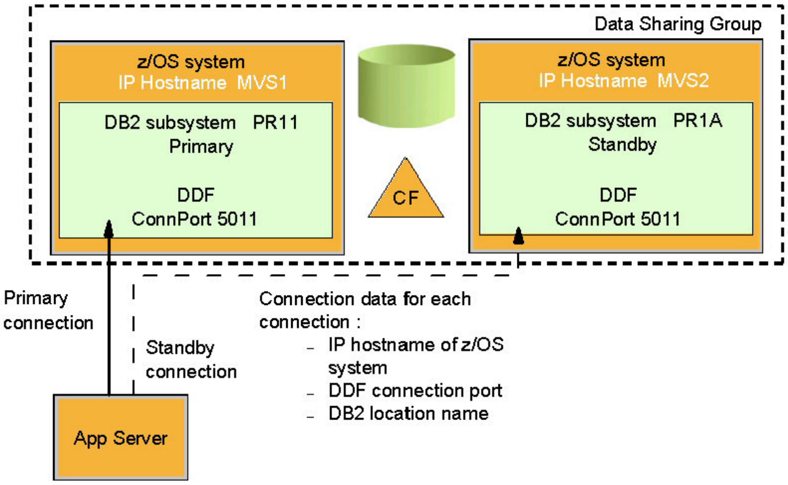


Figure 16. SAP sysplex failover configuration: Option 0 example

SAP sysplex failover is the capability of SAP systems 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.

Both features together, SAP sysplex failover and DB2 data sharing, address failures of, for example, the database server, the network, the ICLI server[1], 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 73), SAP sysplex 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.
- All network components need to be duplicated.
- The SAP failover parameters are set up correctly. For SAP 4.6D, you define them in each instance profile (e.g., standby hostname, ICLI port). Starting with SAP 6.10, they are configured by a profile (connect.ini) that provides a list of database connections for each application server or group of application servers.

It is possible to define different system configurations to handle the failure of one or several components. In the configuration depicted above, 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.

For detailed information on SAP sysplex failover support, see SAP Note 98051.

The same failover concept applies when using DB2 Connect, although the client connects to the DB2 DDF address space rather than to the ICLI server.

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 1 and for a Fast Ethernet link as 3. 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. Over such equal cost paths, OSPF does outbound load balancing.

The OSPF routing protocol is implemented by:

- The OMPROUTE daemon under z/OS
- The gated daemon under AIX
- The zebra and ospfd daemons under Linux on System z
- 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.x and Path MTU discovery

Beginning with AIX 5.3, path MTU (PMTU) discovery can be used on duplicate routes. In prior versions, PMTU discovery cannot be used on duplicate routes for the following reasons:

- Beginning with AIX 5L, you can have several equal-cost routes to a given network, but with different associated gateways, on a system for which PMTU discovery is enabled. When traffic is sent to a host on that specific network, a host route will be cloned from whatever network route was chosen by the cyclic multiplexing code of the multipath routing algorithm. Because the cloned host route is always more specific than the original network route from which the clone was derived, all traffic to that host will use the same gateway as long as the cloned route exists. Consequently, no cyclic multiplexing among the different gateways associated with the equal-cost route to the specific network will take place.
- Since PMTU discovery is enabled by default in AIX 5L, system administrators may consider disabling the network option `tcp_pmtu_discover` to turn off route cloning (in order to take full advantage of the new multipath routing feature). This measure will prevent the creation of the cloned host routes and will instead allow cyclic multiplexing between equal-cost routes to the same network. However, make sure that you have set `tcp_mssdflt` as described under "Performance tuning aspects on AIX over LAN" in the *Connectivity Guide* (SC33-7965-03).

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. A round-robin scheme is used to select a route when there are multiple routes with the same destination and cost. Starting with Version 5.3, there is the ability to define different multipath routing methods via the SMIT mkroute fastpath. We recommend not using the default ″weighted round-robin″ method when you run ICLI client/server. The round-robin approach does not work well with the special connection establishment by ICLI client/server.

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.

A VIPA requires that a dynamic routing protocol like OSPF is used and the VIPA must belong to a different subnet than 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:

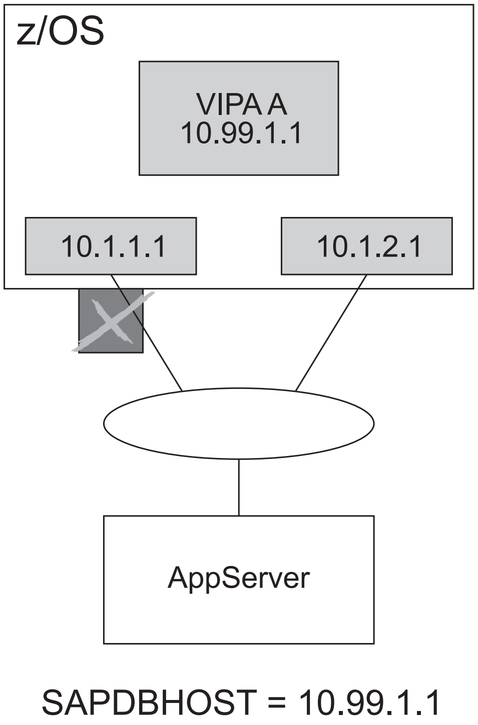


Figure 17. VIPA and OSPF recovery mechanisms under z/OS

The *VIPA A* (10.99.1.1), which belongs to subnet 10.99.1, represents the z/OS application (DDF instance, ICLI server, 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 74 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 mechanism on Windows” on page 78.

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 system, 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.

This is due to the fact that, by default, OSPF manages routes to subnets only. A subnet is either directly accessible or it is a remote subnet and the first gateway in the path to the subnet is directly accessible. Because OSPF by default does not manage host routes, it does not change a route if a host in a 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: If its own primary NIC to a directly accessible subnet fails, it switches to the backup (’secondary’) NIC.
  For OSPF, the primary NIC to a subnet is the adapter which is used to exchange OSPF data. However, in an SAP environment with VIPA support, OSPF’s primary NIC may not be the adapter over which the SAP database traffic flows: In an SAP system, the SAP database traffic always flows over the NIC on which the VIPA is registered. As the algorithm used to register a VIPA on one of several OSA-Express NICs on a machine cannot be controlled, this may be any of the NICs on a machine - not necessarily the one recognized by OSPF as its primary. If the database traffic does not flow over its primary NIC, OSPF will not react when the ’secondary’ NIC fails and the SAP traffic will stop, which results in a downtime for the SAP users.

  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: OSPF only recalculates the route to a subnet which is not directly accessible (’remote’), if its ’gateway’ to the remote subnet is down, i.e., when a complete remote subnet can no longer be reached.

  Consequently, if the NIC on an 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 if a complete remote subnet becomes inaccessible when the NIC on the application server fails. This can be achieved by defining a VIPA (and, therefore, a VIPA subnet) 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 18 on page 77, 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.

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 z9 and zSeries Device Drivers, Features, and Commands*, SC33-8289, available from

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

Note: The Linux documentation is organized by Linux kernel version and stream designation. Versions 2.4 (June 2003 stream) and 2.6 (October 2005 stream) are currently supported.

In order to exploit multiple equal cost paths, you must re-compile `quagga` (included in SLES 9) in order to get multipath support. In the file `/usr/src/packages/SPECS/quagga.spec`, under the ./configure entry for IP multipath support, add the keywords

```
--enable-netlink
--enable-multipath=0 (or the number of multipaths needed)
```

Note: For the **zebra-0.93b** package, which is included in SLES 8, we also had to remove the `--enable-broken-alias` keyword.

The Source VIPA function is also available on Linux on System z via the src_vipa utility. It is a recommended utility that provides a very flexible means of source address selection to arbitrary applications. src_vipa is a user-space utility and involves no kernel changes. You can download it from:

`http://www.ibm.com/developerworks/linux/linux390/useful_add-ons_vipa.html`

Note: The src_vipa utility does not work in the following cases:

- TCP connections that are established from the Linux kernel, such as for NFS
- DB2 Connect connections

  We require that DB2 Connect be run in so-called gateway mode. In this mode, the gateway process (started via **db2start**) opens connections to DB2 on z/OS. This process runs with the s-bit set and is therefore a 'setuid 0' program. src_vipa generally does not work for 'setuid 0' programs (see the following point).
- 'setuid 0' programs such as `ping` or `rlogin`.

  The reason is that the src_vipa utility uses the LD_PRELOAD mechanism, which for security reasons does not work for 'setuid 0' programs.

Using `iptables` to ’simulate’ Source VIPA is a possible circumvention for the above situations. Be aware, however, that `iptables` manipulates IP packets, not TCP connections. The TCP connections on the client still show, as a source IP, the IP address of the real interface and not the Source VIPA IP address. On the server side, the target IP will be the ’replaced’ Source VIPA IP. This mismatch makes it difficult to debug problems. Tools based on TCP connections are not aware of the IP packet manipulation.

For the following recommended network:

- 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.

the `iptables` rules are:

```
iptables -v -t nat -A POSTROUTING -d 10.96.1.1 -p tcp -s 10.1.1.3 -j SNAT --to-source 10.98.1.1
iptables -v -t nat -A POSTROUTING -d 10.96.1.1 -p tcp -s 10.1.2.3 -j SNAT --to-source 10.98.1.1
```

You can check whether the rules are active by using the following command:

```
iptables -t nat -L POSTROUTING
```

The Source VIPA is in this case 10.98.1.1, belonging to subnet 10.98.1.x. It must be assigned, for example, to `dummy0` and defined to the OSA card as a VIPA for both interfaces using

```
qethconf vipa add 10.98.1.1 eth1  and
qethconf vipa add 10.98.1.1 eth2
```

if the SAP application server A interfaces used are eth1 and eth2. To verify that the rules are in effect, issue

```
rlogin 10.96.1.1
```

from SAP application server A. On z/OS USS, enter

```
netstat | grep "INETD"
```

and check for the line with local rlogin port 513:

```
User Id  Conn     Local Socket           Foreign Socket         State
INETD1   0001BBE6 10.96.1.1..513         10.98.1.1..1023        Establsh
```

On Linux for System z, enter

```
netstat -n --tcp
```

and check for the line with remote rlogin port 513:

```
Proto Recv-Q Send-Q Local Address        Foreign Address        State
tcp        0      0 10.1.1.3:1023        10.96.1.1:513          ESTABLISHED
```

The source IP on Linux on System z is the IP address of either eth1 or eth2, depending on which route is currently used. Assuming it is for eth1, enter

```
ifconfig down eth1
```

The rlogin connection should still work when you again run

```
netstat | grep "INETD"
```

The Source VIPA function may be available via a future release of the `quagga` routing daemon, which supports the ’src <IP>’ option for dynamically added routes.

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
- SAP sysplex failover (for application server)
- Different subnets for OSPF
- VIPA exploitation on z/OS
- VIPA and Source VIPA exploitation on the application server side.

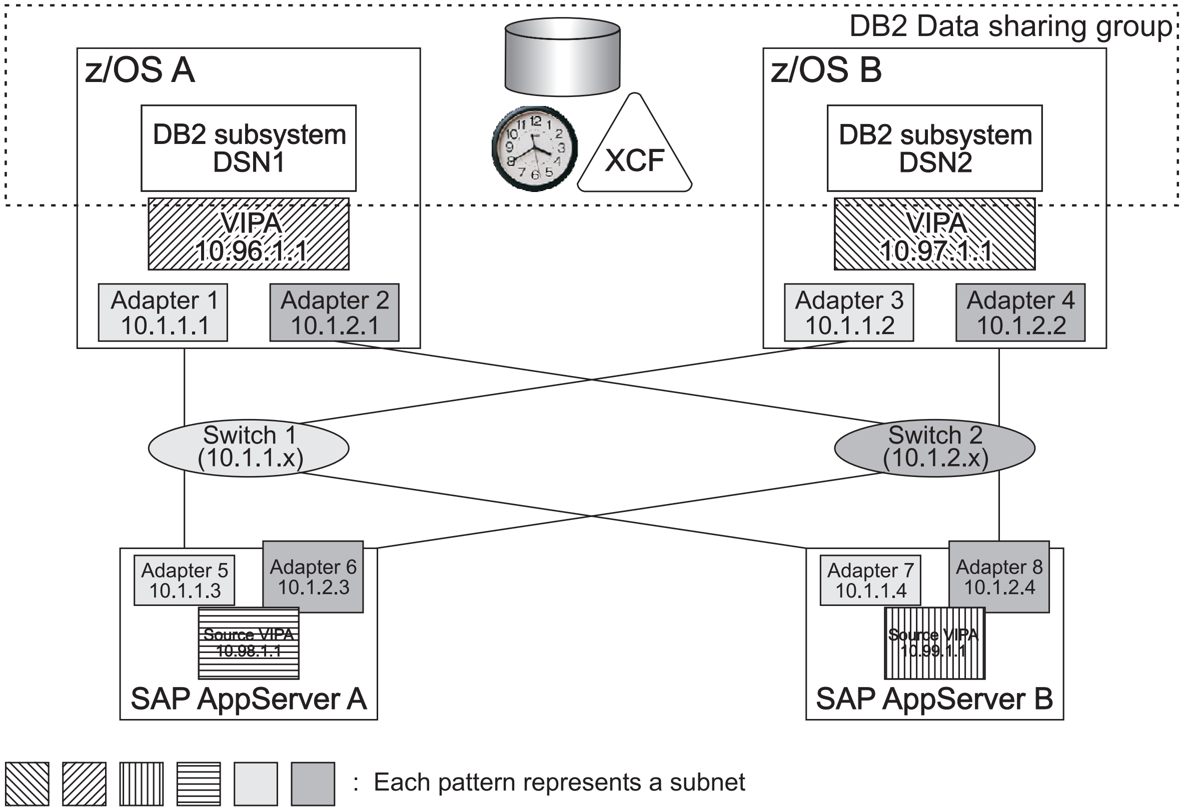


Figure 18. Recommended setup for a high availability network

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.

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

Table 6. 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	SAP sysplex 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 mechanism on Windows

The Windows platform does not support the VIPA recovery mechanism. Therefore, other mechanisms have to be employed to recover from Windows adapter failures. For Windows application servers, 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)

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

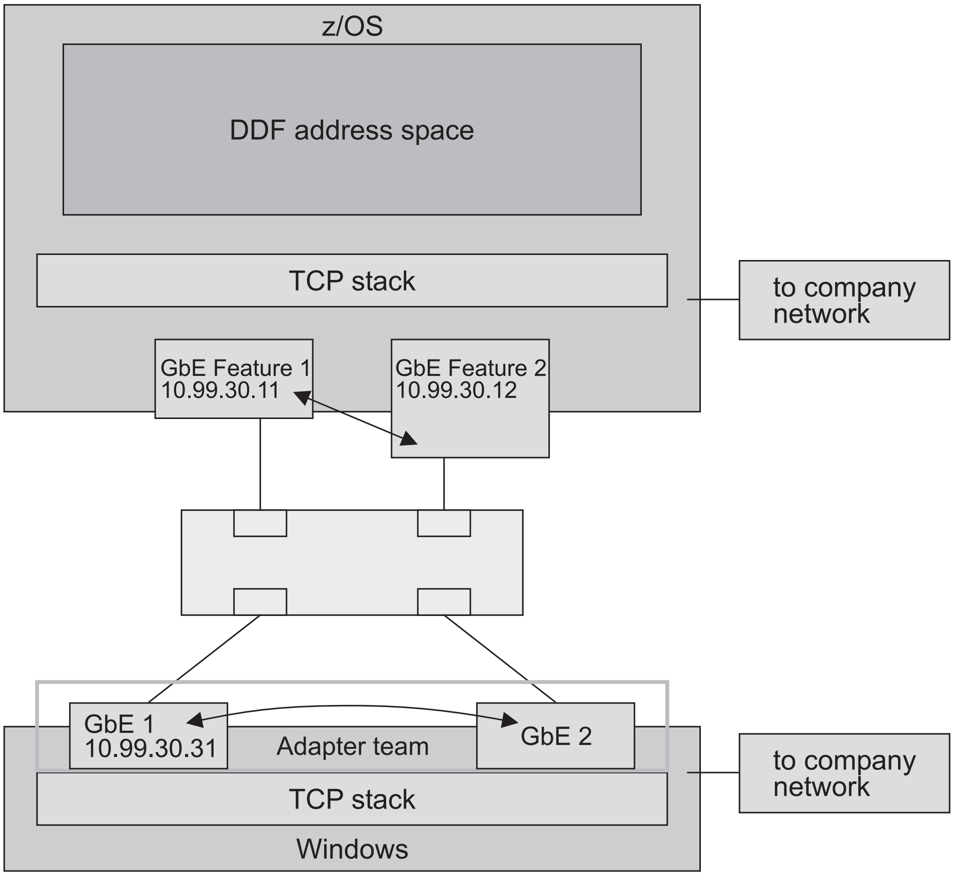


Figure 19. System setup with z/OS ARP takeover and Windows adapter teaming

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.

This solution has the disadvantage, that recoveries from switch failures are not possible.

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.

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.:

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

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 or ICLI.

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

For a detailed description of all (ICLI) environment variables, see the respective *Planning Guide*.

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 7. 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).

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

Timeout behavior of the Linux on System z 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.

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 300.

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:

```
300 + 60 = 360
```

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.

ICLI server-specific keep-alive interval times:

For ICLI only

TCP/IP allows using shorter keep-alive intervals. Additionally, you can set a keep-alive interval specifically for an application independent of the system-wide setting. The ICLI server exploits this function.

You can configure the TCP/IP keep-alive behavior for the ICLI server by setting the environment variable *ICLI_TCP_KEEPALIVE* with a specific value. The range of supported values and their meaning are described below.

If you do not set the environment variable, the ICLI server uses the ICLI server default keep-alive value of 360 seconds. This default value is less than the default value for DB deadlock and timeout detection (which is normally 10 minutes).). This guarantees that an ICLI server thread with a broken connection will not hold a DB2 resource long enough that another ICLI server thread runs into a DB2 resource or deadlock timeout.

Possible *ICLI_TCP_KEEPALIVE* values range from 0 to 2147460 whereby 0 has a special meaning: It disables keep-alive processing. Values between 1 and 2147460 can be set according to Table 8:

Table 8. Possible ICLI_TCP_KEEPALIVE values

Specified TCP_KeepAlive time (T)	**Seconds to first probe**	**No. of probes**	**Probe interval**	**Max. interval**
T=0 (keep-alive disabled)	n.a.	n.a.	n.a.	n.a.
0<T≤5	T	1	1	T+1
5<T≤10	T	1	2	T+2
10<T≤30	T	1	5	T+5
30<T≤60	T	1	10	T+10
60<T≤120	T	1	20	T+20
120<T≤300	T	2	20	T+40
300<T≤600	T	2	30	T+60
600<T≤1800	T	5	30	T+150
1800<T≤3600	T	5	60	T+300
3600<T≤7200	T	10	60	T+600
7200<T≤2147460 (35791 * 60 = 214760)	T	10	75	T+750
T>2147460	2147460	10	75	2147460+750

In Table 8, **T** is the time in seconds after which the keep-alive mechanism starts checking an idle connection. The **No. of Probes** is the total number of probes (including all probes which get additionally sent after ″probe acknowledgement″ timeouts) which will be sent to the other end . The **Probe interval** is the time which the keep-alive mechanism waits for a probe acknowledgement. The **Max. interval** is then the total time after which TCP/IP regards the connection as broken. The following is an example for **T** = 125. Please remember that only **T** can be specified, and that it determines the other values as illustrated in the following:

T = 125 results in **No. of Probes** = 2 and **Probe interval** = 20. If the connection is broken (no acknowledgement received for keep-alive probes), the first probe is sent after 125 seconds and a second probe is send after an additional 20 seconds. The second probe is again not acknowledged after another 20 seconds, after which TCP/IP returns control to the ICLI server. This results in a total of 165 seconds (125 + (2*20) = 165).

As previously mentioned, the default value used by the ICLI server is 360. Using this value, the keep-alive recognizes a broken connection after 7 minutes (360 + 60), sending two probes after 360 seconds in an interval of 30 seconds.

If you want to change the default, your new value should be less than the value specified for DB deadlock and timeout (which is normally 10 minutes). Then an ICLI server thread with a broken connection will not hold a DB2 resource long enough that another ICLI server thread runs into a resource or deadlock timeout.

End of For ICLI only

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 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.

When DB2 Connect and DDF are being used, there is a keep-alive interval that governs when a network connection is considered broken. Once this interval is reached for a connection from a DB2 Connect client, DB2 terminates the connection and releases associated locks. The keep-alive interval is controlled by the DB2 ZPARM TCPKPALV, which defaults to 120 seconds. This is also the recommended value for SAP systems.

For ICLI only

It may happen that an ICLI server work thread holds a lock on a resource, and just at that time its connection to the corresponding ICLI client is lost. Such an ICLI server work thread does not release the lock until it is closed. From SAP release 4.6D, such a situation is handled by the ICLI server as soon as the work process (ICLI client) reconnects. It automatically cancels the DB2 thread. For earlier SAP R/3 releases we recommend that you cancel the DB2 thread manually after you have identified it.

End of For ICLI only

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 3. Application server considerations for high availability

Chapter 6. Architecture for a highly available solution for SAP

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

We discuss the following:
- Architecture components
- Failure scenarios and impact

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 J2EE-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 *J2EE 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 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

- 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.

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 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 20 on page 93, 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.

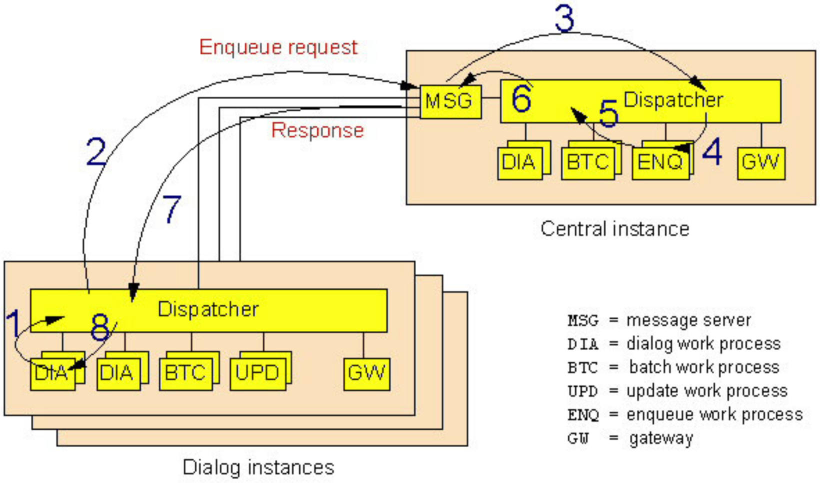


Figure 20. SAP enqueue services with the old central instance concept

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 21 on page 95.

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 21 on page 95 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.

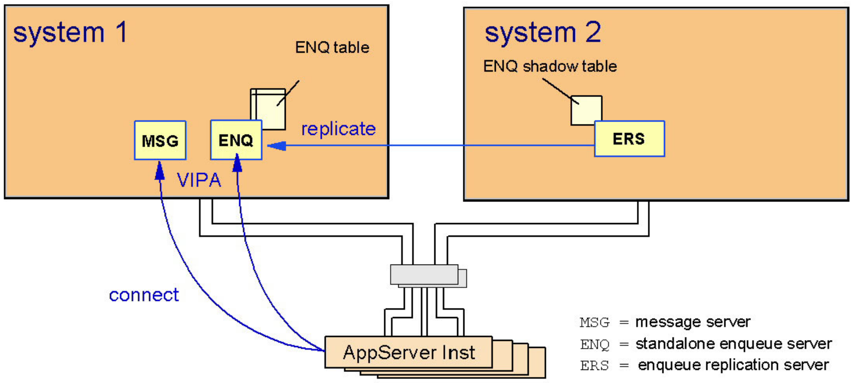


Figure 21. Initial startup of SCS

If the first system fails, the second system takes over the role of the first one, as shown in Figure 22 on page 96:

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

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

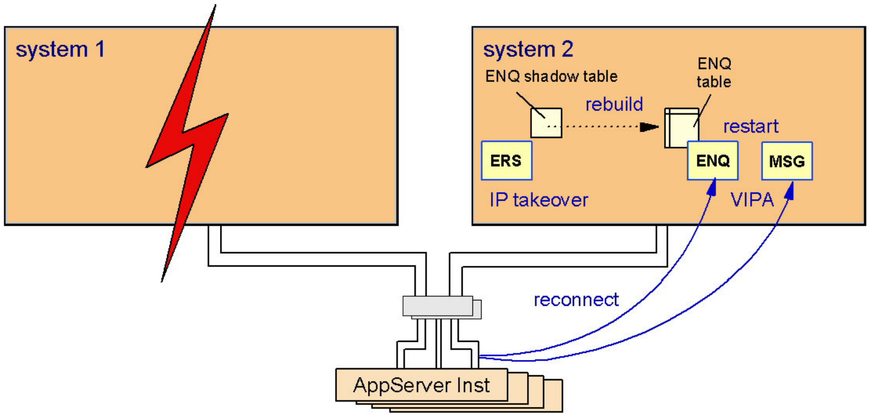


Figure 22. Failure of SCS and recovery of the enqueue table

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 23.

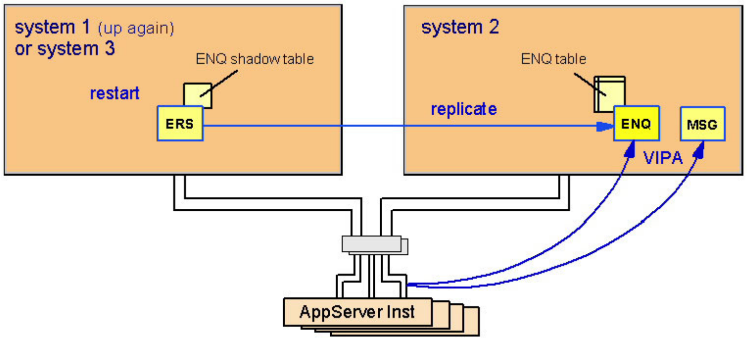


Figure 23. Movement of the enqueue replication server

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* can be activated by and moved with an application.

For a fault-tolerant network it is furthermore recommended to define a VIPA together with the SOURCEVIPA 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 24 shows the general concept of a fault-tolerant network with duplicated network components and VIPA.

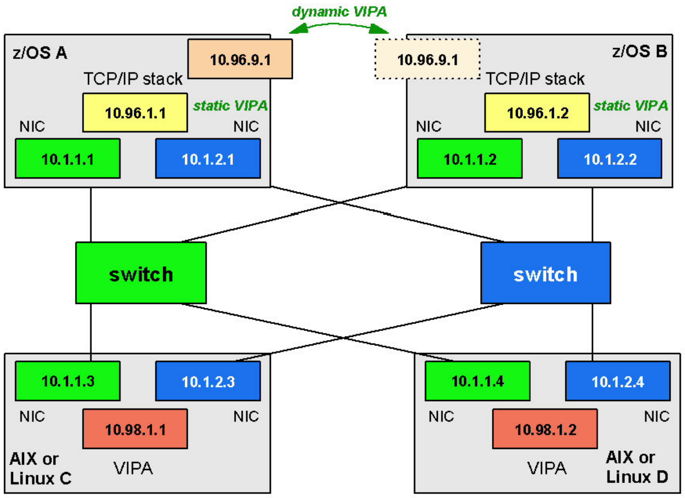


Figure 24. General concept of a fault-tolerant network

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 5, “Network considerations for high availability,” on page 65 for details on how to set up a highly available network.

The following figures show how dynamic rerouting works. In Figure 25 on page 98 the virtual IP address virt_addr_1 on system A can be reached through IP addresses addr_1, addr_2 and addr_3. These real addresses are seen as gateways to the virtual IP address. ENQ and MSG indicate two applications running on that system. You can imagine that these are the SAP enqueue server and the message server.

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

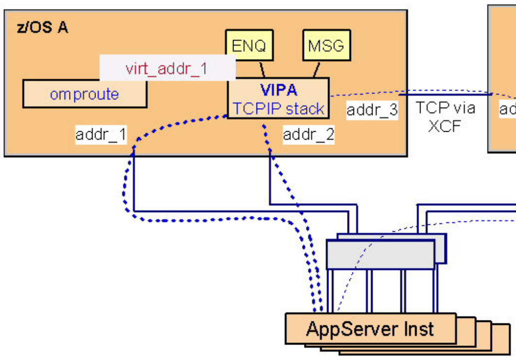


Figure 25. Alternative paths in a duplicated network

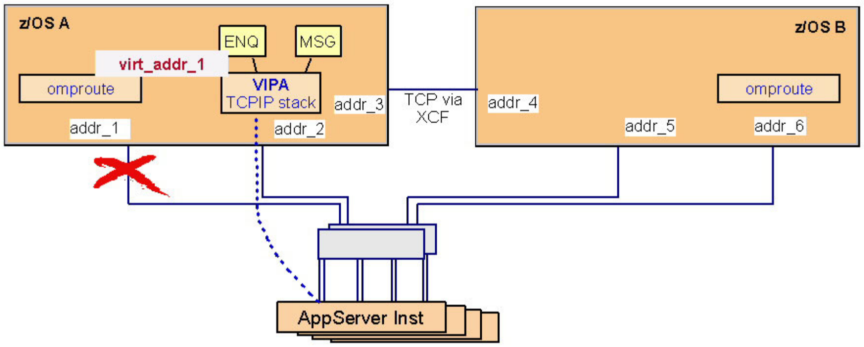


Figure 26. Rerouting if a network adapter card fails

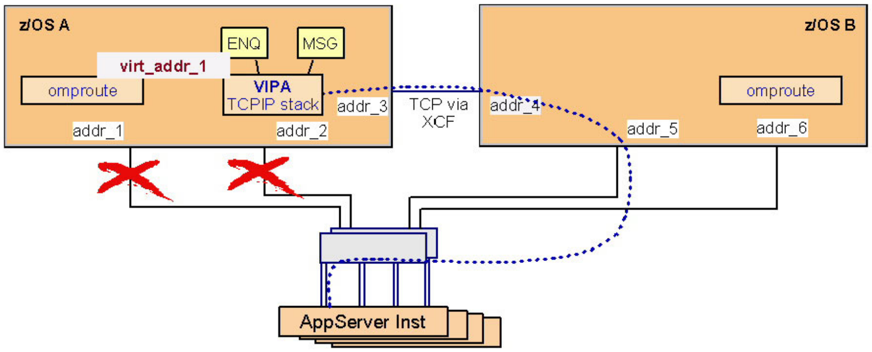


Figure 27. Rerouting in a sysplex even in case of two failing network cards

What happens if network adapter card addr_1 fails? As shown in Figure 26 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 27 on page 99, 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 27 on page 99 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 an 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 28. 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.

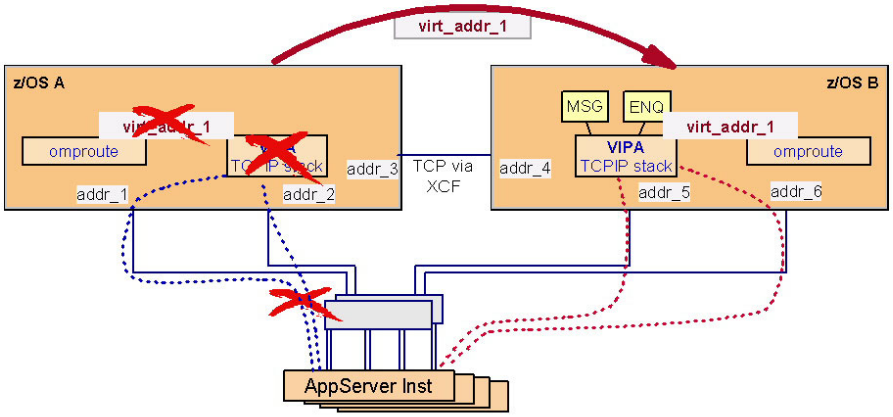


Figure 28. VIPA takeover and dynamic routing

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 to connect to SAP components *that are not moved* between systems, like the DDF address spaces (or ICLI servers) or the application server instances.

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

- NFS server
- SCS
- SAP network interface router (saprouter)

While the rerouting shown in Figure 25 on page 98 through Figure 27 on page 99 is applicable to both static and dynamic VIPAs, the takeover shown in Figure 28 on page 100 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.[3]

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.

[3] 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 107.

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 29 on page 103 and Figure 30 on page 103. Note that the NFS VIPA is different from the VIPA of SCS. So they can be handled independently of each other.

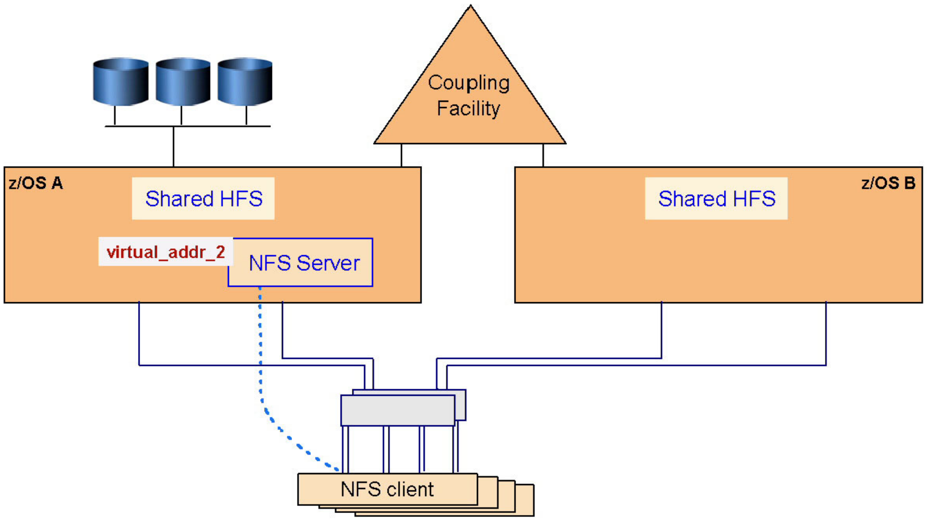


Figure 29. Initial NFS client/server configuration

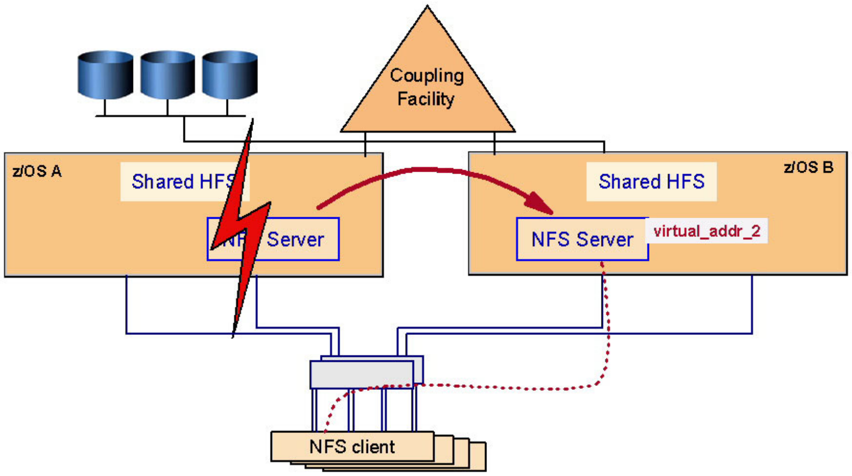


Figure 30. Failover of the NFS server

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 91.

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 the SAP sysplex failover feature to move workload off the DB2 member needing the outage to an available DB2 member in the data sharing group. In the case of an unplanned outage, the sysplex failover feature is used to switch the workload to a surviving DB2 member. In either situation, the SAP system remains available to the end users.

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

Remote application server and sysplex 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 the SAP sysplex failover feature. This removes the database server as a single point of failure.

General information

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. 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`

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). You should also check SAP Note 98051 for details. 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, ICLI server[1], network, or gateways can be achieved with sysplex failover support 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.

ICLI server-specific failover

For ICLI only

For SAP and DB2 combinations using ICLI, we recommend that you dedicate one ICLI server instance to only one SAP system.

You can define to which ICLI server instance on the standby database server an application server connects in the case of failure.

Restarting a failed ICLI server is done by Tivoli System Automation (SA) and is part of the SA z/OS policy. We recommend using SA to recover quickly from an ICLI failure. Alternatively, you can use Automatic Restart Management (ARM) (see *z/OS MVS Setting Up a Sysplex* and the section "ICLI server registration with Automatic Restart Management" in the respective *Planning Guide*.

Figure 31 shows a typical failover setup with three DB2 members and three ICLI servers, each serving as standby for all others.

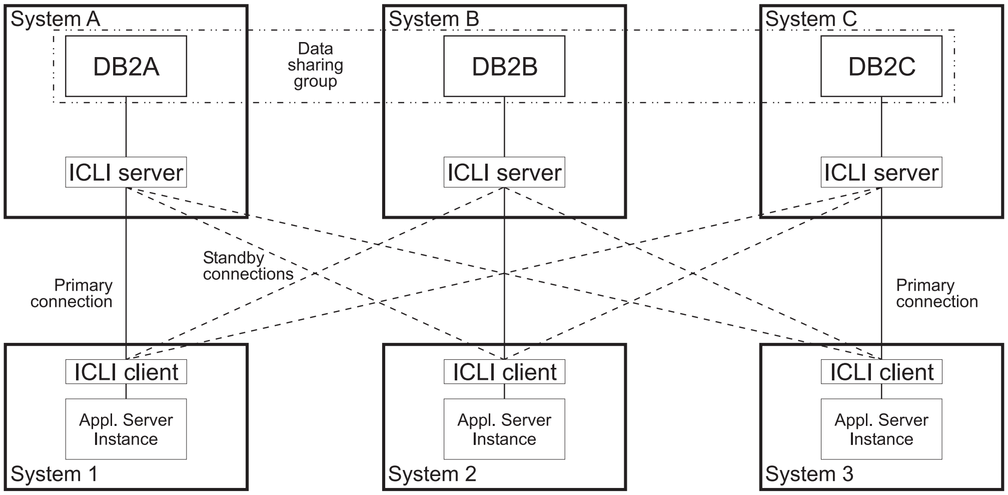


Figure 31. Application servers connected to primary and standby database servers

If a data sharing member cannot handle the additional workload of a failed-over application server because the available DB2 capacity (such as virtual memory) is insufficient, a DB2 subsystem that only acts as a standby database server can be used. This standby DB2 subsystem can either reside on a z/OS system on which primary DB2 subsystems are already running or on a separate z/OS system.

End of For ICLI only

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:

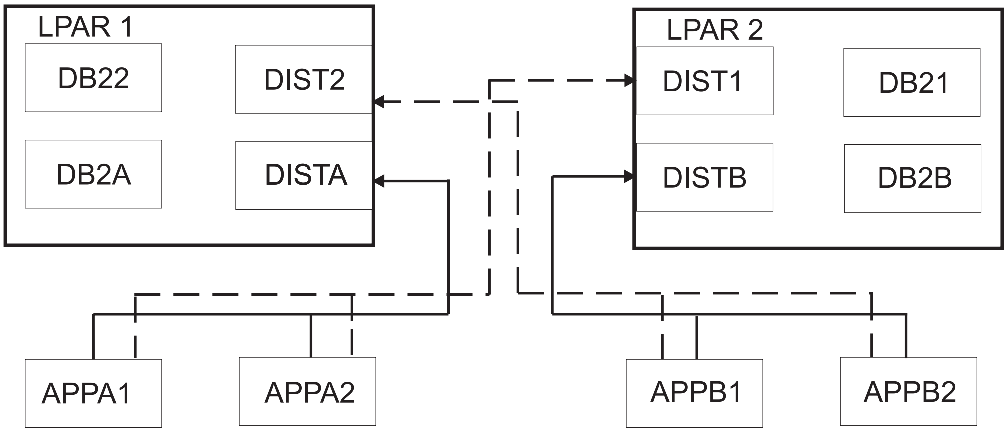


Figure 32. Failover setup using DB2 Connect, with multiple DB2 members in the same LPAR

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.

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 this 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 237.

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 9, 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 9. Simple configuration

Failure	Impact	Actions
DB2	• Rollback of transactions • Application servers wait until DB2 is up again	SA: Restart DB2 User: Restart transactions
DB2 Connect instance	• Rollback of transactions	The SAP application server automatically restarts the DB2 Connect instance.

Table 9. Simple configuration (continued)

Failure	Impact	Actions
ICLI server[1]	• Rollback of transactions • Application servers wait until ICLI server is up again	SA. Restart ICLI server User: Restart transactions
Central instance	• Rollback of transactions • Application servers wait until central instance is up again	SA. Restart central instance User: Restart transactions
Message server	• 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	• 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
Gateway	• For most transactions, no impact • Connections to registered RFC servers inhibited until they have reconnected to gateway	SA. Restart gateway
Syslog collector	• For most transactions, no impact • Global syslog file out of date	SA. Restart syslog collector
saprouter	• User sessions lost • Reconnect inhibited	SA. Restart saprouter User: Reconnect
NFS server	• 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

Table 9. Simple configuration (continued)

Failure	Impact	Actions
File system	• 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)	• Lost connectivity to message server and gateway server (see failures of these components) • Rollback of transactions on remote application servers • Remote application servers wait until network is up again	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
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, sysplex failover, double network (single central instance)

The scenario in Table 10 on page 111 builds on the previous scenario by adding DB2 data sharing, SAP sysplex failover, shared HFS, and a double network with VIPA and OSPF. This scenario is still using the old-style central instance.

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

Table 10. DB2 sysplex data sharing configuration with double network

Failure	Impact	Actions
DB2	• Rollback of transactions • Local z/OS application servers wait until DB2 is up again • *Remote application servers fail over to other DB2 subsystems*	SA. Restart DB2 User: Restart transactions
DB2 Connect instance	• Rollback of transactions	The SAP application server automatically restarts the DB2 Connect instance.
ICLI server[1]	• Rollback of transactions • *Application servers reconnect to ICLI server or fail over to standby ICLI server and DB2 subsystem*	SA. Restart ICLI server User: Restart transactions
Central instance	• Rollback of transactions • Application servers wait until central instance is up again	SA. Restart central instance User: Restart transactions
Message server	• 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
Application server instance	• 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
Gateway	• For most transactions, no impact • Connections to registered RFC servers inhibited until they have reconnected to gateway	SA. Restart gateway
Syslog collector	• For most transactions, no impact • Global syslog file out of date	SA. Restart syslog collector
saprouter	• User sessions lost • Reconnect inhibited	SA. Restart saprouter User: Reconnect

Table 10. DB2 sysplex data sharing configuration with double network (continued)

Failure	Impact	Actions
NFS server	• 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	• 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 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
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 11 on page 113 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 7, “Planning and preparing a System z based end-to-end high availability solution,” on page 115.

Table 11. Fully implemented high availability solution for SAP

Failure	Impact	Actions
DB2	• Rollback of transactions • Local application servers wait until DB2 is up again • Remote application servers fail over to other DB2 subsystems	SA. Restart DB2 User: Restart transactions
DB2 Connect instance	• Rollback of transactions	The SAP application server automatically restarts the DB2 Connect instance.
ICLI server[1]	• Rollback of transactions • Reconnect to ICLI server or failover to standby ICLI server and DB2 subsystem	SA. Restart ICLI server User: Restart transactions
Enqueue server	*None*	SA. Failover enqueue server SA. Move enqueue replication server
Enqueue replication server	*None*	SA. Restart enqueue replication server
Message server	• *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	• 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
Gateway	• For most transactions, no impact • Connections to registered RFC servers inhibited until they have reconnected to the gateway	SA. Restart gateway
Syslog collector	• For most transactions, no impact • Global syslog file out of date	SA. Restart syslog collector
saprouter	• User sessions lost • Reconnect inhibited	SA. Restart saprouter User: Reconnect

Table 11. Fully implemented high availability solution for SAP (continued)

Failure	Impact	Actions
NFS server	• None • If data was written to file, last written data is in doubt	SA. Restart NFS server
File system	• 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, gateway, syslog collector fail (see failures of individual components)	SA. Restart TCP/IP SA. Restart enqueue server, message server, 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 7. Planning and preparing a System z based end-to-end high availability solution

This chapter describes planning tasks to be performed in order to prepare a new, or enable an existing, SAP on DB2 UDB for OS/390 and z/OS system for the high availability solution using SA 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 2, “Network considerations for high availability,” on page 63.

Sample high availability solution configuration for SAP

We designed a configuration to demonstrate how SA z/OS and SA MP can be used to make all of the necessary SAP components highly available. Our configuration included two LPARs running z/OS in a sysplex with a DB2 data sharing database, and one LPAR with z/VM having Linux guests. This is shown in Figure 33 on page 116.

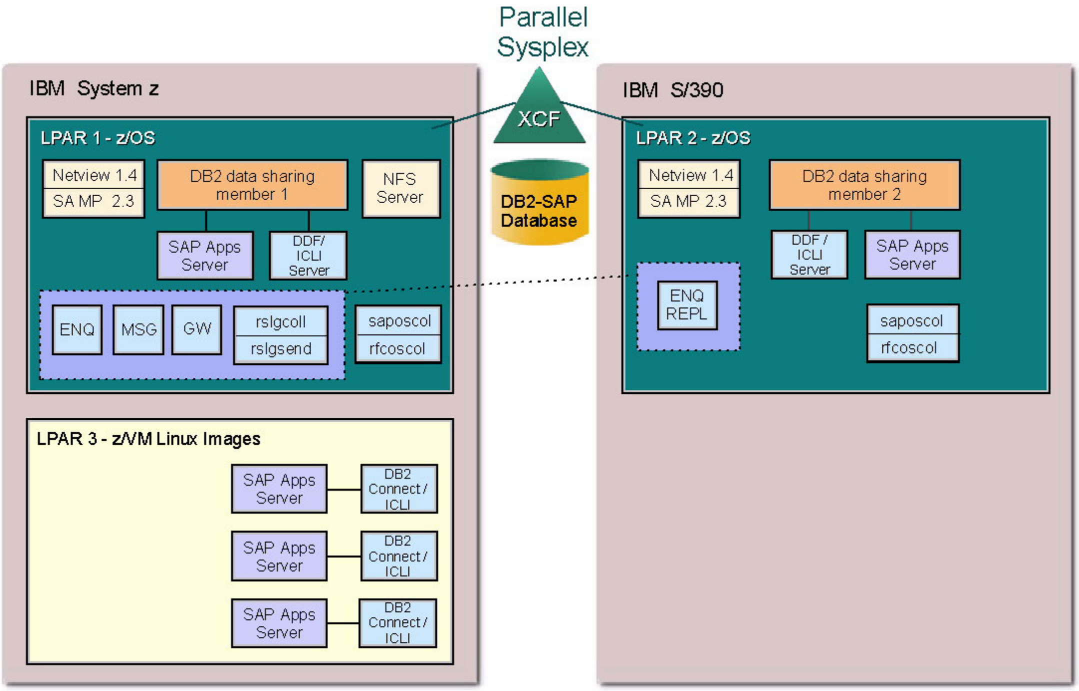


Figure 33. High availability solution configuration for SAP

Software prerequisites

Table 12 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 12. Software requirements for the HA solution

Product name	Minimum level requirement	Recommended level
z/OS	V1.4 APAR OW53313 for the NFS server	V1.7 [1]
DB2 Universal Database for OS/390 and z/OS	Version 7	Version 7 or 8
Tivoli NetView for z/OS (required by Tivoli System Automation for z/OS	V5.1	V5.2 [1]
System Automation for z/OS	V2.3	V3.1 [1]

Table 12. Software requirements for the HA solution (continued)

Product name	Minimum level requirement	Recommended level
SAP R/3 4.6D mySAP SAP Web Application Server 6.20 SAP NetWeaver ’04 SAP NetWeaver 2004s	For the appropriate patch level for the SAP basis release and the enqueue and enqueue replication servers, see SAP Note 524816.	
[1] Or higher supported level		

If you have an SAP release older than 6.30, you must download the standalone enqueue server and associated components as a patch (ENSERVER*.SAR, RFCPING*.SAR) from the SAP Service Marketplace:

`http://service.sap.com/patches`

The standalone enqueue server and its associated components are available as a patch (ENSERVER*.SAR) at the SAP Service Marketplace:

- -> SAP WEB AS 6.20 -> Binary Patches
- -> SAP KERNEL 6.20 32-BIT -> OS390_32 -> Database independent (for z/OS)
- -> SAP KERNEL 6.20 64-BIT -> S390X_64 -> Database independent (for Linux on System z)

The 6.20 standalone enqueue server is compatible with the 4.6D kernel.

Application server instances were installed on z/OS and Linux on System z. Table 13 lists the software used for the application server on Linux.

Table 13. SAP application server for Linux on System z

Product name	Minimum level requirement	Recommended level
Linux on System z	SUSE Linux Enterprise Server 8 for System z (64 bit), SP3	See SAP Note 81737.
ICLI[1]	APAR OW53950	See SAP Note 81737.
Tivoli System Automation for Multiplatforms	1.2.0.6	2.1.1 (2.1 FP2)
z/VM (optional)	V4.3	V4.4 with APARs VM63282, VM63397, or higher supported level

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 running one SAP system on one server. However, one of the strengths of z/OS is the capability of running *multiple* SAP systems on one server.

One possible configuration is to run all production SAP systems on one server or Parallel Sysplex and all non-production SAP systems on another server or Parallel Sysplex. In this hypothetical configuration, each SAP system would normally consist of, among other things, its own DB2 subsystem, its own set of file systems, a large number of SMS VSAM data sets, and its own set of ICLI servers[1]. Some common questions that need answers include:

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

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

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

Concerning a choice of names for the components of one SAP system, we recommend that you use the 3-character SAP system identification <SID> as a prefix in the name of all the components for one SAP system wherever possible. We recommend using SAP as a prefix for all SAP resources not related to a specific SAP system. In Table 14 we list the recommended names of all z/OS-related components of an SAP system, along with how or where to define them. In Table 15 on page 119, we list the recommended names of all components of an individual SAP system that are defined within SA z/OS.

Table 14. Recommended names for all z/OS-related components of an SAP system

Component	Recommended name	Our name	How/where defined
DB2 address spaces	<SID>xMSTR <SID>xDBM1 <SID>xIRLM <SID>xSPAS[4] <SID>xDIST (if you run DDF/DB2 Connect) where x defines the data sharing member	D7XxMSTR D7XxDBM1 D7XxIRLM D7XxSPAS where x defines the data sharing member	PROCLIB member names
ICLI server[1] procedure names	<SID>ICLIx	REDICLIx	PROCLIB member
High Level Qualifier for SAP VSAM objects	<SID>SAP	SAPRED	IDCAMS

[4] DB2-established stored procedure address spaces (<SID>xSPAS) are deprecated with DB2 V8.

Table 14. Recommended names for all z/OS-related components of an SAP system (continued)

Component	Recommended name	Our name	How/where defined
High Level Qualifier for Shared HFS file systems	<SID>SHFS.<instance-name>	SAPRED.SHFS	MOUNT FILESYSTEM command
WLM definitions for service classes	<SID>HIGH, <SID>MED, <SID>LOW	SAPHIGH, SAPMED, SAPLOW	WLM ISPF panels
SMB Share Names	<SID>MNT, <SID>USR, SAPTRANS	SAPMNT, SAPUSR, SAPTRAN	DFS SMB setup
NFS Server procedure name	<SID>NFS or SAPNFS	MVSNFSSA	PROCLIB member
VIPA name for SCS	sap<sid>	sapred	TCP/IP DNS entry
VIPA name for saprouter	saproute	saproute	TCP/IP DNS entry
VIPA name for NFS server	<sid>nfs or sapnfs	sapnfs	TCP/IP DNS entry

Table 15. Recommended names for all components of an individual SAP system

Component	Recommended name	Our name
Jobname for enqueue server	<SID>ADMES	REDADMES
Jobname for enq. replication server	<SID>ADMER	REDADMER
Jobname for message server	<SID>ADMMS	REDADMMS
Jobname for gateway	<SID>ADMGW	REDADMGW
Jobname for syslog collector	<SID>ADMCO	REDADMCO
Jobname for syslog sender	<SID>ADMSE	REDADMSE
Jobname(s) for rfcoscol	<SID>ADMRx	REDADMR1
Jobname for saposcol	SAPOSCOL	SAPOSCOL
Jobname for saprouter	SAPROUTE	SAPROUTE
Jobnames for application server instances and their monitors	<SID>ADMnn	APPSRVnn

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 “Defining the SAP-related resources” on page 163:

Table 16. Naming conventions for SA z/OS resources

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

Table 16. Naming conventions for SA z/OS resources (continued)

Type of resource	Naming convention
Groups with sysplex scope	*PLEX
Jobnames for SAP RED	REDADM*
Jobnames for general SAP	SAP*

Tivoli System Automation for Multiplatforms

All resources (except those for application servers) that are generated by the mksap script (see Appendix F, “Detailed description of the sample Tivoli System Automation for Multiplatforms high availability policy for SAP,” on page 327) 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 XI system, 'XI_ABAP' and 'XI_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 'XI_ABAP' and an SAP system ID of EP0 is called 'XI_ABAP_EP0_ENQ_ES', and the IP address of the enqueue server is called 'XI_ABAP_EP0_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 ‘add-in’ dialog instances, this is normally `D<instance_number>` and for J2EE-only instances `J<instance_number>`.

For example, the group for an application server of the SAP system with PREF value XI_ABAP and an SAP system ID of EP0 running on host p570sa04 with instance directory D00 is called

```
XI_ABAP_EP0_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

```
XI_ABAP_EP0_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, it is not effective to make 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.

ARM policy

In case of an LPAR failure, ARM is needed to allow recovery of the ’failed’ DB2 subsystem on a different LPAR.

Our ARM policy is shown in Appendix C, “ARM policy,” on page 301. It is set up according to the following requirements:

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

The LIGHT option of the START DB2 command is available starting with DB2 Version 7. It lets you restart a DB2 data sharing member with a minimal storage footprint, and then terminate normally after DB2 frees retained locks.

For details about ARM and light restart, refer to the DB2 publication *Data Sharing Planning and Administration*.

DB2 Connect and ICLI

For planning information concerning the DB2 Connect setup and ICLI servers, see Chapter 3, “Architecture options and trade-offs,” on page 23.

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/<sid>/exe. For our configuration we defined the following directory for Linux:

```
/sapmnt/RED/Linux/exe
```

Figure 34 on page 123 shows the SAP directory structure and file systems for SCS. This is similar to the old central instance except that the instance name is different.

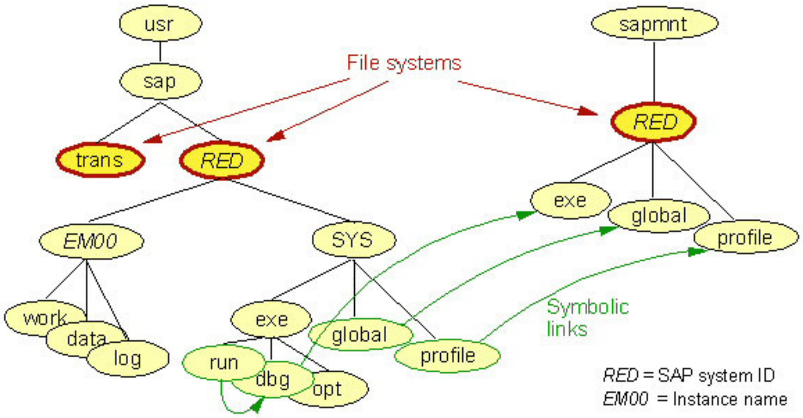


Figure 34. Directory tree

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

SAP directory definitions

The following directories must be defined:

SAP global transport directory

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

SAP system-wide directories

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

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

SAP local directories

On z/OS the directory /usr is a symbolic link to $VERSION/usr . This means that the contents of the /usr directory is different on every LPAR. This, however, is not practical for the /usr/sap directory. We propose to create the directory /sap in the root file system and to define symbolic links for /usr/sap to point to /sap. The symbolic links must be defined on each LPAR, i.e. in each $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.)

If you run multiple instances on z/OS belonging to the same SAP system, for example SCS and additional dialog instances, we recommend setting up multiple file systems. In our case, we run SCS ASCS00 and the dialog instances D10 and D11. Two additional file systems should be created and mounted on /usr/sap/RED/D10 and /usr/sap/RED/D11. While the ownership of /usr/sap/RED/ASCS00 is moved with SCS, /usr/sap/RED/D10 and /usr/sap/RED/D11 are always owned by the LPAR the instance is configured for. This ensures optimal separation and performance.

Administrator’s home directory

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

SAPOSCOL/RFCOSCOL directory

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

NFS server on z/OS

An SAP system setup for the high availability solution is capable of running on a server hosting other SAP systems. Also, you could define multiple HA SAP systems in one server or Parallel Sysplex. In such an environment, the question arises as to how many NFS servers are needed. If you separate the SAP systems and assign two or more LPARs to each SAP system, you will probably want to configure multiple NFS servers, one per SAP system, so that a failure of one NFS server does not affect other SAP systems. If you run several SAP systems on the same set of LPARs, it is sufficient to have one NFS server that serves multiple SAP systems.

Another reason for multiple NFS servers is that the SAF security option is not useful for server-to-server communication. Instead, you need the export file to provide standard UNIX security. However, if the number of available LPARs is limited and you consider exporting user file systems, you may choose to run multiple NFS servers on the same LPAR. One NFS server exports only the global SAP directories to a list of UNIX servers. The second NFS server uses SAF security to export user directories and let users authenticate with their password. Avoid running multiple NFS servers in one LPAR, because this would require multiple TCP/IP stacks. Run them in different LPARs.

To allow transparent failover of the NFS server, the mount handle databases must be shared between the systems. 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 protocol expects that the mount handles are still valid. The client is not aware of the reason for the failure.

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

The automated restart and failover capability of the NFS server requires attribute SECURITY(EXPORTS), i.e., default UNIX security. NFS clients cannot reconnect transparently to the application if SAF security is used. The reason is that the SAF authentication, which is acquired via mvslogin, is kept in memory. If the NFS server is stopped and restarted, the SAF authentication is lost and a running application on the client side receives an I/O error after it has reconnected because the permission is denied. The authentication can be reestablished later on with a new mvslogin. However, this cannot be accomplished transparently to the application.

Alternatively, you can run with the security attribute SECURITY(SAFEXP) for HFS data and define in the checklist file which entries for directories are to be exempt from SAF checking. This requires that the CHECKLIST option also be specified as a site attribute. For the SAFEXP security option, the EXPORTS requirements have to be met before the CHECKLIST requirements are examined.

Security considerations

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

NFS server on Linux on System z

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 207.

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 following z/OS resources should be defined to SA z/OS:

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

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

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.

Stop the system and re-IPL it. 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 11, “Customizing the Tivoli System Automation for Multiplatforms base component,” on page 199 and Chapter 12, “Customizing the Tivoli System Automation for Multiplatforms end-to-end component,” on page 221.

Control of remote ABAP application server instances

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

1. Via the strategic SA MP and end-to-end based solution. With the availability of SA MP, together with its end-to-end management automation component, we strongly recommend using SA MP with the end-to-end component to control remote ABAP application servers running on platforms supported by SA MP. This requires the installation and setup of SA MP and the end-to-end component.
2. Via the tactical solution using ssh/rsh. We created three shell scripts (see Appendix E, “Detailed description of the z/OS high availability scripts,” on page 315) 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 virtual IP address (VIPA) 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. Request an SAP license for each CEC that will host SAP Central Services. For further details, see SAP Note 538081 (“SAP Notes” on page 373) and “ABAP SAP Central Services (ASCS)” on page 139.

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 reconnection 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 ’add-in’ 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.

Chapter 8. Customizing SAP for high availability

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

The chapter covers the following:

- How to configure the ABAP and/or J2EE variants of SAP Central Services. Both SCS variants comprise 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/36/67973c3f5aff39e10000000a114084/content.htm`

Alternatively, you can go to

`http://help.sap.com`

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

Note: 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 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.

Installing and configuring ABAP SAP Central Services (ASCS)

ABAP SAP Central Services (ASCS) [5] 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 an 'add-in' system. For an 'add-in' system, we strongly recommend running both the ABAP SCS and J2EE 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 or on any other application server platforms supported by the SAP on System z solution and SA MP. For such a heterogeneous environment, we strongly recommend running the SA MP end-to-end management automation component.

Prior to SAP NetWeaver 2004s, the SAP installation routine (**SAPinst**) does 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 4.6D or 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 could, at a minimum, perform the installation steps for one dialog instance. This ensures that the parameters for UNIX System Services are

[5] See the note on page 91 concerning the abbreviations *SCS* and *ASCS*.

appropriate for SAP; the sap<sapsid> user environment is defined; and the standard SAP directory structure is created.

For details about preparation, refer to “SAP installation aspects” on page 127.

Getting the current standalone enqueue server code from SAP

Everything about high availability from the SAP point of view can be obtained at the SAP Service Marketplace:

`http://service.sap.com/ha`

This site requires registration, including a customer or installation number. To register, go to:

`http://service.sap.com`

Starting with SAP Web Application Server 6.20, the standalone enqueue server is included in the standard delivery. It is also compatible with kernel release 4.6D (refer to SAP Note 524816) and can be downloaded as a binary patch under the section SAP WEB AS 6.20 as described in “Software prerequisites” on page 116.

The following parts make up the highly available enqueue server:

enserver	Standalone enqueue server
enrepserver	Enqueue replication server
ensmon	Enqueue monitor

The package also contains the current version of the enqueue test tool ’enqt’, which you can use to verify the ASCS setup. Install the parts in the executable directory (SYS/exe/run).

Configuring ABAP SAP Central Services

ASCS requires a virtual host name in order to be ‘movable’. In our case we used the name 'sapred' and 172.20.10.1 as the associated IP address.

If your SAP system runs a 6.40 kernel, SAP Note 821904 describes how to separate the ABAP SCS from the Central Instance. Attached to 821904 are scripts that do the actual changes. Ensure that you set parameter `gateway_on_ascs` in the configuration file `createscs.cfg-scs` to 1, in order to start a standalone gateway with the generated start profile.

For 4.6D, or if you want to do the ASCS separation manually, follow the steps listed below.

The enqueue architecture represented by SAP Central Services (SCS) is activated by changing some profile parameters in the DEFAULT.PFL profile and the application server instance profiles; these parameters are described in “SAP profile parameters” on page 133.

We strongly recommend saving the DEFAULT.PFL profile and the application server instance profiles before starting to change them.

1. Create an instance profile that is used by all components that belong to ASCS. ASCS has its own instance number, instance name, instance directory, and virtual instance host name.

   We chose instance number 00, instance name 'ASCS00', and host name 'sapred'. The profile RED_ASCS00_sapred is shown in the following:

```
# Profile for enqueue server/message server/gateway/syslog collector...
SAPSYSTEMNAME = RED
INSTANCE_NAME = ASCS00
SAPSYSTEM = 00

enque/process_location = LOCAL
enque/server/replication = true
enque/server/threadcount = 3
enque/encni/repl_port = 6000                          [see note]
enque/backup_file = $(DIR_GLOBAL)/ENQBCK

ipc/shm_psize_34 = 0
```

Note: 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. If replication is set to true but the repl_port is not defined, the port of the gateway service will be used (3300 as described below). In this case, either the gateway service or the replication will fail, depending on which service can bind to the port.

2. Create the instance directory and its subdirectories. Create the directories locally on each node where the ASCS instance can run. In our case, the commands are:

```
mkdir /usr/sap/RED/ASCS00
mkdir /usr/sap/RED/ASCS00/data
mkdir /usr/sap/RED/ASCS00/log
mkdir /usr/sap/RED/ASCS00/work
```

3. Modify the DEFAULT.PFL profile. Prior to doing this, save the old DEFAULT.PFL (for example as DEFAULT.CentralInstance). This will allow you to easily fall back to the old architecture as described below. The following example shows the entries that need to be changed.

```
rdisp/mshost=sapred
rdisp/msserv = 3600
ms/server_port_0=PROT=HTTP, PORT=8100

rdisp/sna_gateway = sapred
rdisp/sna_gw_service = sapgw00

# enqueue server parameter for ABAP SCS
# rdisp/enqname = $(rdisp/myname)       this must be inactive!
enque/process_location = REMOTESA
enque/serverhost = sapred
enque/serverinst = 00

# the central system log collector runs in the ASCS instance
rslg/collect_daemon/host = sapred
rslg/collect_daemon/listen_port = 1400
rslg/collect_daemon/talk_port = 1500
rslg/send_daemon/listen_port = 1200
rslg/send_daemon/talk_port = 1300
```

Remember that ’sapred’ is the hostname of the virtual IP address that belongs to the ASCS.

4. Add the following parameters to all instance profiles:

```
enque/con_retries = 120
enque/dequeu_wait_answer = TRUE
```

In the 'old' CI instance profile, the following setting is required:

```
rdisp/wp_no_enq = 0
```

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

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

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

   The message server sapms<sapid> must also have an entry like:

   ```
   sapmsRED 3600/tcp # SAP System Message Port
   ```

6. If the CI is already installed on z/OS, you must remove the entries that start the message server and syslog collector daemon from the Central Instance Start profile START_DVEBMGS00_<CI host name>.

Verifying the ASCS changes

First, you must have activated the virtual hostname and its associated IP address belonging to the ASCS (sapred). Then, start all ASCS components. As <sapsid>adm, do:

1. Start the message server:

   ```
   msg_server pf=/usr/sap/RED/SYS/profile/RED_ASCS00_sapred &
   ```

2. Start the enqueue server:

   ```
   enserver pf=/usr/sap/RED/SYS/profile/RED_ASCS00_sapred &
   ```

3. Start the enqueue replication server:

   ```
   enrepserver pf=/usr/sap/RED/SYS/profile/RED_ASCS00_sapred &
   ```

Then you can verify manually that the SAP system is running correctly with ASCS by using SAP transaction SM12 to generate entries in the enqueue table (see “Preparation for the test (unplanned outage only)” on page 242).

Generating entries and displaying them must be possible when the enqueue server of ASCS is running, and not when it is stopped. Use

```
ensmon pf= <profile> -H <hostname>
```

for example:

```
ensmon pf=/usr/sap/RED/SYS/profile/RED_ASCS00_sapred -H sapred
```

to check that the replication server has successfully connected to the ABAP standalone enqueue server. From the main menu, run:

1. 'Dummy request'. This request should execute successfully.
2. 'Get replication information'. This should return

   ```
   "Replication is enabled in server, repl. server is connected
   Replication is active"
   ```

   and should also show statistics.

Tip

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

- Stop all SAP instances including ASCS.
- Save the newly created and adapted SAP profiles.
- Restore the saved old DEFAULT.PFL and application server instance profiles.
- Start the central instance (and optionally the dialog instances).

Any other change does not affect the ability to start SAP in the old way.

SAP profile parameters

The following table lists and describes the profile parameters that are related to ASCS.

Table 17. SAP profile parameters relevant for the high availability solution

Parameter	Description	Default value	Recommended value
enque/serverhost	Host of the enqueue server.		<virtual hostname>
enque/serverinst	Instance number of the enqueue server.		<instance number>
enque/process_location	Specifies where the enqueue requests are processed.	OPTIMIZE	REMOTESA (for application servers) LOCAL (for the enqueue server)
enque/server/replication	Enables replication.	false	true
enque/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>
enque/server/threadcount	Number of I/O threads in the enqueue server.	1	2 or 3

Table 17. SAP profile parameters relevant for the high availability solution (continued)

Parameter	Description	Default value	Recommended value
enque/deque_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.	$(DIR_LOGGING)/ ENQBCK	$(DIR_LOGGING)/ 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.	6	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<instnr>
rslg/collect_daemon/host	Host of the syslog collector.		<virtual hostname>
rdisp/enqname	Application server instance running the (old style) enqueue work process (obsolete).		# comment out
rdisp/btcname	Application server that does the event processing for the batch scheduler (obsolete; every instance can process its own events).		$(rdisp/myname)
rdisp/vbname	Application server that runs update work processes (obsolete; update requests are dispatched among appropriate instances automatically).		$(rdisp/myname)

Table 17. SAP profile parameters relevant for the high availability solution (continued)

Parameter	Description	Default value	Recommended value
SAPDBHOST	Location of the database server. (Forward reference to dbs/db2/hosttcp, which is defined in each instance profile to specify the primary database server of the instance, while SAPDBHOST is usually defined in DEFAULT.PFL.)		$(dbs/db2/hosttcp)
ipc/shm_psize_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.		0

Installing and configuring J2EE SAP Central Services

J2EE SAP Central Services can run under z/OS UNIX System Services or Linux on System z. For high availability reasons, we recommend running the J2EE SCS on z/OS. This recommendation holds for J2EE-only systems as well as for an ‘add-in’ system. For an ‘add-in’ system, we strongly recommend running both the ABAP SCS and J2EE SCS on z/OS. Preferably, you should also allocate SAP-required file systems on z/OS.

It is also possible to run J2EE SCS on Linux on System z or on any other application server platform supported by the SAP on System z solution and SA MP. For such a heterogeneous environment, we strongly recommend running the SA MP end-to-end management automation component.

SAPinst does not support the installation of a J2EE SCS instance under z/OS for SAP NetWeaver '04. Therefore, you must create the J2EE SCS instance manually under z/OS UNIX System Services. In order to do this, you must have a running SAP system based on SAP kernel version 6.40. SAP Note 821904 describes how to create a J2EE SCS under z/OS UNIX System Services. Attached to 821904 are scripts that do the actual changes. Ensure that you set the parameter `gateway_on_ascs` in the configuration file `createscs.cfg-scs` to 0, because there is no standalone gateway in the J2EE SCS.

After you have created the J2EE SCS instance under z/OS USS, check the created instance profile. It should have entries similar to those in profile RED_SCS01_siccps40:

```
# Profile for J2EE enqueue and message server...
SAPSYSTEMNAME = RED
INSTANCE_NAME = SCS01
SAPSYSTEM = 01

# message server port for J2EE SCS
rdisp/msserv = 3601
rdisp/mshost= siccps40
ms/server_port_0= PROT=HTTP, PORT=8101

enque/process_location = LOCAL
enque/serverhost = siccps40
enque/serverinst = 01
enque/backup_file = $(DIR_GLOBAL)/ENQBCK_J2EE

# entries to enable enque table replication
enque/server/replication = true
enque/encni/repl_port = 6001    [see note]

ipc/shm_psize_34 = 0
```

Comment out the `enque/encni/port` entry, since there is no gateway running for this instance.

Note: Ensure that the port number chosen for replication 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.

Additionally, you need to create the instance directory and its subdirectories as local directories on each z/OS system on which the J2EE SCS instance can run. In our case, the commands are:

```
mkdir /usr/sap/RED/SCS01
mkdir /usr/sap/RED/SCS01/data
mkdir /usr/sap/RED/SCS01/log
mkdir /usr/sap/RED/SCS01/work
```

Finally, check that the port names used for J2EE SCS are defined in `/etc/services` on all application server nodes. Otherwise, the application server instances will not be able to connect to the enqueue or message server.

Assuming that the instance number of J2EE SCS is 01, the following entry is needed:

```
sapdp01     3201/tcp
```

The message server sapms<sapid> must also have an entry such as:

```
sapmsRED 3601/tcp # SAP System Message Port
```

Verifying the J2EE SCS changes

At this point, you need to verify that your SAP system runs properly with the virtual hostname and replication server. You can start the J2EE SCS services manually under Linux as <sapid>adm:

1. Activate the virtual IP address that belongs to J2EE SCS, for example via the ifconfig command:

   ```
   ifconfig eth0:2 9.153.165.99 broadcast 9.153.165.255 netmask 255.255.255.0 mtu 1500 up
   ```

   Note: Check via `ifconfig –a` that alias eth0:2 is not already in use.

2. Start the message server:

   ```
   msg_server pf=/usr/sap/RED/SYS/profile/RED_SCS01_siccps40
   ```
3. Start the enqueue server:

   ```
   enserver pf=/usr/sap/RED/SYS/profile/RED_SCS01_siccps40
   ```
4. Start the enqueue replication server:

   ```
   enrepserver pf=/usr/sap/RED/SYS/profile/RED_SCS01_siccps40
   ```

You can also use

```
startsap r3 SCS01 siccps40
```

as an alternative to steps 2 and 3 above.

Afterwards, manually verify that the SAP system is running correctly. Use the SAP-provided `enqt` utility. To use `enqt`, you need to create a special profile to be able to connect to the J2EE enqueue server. This is necessary because the J2EE application servers read their profile parameters from the *configuration database* and not from a profile in directory

```
/usr/sap/<SID>/SYS/profile
```

The predefined SA MP policy for SAP comes with a sample file named `enqt.pf_scs01`. You must adapt the following entries to your environment:

```
SAPSYSTEM        = 01
SAPSYSTEMNAME    = LOP

enque/serverhost = siccps40

enque/serverinst = 01
# enque/encni/port = 3201
```

Then, as <sapid>adm, run the following from the SA MP predefined policy installation directory (with your adapted `enqt` profile):

```
enqt pf=enqt.pf_scs01 97
```

You will see output like:

```
---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. 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.

Use `ensmon pf= <profile> -H <hostname>`, for example:

```
ensmon pf=/usr/sap/RED/SYS/profile/RED_SCS01_siccps40 -H siccps40
```

to check that the replication server has successfully connected to the J2EE standalone enqueue server. From the main menu run:

1. 'Dummy request'. This request should execute successfully.
2. 'Get replication information'. This should return

   ```
   "Replication is enabled in server, repl. server is connected
   Replication is active"
   ```

   and should also show statistics lists.

Your J2EE SCS setup for high availability is now complete, and you can start to create the SA MP resources for your SAP system.

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 E, “Detailed description of the z/OS high availability scripts,” on page 315.

C-shell and logon profiles

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

The C-shell knows four profiles:

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

When the -c option is used, the files /etc/csh.login and $HOME/.login are *not* processed. This is the case when programs are invoked via BPXBATCH in a started task, or via the 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.

ICLI servers

For ICLI only

If you still run ICLI servers[1], they can be started by using a shell script (iclistart), or by submitting a job, or by invoking a started task. We decided to use started tasks. For each ICLI server, we created a separate procedure.

If you are using the iclistart shell script to start the ICLI server, take out the *nohup* and the *&* and add the console message as the last line.

```
export ICLI_TRACE_LEVEL=0
export STEPLIB='DB7X7.SDSNEXIT:DB7X7.SDSNLOAD'
/usr/sbin/fome46ds -PLAN FOMEP46D -LOGDIR /usr/sap/RED/icli/icli6 -PORT 5006
echo "$_BPX_JOBNAME ENDED" > /dev/console
```

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

Started tasks or UNIX shell scripts

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

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

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

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

End of For ICLI only

ABAP SAP Central Services (ASCS)

ASCS is a collection of single-instance SAP resources that share the same instance profile and instance directory:

- ABAP enqueue server
- ABAP message server
- Gateway server
- Syslog collector
- Syslog sender

In order to allow transparent failover of the ASCS to another system, the enqueue replication server must run 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 startsap_em00 shell script. See Appendix E, "Detailed description of the z/OS high availability scripts," on page 315 for a detailed description.

The shell script must be adapted to your environment.

The individual components are started as follows:

Table 18. Using startsap_em00

Command	Result
startsap_em00 ES	Starts the ABAP enqueue server
startsap_em00 ERS	Starts the ABAP enqueue replication server
startsap_em00 MS	Starts the ABAP message server
startsap_em00 GW	Starts the gateway
startsap_em00 CO	Starts the syslog collector
startsap_em00 SE	Starts the syslog sender
startsap_em00 CHECK	Performs a health check on the ABAP enqueue server. This implicitly tests the validity of the SAP license of the system where the ABAP enqueue server is running.

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.

J2EE Central Services

J2EE SCS is a collection of single-instance SAP resources that share the same instance profile and instance directory:

- J2EE enqueue server
- J2EE message server

In order to allow transparent failover of the J2EE SCS to another system, the enqueue replication 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 startsap_jscs shell script. See Appendix E, "Detailed description of the z/OS high availability scripts," on page 315 for a detailed description.

The shell script must be adapted to your environment.

The individual components are started as follows:

Table 19. Using startsap_jscs

Command	Result
startsap_jscs ES	Starts the J2EE enqueue server
startsap_jscs ERS	Starts the J2EE enqueue replication server
startsap_jcsc MS	Starts the J2EE message server
startsap_jscs CHECK	Performs a health check on the J2EE enqueue server. This implicitly tests the validity of the SAP license of the system where the J2EE enqueue server is running. **Important:** The statements under "ABAP SAP Central Services (ASCS)" on page 139) concerning the SAP license also apply.

ABAP application server instances

We created three shell scripts (see Appendix E, "Detailed description of the z/OS high availability scripts," on page 315) that allow you to start, stop and check local (remote) ABAP application server instances. With the availability of SA MP, together with its end-to-end management automation component, we strongly recommend using SA MP with the end-to-end component to control remote ABAP application servers running on platforms supported by SA MP.

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 E, "Detailed description of the z/OS high availability scripts," on page 315. The necessary adaptations for the 6.40 and 4.6D kernel versions are also described therein. The host name (<hostname>), instance number (<instnr>), and instance directory (<instancedir>) identify the instance to be managed.

Starting and stopping an ABAP 6.40 application server instance requires the <instancedir> parameter if more than one instance is running, and this is always the case in which ASCS and an application server are running.

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.

The scripts can be used for both local z/OS ABAP application servers and remote ABAP application servers. For remote ones, as mentioned above, we strongly recommend using SA MP and its end-to-end management automation component to control the application servers on all platforms supported by SA MP.

What the shell scripts do

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

startappsrv_v5:

- For a local 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 the SAP utility rfcping (see “rfcping” on page 143). 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.

  With 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 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.

  If the <hostname> matches the local host name, the commands are executed directly. Otherwise, a remote execution is performed (see “Remote execution” on page 143).
- 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>
  ```

  If the <hostname> matches the local host name, the command is executed directly. Otherwise, a remote execution is performed (see “Remote execution” on page 143).

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” on page 143.

  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.

rfcping: This utility is part of the SAP 6.40 (or higher level) kernel and can be downloaded as a binary patch from the SAP Service Marketplace, section SAP WEB AS 6.40 (or higher level). The version is compatible with previous SAP releases.

rfcping 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.

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

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 is not recommended for these platforms.

Remote execution using the *rexec* command is not recommended for these platforms, because it implies 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 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
```

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 E, "Detailed description of the z/OS high availability scripts," on page 315 for details.

J2EE and add-in application server instances

J2EE and 'add-in' application server instances run only on non-z/OS systems. We strongly recommend using SA MP with the end-to-end component to control such instances and do not provide sample USS shell scripts for them.

saposcol

Normally, the SAP utility saposcol is started implicitly (outside of Tivoli System Automation control) when you start the first application server instance on a machine via the SAP-provided startsap script. However, it is not stopped automatically when all application servers are stopped. It keeps running and holds a file handle to the SAP executable directory. For example, if you have stopped all SAP resources via Tivoli System Automation and you want to unmount the SAP executable directory, this open file handle from saposcol causes the umount command to fail.

Therefore, we strongly recommend starting and stopping `saposcol` directly by Tivoli System Automation. In this way, you avoid problems such as those described. Remove the invocation of saposcol that is done in the standard SAP start scripts, and instead leave the starting and stopping of saposcol solely to Tivoli System Automation.

For 4.6D, you must edit, in the home directory of <sapid>adm, the startsap_<hostname>_<instnr> shell scripts for the application server instance(s) that are *running on z/OS*. Comment out the following line:

```
start_saposcol;
```

As of 6.40, you must edit the startsap shell script in the SAP executable directory. Comment out all lines

```
'start_saposcol'
```

and change all lines

```
echo "start_saposcol;"
```

to

```
echo "NO start_saposcol;"
```

In a z/OS sysplex environment, it is sufficient to run saposcol once, because it gathers the RMF data from all the z/OS LPARs in the sysplex in any event. For this case, configure an additional rfcoscol as described under "Additional rfcoscol" on page 145, put both into a basic group, and add this group to a sysplex move group. See "Alternative: Sysplex-wide SAPOSCOL" on page 188. You do not need to create the application group SAP_LOCAL.

rfcoscol

The SAP utility rfcoscol[6] is started with the following shell script ("`start_rfcoscol`"):

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

The corresponding RFC definition file (`saprfc.ini`) in our case is located in /usr/sap/RED/rfc; the following shows the entries:

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

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

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

Additional rfcoscol

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

```
#!/bin/sh
export SAP_CODEPAGE=1100
export RFCOSCOL_RETRY=1
cd /usr/sap/RED/rfc
ln -sf /usr/sap/RED/rfc/rfcoscol rfcoscol_FCSPLEX
./rfcoscol_FCSPLEX -RFC -a fplexrfc -g sapred -x sapgw00
echo "$_BPX_JOBNAME ENDED" > /dev/console
```

The meanings of the other command line parameters are:

```
-a <Program ID>
-g <Gateway host>
-x <Gateway service name>
```

[6] As of NetWeaver 2004s, the function of rfcoscol is replaced by the stored procedure SAPCL.

Make sure that the rfcoscol instances are started with different PROGIDs, since they must register at the gateway with a unique PROGID.

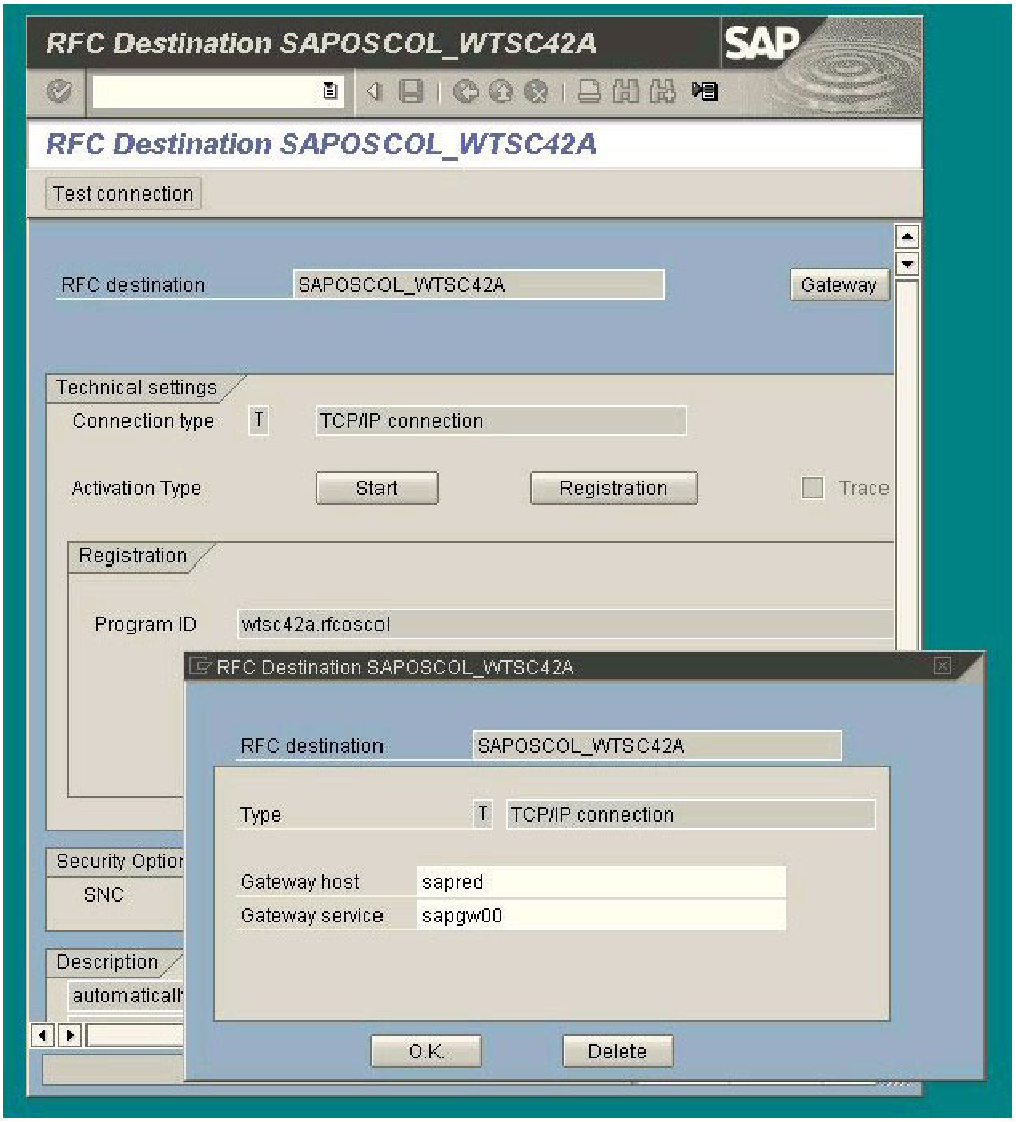
Additional SAP setup for RFC connections

Because the standalone gateway server that is started as part of 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.



RFC Destination SAPOSCOL_WTSC42A

RFC destination SAPOSCOL_WTSC42A

Connection type T TCP/IP connection

Program ID wtsc42a.rfcoscol

RFC Destination SAPOSCOL_WTSC42A

RFC destination SAPOSCOL_WTSC42A

Type T TCP/IP connection

Gateway host sapred

Gateway service sapgw00

Figure 35. Defining the gateway host for rfcoscol with SAP transaction SM59

Special considerations for z/OS application servers running the 6.40 kernel

The SAP-provided ’startsap’ script of SAP kernel version 6.40 starts the ’DB2 Performance data collector’ when running on z/OS. This eventually starts an rfcoscol with the parameters:

```
-a <host name>.rfcoscol
-g <host name>
-x sapgw<application server instance number>
```

In order to automatically stop this collector, add the code contained in file `640stopsapCodeForzOS.txt` to the 6.40 ’stopsap’ script provided by SAP and located in the

```
/usr/sap/<SAPSID>/SYS/exe/run
```

directory.

SAProuter

The SAProuter an 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 20 summarizes the start, stop and monitoring commands that are needed when you set up the Tivoli System Automation policies for SAP.

Table 20. Summary of start/stop monitoring commands

Actions	**Value or command**
SAP system name	RED
User ID	redadm
Home directory	/u/redadm
ICLI servers:[1]	
- start command (started task)	S REDICLI# (# = 6, 7, 8, 9)
- stop command	F REDICLI#,APPL=STOP TIMEOUT(60)
VIPA for ASCS	
- start command (started task)	S TCPVIPA,VIPA=172.20.10.1
Enqueue server:	
- start command	/u/redadm/startsap_em00 ES
- stop command	/bin/kill -2 %PID%
- process name to be monitored	./es.sapRED_ASCS00
- additional monitor	/u/redadm/startsap_em00 CHECK
Enqueue replication server:	
- start command	/u/redadm/startsap_em00 ERS
- stop command	/bin/kill -2 %PID%
- process name to be monitored	./ers.sapRED_ASCS00
Message server:	
- start command	/u/redadm/startsap_em00 MS
- stop command	/bin/kill -2 %PID%
- process name to be monitored	./ms.sapRED_ASCS00
Gateway:	
- start command	/u/redadm/startsap_em00 GW
- stop command	/bin/kill -2 %PID%
- process name to be monitored	./gw.sapRED_ASCS00

Table 20. Summary of start/stop monitoring commands (continued)

Actions	Value or command
Syslog collector:	
- start command	/u/redadm/startsap_em00 CO
- stop command	/bin/kill -2 %PID%
- process name to be monitored	./co.sapRED_ASCS00
Syslog sender:	
- start command	/u/redadm/startsap_em00 SE
- stop command	/bin/kill -2 %PID%
- process name to be monitored	./se.sapRED_ASCS00
Application server instances:	
- start command	/u/redadm/startappsrv_v5 <hostname> <instnr> <instancedir> [<via>]
- poststart (monitor) command	/u/redadm/checkappsrv_v4 <hostname> <instnr>
- stop command	/u/redadm/stopappsrv_v5 <hostname> <instnr> <instancedir> [<via>]
- process name to be monitored	/usr/sap/RED/rfc/rfcping_<hostname>_<instnr>
saposcol:	
- start command	/usr/sap/RED/SYS/exe/run/saposcol -l
- stop command	/bin/kill -2 %PID%
- process name to be monitored	/usr/sap/RED/SYS/exe/run/saposcol
rfcoscol:	
- start command	/u/redadm/start_rfcoscol
- stop command	/bin/kill -2 %PID%
- process name to be monitored	/usr/sap/RED/SYS/exe/run/rfcoscol
Additional rfcoscol:	
- start command	/u/redadm/start_rfcoscol_FCSPLEX
- stop command	/bin/kill -2 %PID%
- process name to be monitored	./rfcoscol_FCSPLEX
VIPA for saprouter:	
- start command (started task)	S TCPVIPA,VIPA=172.20.10.3
saprouter:	
- start command	/usr/sap/RED/SYS/exe/run/saprouter -r
- stop command	/usr/sap/RED/SYS/exe/run/saprouter -s
- process name to be monitored	/usr/sap/RED/SYS/exe/run/saprouter

Chapter 9. Change management

This chapter discusses the procedures to update the SAP kernel and the ICLI server and client in the environment presented in this book. 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.

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 153 for more information.

Note: A rolling kernel upgrade is not yet available for SAP Web Application Server 6.xx.

Updating the SAP kernel (release 4.6 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 dispatcher

If the dispatcher (disp+work) or one of its dynamically loaded modules (dbdb2slib.*, ibmiclic.*)[7] 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 (enrepserver), perform these steps:

[7] With 6.20, the kernel is split into several load modules. The following dynamic load modules also belong to the kernel: dw_xml.*, dw_stl.*, dw_xtc.*.

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.

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 684835, ″Availability of Rolling Kernel Upgrades for 4.6D_EXT″, for further information.

Each patch of the 4.6D_EXT kernel is classified by two numbers, the *update level* and the *patch number*:

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 dispatcher" on page 152.

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 SAP 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.

Updating the ICLI client and server

For ICLI only

The ICLI client and server for a given SAP kernel release are characterized by two-level versioning:

1. Protocol version

2. Internal version

As long as the protocol version remains the same, then the ICLI server and client can be upgraded independently. The objective of the ICLI development team is to always keep the protocol compatible within one SAP kernel release. In fact, for 4.6D, the protocol version has never been changed.

The versions are displayed when invoking the ICLI server with the command line parameter -HELP. You also can find the protocol version of the current ICLI server in the log file (message ICLS1011I).

Note: The descriptions of rolling ICLI client and server upgrades given in the following sections apply as long as the protocol versions are the same.

Rolling upgrade of the ICLI client

The ICLI client can be downloaded and updated when the SAP kernel is updated (refer to “Updating the SAP kernel” on page 151).

If, for any reason, only the ICLI client is to be updated, proceed as follows. This procedure is valid for UNIX systems and has been tested on AIX 5.1 and on Linux on System z.

1. Save the old ICLI client.
2. Download the new client and adjust the permission bits, as described in the respective *Planning Guide*.
3. Choose *one* of the following options to restart the work processes.
   a. Restart the application server instance.
   b. Restart the work processes via SAP transaction SM50. In this transaction, the work processes are restarted by selecting Process -> Cancel without core.
   c. Wait for the automatic restart of the work processes according to the respective SAP profile parameters. The following parameter settings mean that the work processes are started once a day (as suggested in SAP Note 182207).

   ```
   rdisp/wp_auto_restart = 86400
   rdisp/noptime = 87000
   ```

   Each work process continues to run with the old ICLI client until it is restarted. When the work process restarts, it loads the new ICLI client.

For Windows, such a mechanism is not available, and a restart of the SAP instance is unavoidable.

Rolling upgrade of the ICLI server

The following method describes how to perform a rolling ICLI server update.

1. Apply the ICLI PTF.

   For SAP kernel releases through 6.20, the ICLI server load module resides in an APF-authorized library. By default it is SYS1.LINKLIB. The corresponding USS executable has the sticky bit turned on, which means that a load module with the same name is searched for in the usual z/OS library concatenation.

   When using the ICLI client/server with the downward compatible 6.40 SAP kernel, there is no load module `FOME640S` in SYS1.LINKLIB. Instead, there is a UNIX System Services executable `fome640s` in the HFS directory `/usr/sbin`.

This is a symbolic link to `/usr/lpp/icli/sbin/fome640s`, which has the extended attribute set for the APF authorization. See ″ICLI client and server″ in the *6.20 Planning Guide*.

2. Perform a DB2 bind for the new packages without binding the plan.

   For this step, take the sample bind job and remove the BIND PLAN step so that you only bind the packages. The DBRMs that come with the new ICLI have a unique version that corresponds to the new ICLI server version. Therefore, the new DBRMs can be bound to packages while the old ICLI server is running. The ICLI plan does not need to be re-bound in this process and, at any rate, it would not be possible to bind it while it is in use.

   More information on DB2 binding for the ICLI server can be found in the *6.20 Planning Guide*.

3. Stop the ICLI servers and start them again.

   In a data sharing environment use the same mechanism explained in “Updating DB2 or z/OS” on page 156 to:

   a. Stop any batch processing on application servers connected to this ICLI server
   b. Move any SAP work away from the ICLI server that is being updated
   c. Stop and restart the ICLI server. If you want to take advantage of the upgraded ICLI server immediately, you must repeat the two preceding steps on the application servers that you previously moved. This will re-connect them to the upgraded ICLI server.

   With SA, use the STOP command to stop the ICLI servers. In the sample policy, the STOP command uses the smooth stopping method that waits up to 60 seconds to allow a running transaction to complete its unit of work. Cancel the STOP vote and SA will automatically restart the ICLI servers.

Updating an ICLI server with a new protocol version

If the protocol version has changed, you can follow these steps to upgrade the ICLI client and server at the same time.

1. Apply the new ICLI PTF.
2. Perform a DB2 bind for the new packages without binding the plan.
3. Save the old ICLI client.
4. Download the new client and adjust the permission bits.

   If you use more than one application server platform, repeat step 3 and 4 for each of them accordingly.
5. Stop the application servers.
6. Stop and restart the ICLI servers.
7. Restart the application servers.

End of For ICLI only

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 SAP 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 that you can run multiple software releases in a Parallel Sysplex data sharing environment. z/OS supports n+3 releases. This means that four consecutive releases can run in the same Parallel Sysplex, for example z/OS 1.4, 1.5, 1.6, and 1.7

DB2 supports n+1 releases. For example, DB2 V7 can run in the same data sharing group. However, DB2 UDB for z/OS V8 can run in parallel with DB2 UDB V7 in a data sharing group only as long as DB2 V8 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 the SAP sysplex failover feature. The following steps should be used for each LPAR to be updated:

1. Build new DB2 loadlibs with the DB2 maintenance applied for each DB2 data sharing member.
   A suggested name would be
   <db2 member name>.SDSNLOAD.NEW
2. Stop all SAP 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 application server. In order to do this, you must have set up an operation mode without any batch work processes.
3. Activate SAP sysplex failover by using SAP transaction 'DB2' to move each application server away from the LPAR that is going to be updated.
4. Stop the DB2 data sharing members in the LPAR.
   Issue a DB2 Display Thread command to ensure that there are no active connections to this DB2 member before issuing the Stop DB2 command.
5. Switch from current DB2 loadlibs to new DB2 loadlibs.
   This can be accomplished by renaming the loadlibs as follows:
   RENAME D7X1.SDSNLOAD to D7X1.SDSNLOAD.OLD
   RENAME D7X1.SDSNLOAD.NEW to D7X1.SDSNLOAD
6. At this point, z/OS can be stopped and re-IPLed to activate z/OS updates.
7. Restart the DB2 data sharing members in the LPAR.
8. Restart any ICLI servers that were previously stopped.
9. Switch back to the normal configuration using SAP transaction DB2.
10. Restart all SAP batch processing on application servers connected to DB2 in this LPAR. Use “Opt Mode Switch” to add batch work processes.
11. Repeat step 1 through step 10 for each LPAR in the sysplex.

Change management

Part 4. Autonomic operation of the high availability solution for SAP

Chapter 10. 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 and formerly known as System Automation for OS/390, or SA OS/390) for the high availability solution for SAP.

Note that, along with the following instructions, detailed knowledge of SA z/OS is required to make SAP high availability work.

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 126. 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 the systems 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" 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
```

If the region size of the NetView started procedure is too small, you may receive the following error message:

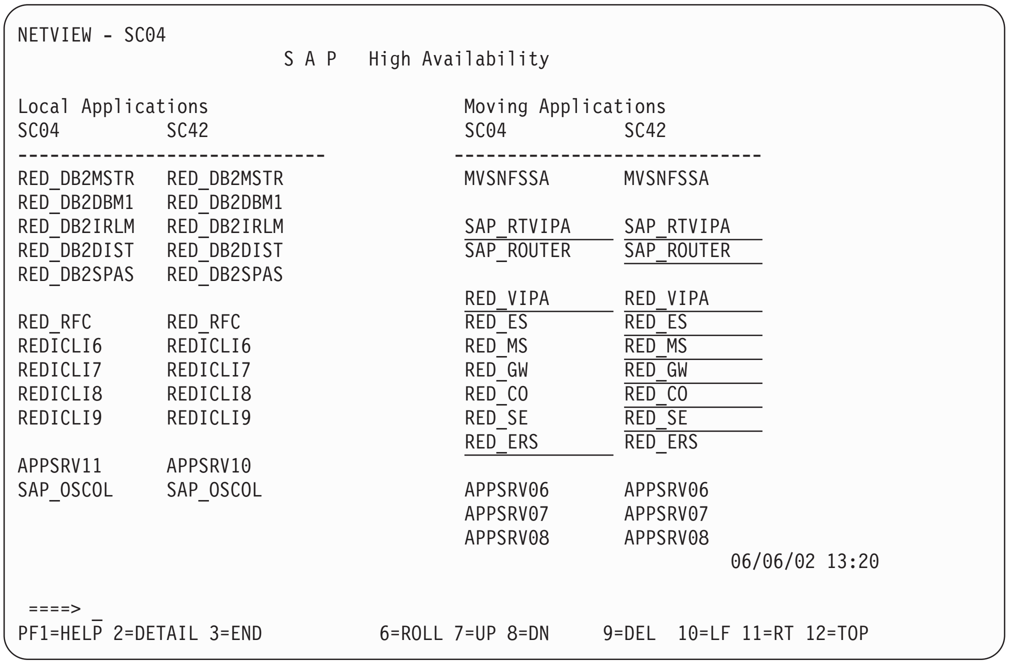
```
EA995I SYMPTOM DUMP OUTPUT
USER COMPLETION CODE=4091 REASON CODE=0000000C
TIME=14.34.23 SEQ=05730 CPU=0000 ASID=00D1
PSW AT TIME OF ERROR 078D1000 89E3555A ILC 2 INTC 0D
NO ACTIVE MODULE FOUND
NAME=UNKNOWN
DATA AT PSW 09E35554 - 00181610 0A0D47F0 B10A1811
AR/GR 0: 153B8498/84000000 1: 00000000/84000FFB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

Customizing the Status Display Facility (SDF)

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

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

The following depicts our SDF panel AOFSAP.



```
NETVIEW - SC04
                         S A P   High Availability

Local Applications                        Moving Applications
SC04          SC42                        SC04           SC42
-----------------------------            -----------------------------
RED_DB2MSTR   RED_DB2MSTR                 MVSNFSSA       MVSNFSSA
RED_DB2DBM1   RED_DB2DBM1
RED_DB2IRLM   RED_DB2IRLM                 SAP_RTVIPA     SAP_RTVIPA
RED_DB2DIST   RED_DB2DIST                 SAP_ROUTER     SAP_ROUTER
RED_DB2SPAS   RED_DB2SPAS
                                          RED_VIPA       RED_VIPA
RED_RFC       RED_RFC                     RED_ES         RED_ES
REDICLI6      REDICLI6                    RED_MS         RED_MS
REDICLI7      REDICLI7                    RED_GW         RED_GW
REDICLI8      REDICLI8                    RED_CO         RED_CO
REDICLI9      REDICLI9                    RED_SE         RED_SE
                                          RED_ERS        RED_ERS
APPSRV11      APPSRV10
SAP_OSCOL     SAP_OSCOL                   APPSRV06       APPSRV06
                                          APPSRV07       APPSRV07
                                          APPSRV08       APPSRV08
                                                                   06/06/02 13:20

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

Our definitions, including the new SDF panel AOFSAP, the modified SDF tree definition member AOFTSC04, and the modified SDF start screen AOFPSYST, can be found in “Status Display Facility definition” on page 303. These samples can be used as a base to build your own customized SDF panel.

A detailed description of how to customize SDF can be found in the SA z/OS *Programmer's Reference* (see Table 48 on page 370). Of course, you can also use the NetView Management Console (NMC) to monitor SAP application status.

Sending UNIX messages to the syslog

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

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

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

Setting MAXFILEPROC in BPXPRMxx

The USS parameter MAXFILEPROC, which is defined in the member BPXPRMxx of the PARMLIB, should be set to a reasonable value, such as 1000. It must not be set to the maximum of 65,536.

| This parameter influences the size of the file table that is allocated in each UNIX
| process. If the value is too high, SA z/OS will not be able to issue multiple
| *INGUSS* commands in parallel; the *INGUSS* commands will fail with an error
| message saying that a resource is temporarily not available. The *Planning Guides*
| recommend that you use the default value, which is 1000.

Defining the SAP-related resources

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

Notes:

1. We provide our SA z/OS policy database; for information on how to retrieve it, refer to Appendix E, "Detailed description of the z/OS high availability scripts," on page 315.
2. For the naming conventions used, see "Conventions used in the SA z/OS policy" on page 119

Overview of the resources

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

- Resources that are related to a specific SAP system (in our case, RED):
  - Database server
  - SCS, including enqueue server, message server, gateway, syslog collector, and syslog sender
  - Enqueue replication server
  - Application servers (both local and remote)
  - Local applications: rfcoscol and, if you still need to run them, ICLI servers
- Resources that are common to all the SAP systems:
  - NFS server
  - saprouter
  - Local applications: saposcol

Classes

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

In our environment, we used three classes:

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

USS_APPLICATION

This class is provided with the sample policy database of SA 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.

Definition:

```
Entry Name: USS_APPLICATION
Object Type: CLASS

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

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

CLASS_DB2_MASTER

This class is used for defining the DB2 master address space for all DB2 subsystems running in the sysplex.

Definition:

```
Entry Name: CLASS_DB2_MASTER
Object Type: CLASS

Relationships
HASPARENT JES2/APL/=

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

Shutdown NORM
1
INGRDTTH &SUBSAPPL S

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

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

CLASS_RED_DB2_CHILDS

This class[8] is used for defining the subordinate DB2 address spaces (DBM1, DIST, IRLM and SPAS) for the DB2 subsystem related to SAP RED.

The subordinate resources are defined for monitoring purposes only. Therefore, they are defined with the attributes “External startup” and “External shutdown” set to ALWAYS.

Definition:

```
Entry Name: CLASS_RED_DB2_CHILDS
Object Type: CLASS

Automation Info
External Startup . ALWAYS
External Shutdown. ALWAYS

Relationships
HASPARENT . . . RED_DB2MSTR/APL/=
Condition . . . StartsMeAndStopsMe
```

Database server

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

[8] At the time of this writing, the class name was limited to 20 characters. Therefore, ″CHILDS″ was selected instead of ″CHILDREN″.

System definition

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

The following shows how the name of the DB2 subsystem D7X2 is defined in the &AOCCLONE2 variable of system SC04.

```
Entry Type : System               PolicyDB Name   : SAP_HA_SAP
Entry Name : SC04                 Enterprise Name : SAP_HA

Operating system . . . . . MVS        MVS VM TPF VSE CF LINUX

Specify information (MVS systems only):
   MVS SYSNAME. . . . . . . . SC04       MVS system name

   Clone Id . . . . . . . . . 04             &AOCCLONE.
   Clone Id 1 . . . . . . . .                &AOCCLONE1.
   Clone Id 2 . . . . . . . . D7X2           &AOCCLONE2.
```

Applications

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

RED_DB2MSTR: This application corresponds to the DB2 MSTR address space. The following shows the definition of the application RED_DB2MSTR.

Note: We require that the Automatic Restart Manager (ARM) recover from DB2 failures, because with ARM you can easily exploit the DB2 LIGHT(YES) start option (see “ARM policy” on page 121). Therefore, we set the critical threshold number to 1 to tell SA z/OS not to recover the resource.

Definition:

```
Entry Name: RED_DB2MSTR
Link to Class CLASS_DB2_MSTR

Application Information
Application Type. . . DB2
Subtype . . . . . . . MSTR
Clone Job Name. . . . YES
Job Name. . . . . . . &AOCCLONE2.MSTR

Automation Information
Command Prefix. . . . -&AOCCLONE2.

Thresholds
        Critical            Frequent         Infrequent
Resource     Number   Interval   Number  Interval   Number   Interval
RED_DB2MSTR    1       01:00       1      05:00       1       24:00
```

RED_DB2DBM1: This application corresponds to the DB2 DBM1 address space.

Definition:

```
Entry Name: RED_DB2DBM1
Link to Class CLASS_RED_DB2_CHILDS

Application Information
Application Type. . . DB2
Subtype . . . . . . . DBM1
Clone Job Name. . . . YES
Job Name. . . . . . . &AOCCLONE2.DBM1
```

RED_DB2DIST: This application corresponds to the DB2 DIST address space.

```
Entry Name: RED_DB2DIST
Link to Class CLASS_RED_DB2_CHILDS

Application Information
Application Type. . . DB2
Subtype . . . . . . . DIST
Clone Job Name. . . . YES
Job Name. . . . . . . &AOCCLONE2.DIST

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

RED_DB2IRLM: This application corresponds to the DB2 IRLM address space.

```
Entry Name: RED_DB2IRLM
Link to Class CLASS_RED_DB2_CHILDS

Application Information
Application Type. . . DB2
Subtype . . . . . . . IRLM
Clone Job Name. . . . YES
Job Name. . . . . . . &AOCCLONE2.IRLM
```

RED_DB2SPAS: The application RED_DB2SPAS corresponds to the DB2 SPAS address space.

```
Entry Name: RED_DB2SPAS
Link to Class CLASS_RED_DB2_CHILDS

Application Information
Application Type. . . DB2
Subtype . . . . . . . SPAS
Clone Job Name. . . . YES
Job Name. . . . . . . &AOCCLONE2.SPAS
```

Application groups

After having defined the applications, we group them as shown in Figure 36 on page 168. One DB2 subsystem is to be active on each LPAR (active applications are represented as shaded boxes).

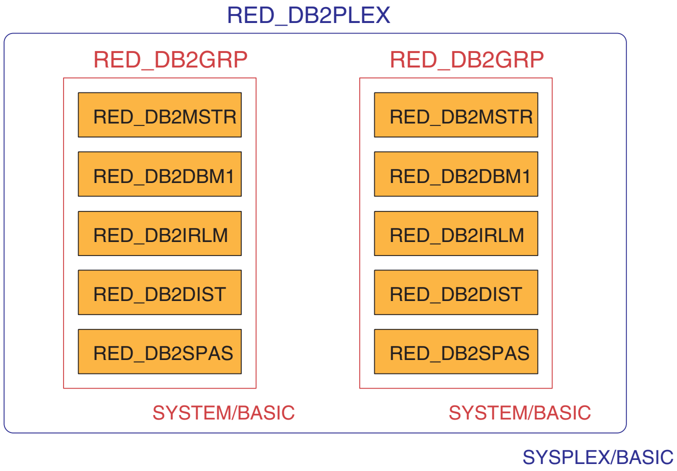


Figure 36. RED_DB2PLEX application group

RED_DB2GRP: This SYSTEM group combines the DB2 applications on a single system.

Definition:

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

Select applications:
RED_DB2DBM1
RED_DB2DIST
RED_DB2IRLM
RED_DB2MSTR
RED_DB2SPAS

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

RED_DB2PLEX: This superior application group is of scope SYSPLEX. It determines that the application group RED_DB2GRP is to be activated on the two specified systems SC04 and SC42.

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

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

SAP Central Services and the enqueue replication server

In this section, we provide the definition of an ABAP SCS. And because it is closely related, we also describe the definition of the enqueue replication server. The setup of a J2EE SCS is similar to that of an ABAP SCS.

Applications

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

RED_ES: This application corresponds to the enqueue server.

```
Entry Name: RED_ES
Link to Class USS_APPLICATION

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

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

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

Thresholds
                    Critical           Frequent         Infrequent
Resource     Number  Interval   Number  Interval   Number  Interval
   RED_ES         1      01:00       1      02:00       1      12:00
USS Control
User ID. . . . . . . REDADM
Command/Path . . . . ./es.sapRED_ASCS00
```

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 will be triggered whenever the enqueue server terminates.

RED_MS: This application corresponds to the message server.

```
Entry Name: RED_MS
Link to Class USS_APPLICATION

Application Information
Application Type. . . USS
Job Name. . . . . . . REDADMMS

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

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

USS Control
User ID. . . . . . . REDADM
Command/Path . . . . ./ms.sapRED_ASCS00
```

RED_CO: This application corresponds to the syslog collector. The purpose of the relationship definitions is explained in “RED_COPLEX” on page 175.

Definition:

```
Entry Name: RED_CO
Link to Class USS_APPLICATION

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

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

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

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

USS Control
User ID. . . . . . . REDADM
Command/Path . . . . ./co.sapRED_ASCS00
```

RED_SE: This application corresponds to the syslog sender.

```
Entry Name: RED_SE
Link to Class USS_APPLICATION

Application Information
Application Type. . . USS
Job Name. . . . . . . REDADMSE

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

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

USS Control
User ID. . . . . . . REDADM
Command/Path . . . . ./se.sapRED_ASCS00
```

RED_GW: This application corresponds to the SAP gateway.

```
Entry Name: RED_GW
Link to Class USS_APPLICATION

Application Information
Application Type. . . USS
Job Name. . . . . . . REDADMGW

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

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

USS Control
User ID. . . . . . . REDADM
Command/Path . . . . ./gw.sapRED_ASCS00
```

RED_VIPA: This application corresponds to the VIPA associated with SCS.

Definition:

```
Entry Name: RED_VIPA

Application Information
Application Type. . . STANDARD
Job Name. . . . . . . TCPVIPA1
JCL Procedure Name. . TCPVIPA

Application Automation Definition
Job Type. . . . . . . TRANSIENT
Transient Rerun . . . YES

Startup
   Parameters. . . . . . ,VIPA='172.20.10.1'

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

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

RED_ERS: This application corresponds to the enqueue replication server.

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

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

The INGGROUP commands in the application automation definitions of the RED_VIPA resource (refer to “RED_VIPA”) ensure that the enqueue replication server is not started where the enqueue server (actually the related VIPA) is currently running (1). This is accomplished by setting the PREFerence value to 1 for the ERS and the system where the VIPA for the ES is running.

The INGGROUP commands in the startup poststart definitions of the RED_ERS resource (see the following) ensure that the enqueue replication server attracts the enqueue server if this fails (2). This is accomplished by setting the PREFerence value to 700 for the EMGRP and the system where the ERS is running.

The MAKEAVAILABLE WhenObservedSoftDown relationship against RED_EMGRP/APG/= will prevent the start of RED_ERS whenever the RED_EMGRP on the same system is in HARDDOWN status (3). This means that an SA operator has to manually change the status of the ES to AUTODOWN (after he has investigated/resolved the cause for the ES failure) in order to allow the ERS to start on that system.

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 RED_ERS definition:

1. Remove the 'MAKEAVAILABLE/WhenObservedSoftDown' relationship to RED_EMGRP/APG/=.
2. Add to the list of POSTSTART commands: SETSTATE RED_ES,AUTODOWN.

Note: One possible consequence of using SA to automatically reset the ES status instead of letting an SA Operator do it manually is that the ES may start to move back and forth if the ES fails always with the same error:

1. ES fails on LPAR1
2. SA moves it to LPAR2. There the ES fails again.
3. SA then moves the ES back to LPAR1, and so on.

This is avoided if you do not change the ERS definition. You need to decide what the best behavior is for your installation.

Definition:

```
Entry Name: RED_ERS
Link to Class USS_APPLICATION

Application Information
Application Type. . . USS
Job Name. . . . . . . REDADMER

Relationships
Relationship Type. . MAKEAVAILABLE
Supporting Resource. RED_EMGRP/APG/=
Automation . . . . . PASSIVE
Chaining . . . . . . WEAK
Condition  . . . . . WhenObservedSoftDown

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

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

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

Startup POSTSTART
INGGROUP RED_EMPLEX/APG,ACTION=RESET

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

Shutdown NORM
1
```

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

USS Control
User ID. . . . . . . REDADM
Command/Path . . . . ./ers.sapRED_ASCS00
```

Application groups

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

RED_EMGRP: This SYSTEM group combines the components of SCS.

Definition:

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

Select applications:
RED_CO
RED_ES
RED_GW
RED_MS
RED_SE
RED_VIPA

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

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

This will ensure that only one SCS and one enqueue replication server are started at a time, and that they are running on different systems, as shown in Figure 37 on page 174 (active applications are represented as shaded boxes).

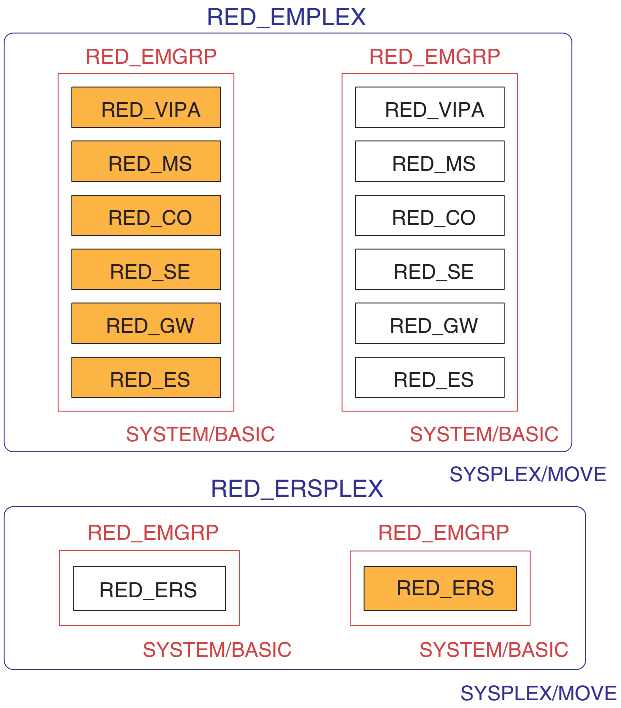


Figure 37. RED_EMPLEX and RED_ERSPLEX application groups

Definition of RED_EMPLEX:

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

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

| Definition of RED_ERSPLEX:

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

Select applications:
RED_ERS

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

RED_VPLEX: This application group is a SYSPLEX/MOVE PASSIVE group defined for the VIPA associated with SCS. Its purpose is to define a relationship between the enqueue server and its VIPA. This ensures that the *INGGROUP* command in the application automation definitions of the RED_VIPA resource (see “RED_VIPA” on page 171) is processed by SA z/OS prior to the decision where to place the enqueue replication server.

Since RED_VIPA is a MOVE group, only one of the applications in the group is started at a time, as shown in Figure 38 (active applications are represented as shaded boxes).

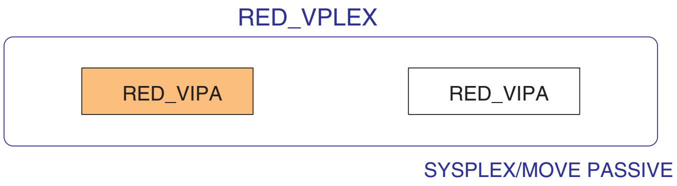


Figure 38. RED_VPLEX application group

Definition:

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

Select applications:
RED_VIPA
```

RED_COPLEX: This application group is a SYSPLEX/MOVE PASSIVE group defined for the syslog collector. Its purpose is to ensure that only one collector daemon is started or active at a time, as shown in Figure 39 on page 176 (active applications are represented as shaded boxes).

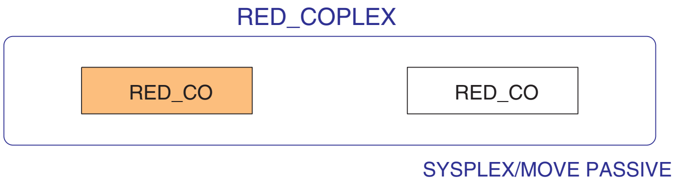


Figure 39. RED_COPLEX application group

Definition:

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

Select applications:
RED_CO
```

Application servers

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

Applications

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

APPSRV06: This application corresponds to the remote application server running on VMLINUX6.

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

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

- One SHUTNORM command, which kills only the monitor routine. When the monitor routine is gone, the remote application server appears to be down for SA 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 really does stop the application server.

Definition:

```
Entry Name: APPSRV06
Link to Class USS_APPLICATION

Application Information
Application Type. . . USS
Job Name. . . . . . . APPSRV06

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

Startup STARTUP
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startappsrv_v5 vmlinux6 00 D00 REXEC
 >/u/redadm/startappsrv.vmlinux6.00.log'

Startup POSTSTART
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/checkappsrv_v4 vmlinux6 00 >&
 /u/redadm/checkappsrv.vmlinux6.00.log'

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

Shutdown FORCE
1
INGUSS /bin/tcsh -c '/u/redadm/stopappsrv_v5 vmlinux6 00 D00 REXEC  >&
 /u/redadm/stopappsrv.vmlinux6.00.log'

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

APPSRV10: This application corresponds to the local application server running on SC42.

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

Definition:

```
Entry Name: APPSRV10
Link to Class USS_APPLICATION

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

Application Automation Definition
Job Type. . . . . . . NONMVS
Start Timeout . . . . 00:08:00
Shutdown Delay  . . . 00:05:00

Relationships
Relationship Type . . HASPARENT
```

```
Supporting Resource . OMPROUTE/APL/=

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

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

Startup STARTUP
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startappsrv_v5 wtsc42a 10 D10 >&
/u/redadm/startappsrv.wtsc42a.10.log'

Startup POSTSTART
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/checkappsrv_v4 wtsc42a 10 >&
/u/redadm/checkappsrv.wtsc42a.10.log'

Shutdown NORM
1
INGUSS /bin/tcsh -c '/u/redadm/stopappsrv_v5 wtsc42a 10 D10 >&
/u/redadm/stopappsrv.wtsc42a.10.log'
2
INGUSS /bin/kill -9 %PID%

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

APPSRV11: This application corresponds to the local application server running on SC04.

```
Entry Name: APPSRV11
Link to Class USS_APPLICATION

Application Information
Application Type. . . USS
Job Name. . . . . . . APPSRV11

Application Automation Definition
Job Type. . . . . . . NONMVS
Start Timeout . . . . 00:08:00
Shutdown Delay  . . . 00:05:00

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

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

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

Startup STARTUP
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startappsrv_v5 wtsc04a 11 D11 >&
/u/redadm/startappsrv.wtsc04a.11.log'

Startup POSTSTART
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/checkappsrv_v4 wtsc04a 11 >&
/u/redadm/checkappsrv.wtsc04a.11.log'

Shutdown NORM
1
INGUSS /bin/tcsh -c '/u/redadm/stopappsrv_v5 wtsc04a 11 D11  >&
/u/redadm/stopappsrv.wtsc04a.11.log'
2
INGUSS /bin/kill -9 %PID%
```

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

Application groups

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

RED_RASGRP: This application group is created to combine the remote application servers, although we have only one remote application server.

Definition:

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

Select applications:
APPSRV06
```

Our sample policy does not contain an explicit MAKEAVAILABLE/WhenAvailable relationship between the remote 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 by executing an R3trans command. If R3trans fails, the application server is not started. This test is part of the SAP code and means that the relationship is already ’covered’ by SAP.

   Additionally, we have enhanced our startappsrv_v5 script to test the database connection via R3trans as well. We therefore get an error message during startup in the log which the startappsrv_v5 script writes. This makes it easier to detect a problem during database connection.
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 RASGRP and this DB2 group. However, there is no check that the SAP profile definitions for sysplex failover in 4.6D or the connect.ini entries starting with 6.10 are consistent with these SA definitions. It is nearly impossible to detect and identify problems caused by such inconsistent definitions. Therefore, we do not recommend such an SA policy extension.

RED_RASPLEX: This application group is a SYSPLEX/MOVE group defined for the remote application servers. These application servers are running on remote systems like UNIX or Windows. They are monitored by SA z/OS on one only LPAR, as shown in Figure 40 on page 180 (active applications are represented as shaded boxes). If the LPAR has to be stopped, only the monitoring of the servers is moved via the MOVE group. The application servers themselves will not be stopped.

The application group RED_RASPLEX 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.

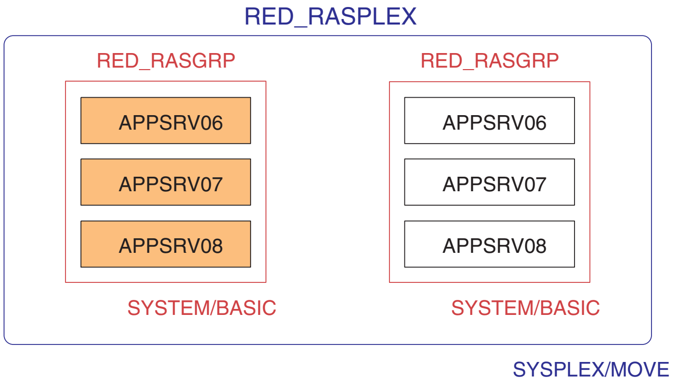


Figure 40. RED_RASPLEX application group

Definition:

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

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

Relationships
Relationship Type. . HASPARENT
Supporting Resource. NFS_HAPLEX/APG
```

RED_LASPLEX: This application group is a SYSPLEX group defined for the local application servers. One application server is running on each system, as shown on Figure 41 on page 181 (active applications are represented as shaded boxes).

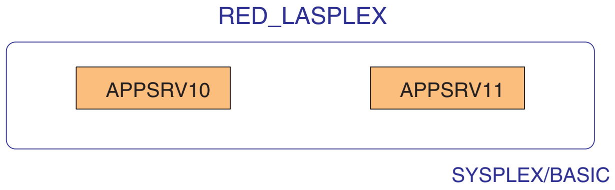


Figure 41. RED_LASPLEX application group

Definition:

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

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

SAP RED local applications

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

Applications

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

REDICLI6: This application corresponds to the ICLI server used by APPSRV06 to connect to the database server.

The following is the definition of the application REDICLI6. Because we have chosen to start the ICLI servers via a start procedure, this application is defined as a STANDARD application.

```
Entry Name:REDICLI6

Application Information
Application Type. . . STANDARD
Job Name. . . . . . . REDICLI6
```

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

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

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

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

RED_RFC: This application corresponds to rfcoscol.

Definition:

```
Entry Name: RED_RFC
Link to Class USS_APPLICATION

Application Information
Application Type. . . USS
Job Name. . . . . . . REDADMR1

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

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

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

Startup
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/start_rfcoscol >&
/u/redadm/rfcoscol.&SYSNAME..log'

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

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

Application group

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

RED_LOCAL: This SYSTEM group combines the SAP RED local applications running on a single system, as shown on Figure 42 on page 183 (active applications are represented as shaded boxes).

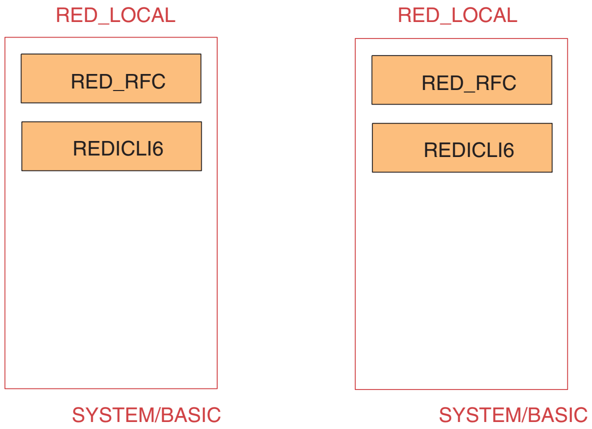


Figure 42. RED_LOCAL application group

Definition:

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

Select applications:
RED_RFC
REDICLI6

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

NFS server

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

Application

We define one application for the NFS server.

MVSNFSSA: This application corresponds to the NFS server.

Definition:

```
Entry Name:
MVSNFSSA

Application Information
Application Type. . . STANDARD
Job Name. . . . . . . MVSNFSSA

Relationships
Relationship Type . . MAKEAVAILABLE
Supporting Resource . NFS_HAPLEX/APG
```

```
Automation. . . . . . PASSIVE
Chaining. . . . . . . WEAK
Condition . . . . . . WhenObservedDown

Relationship Type . . MAKEAVAILABLE
Supporting Resource . OMPROUTE/APL/=
Automation. . . . . . ACTIVE
Chaining. . . . . . . WEAK
Condition . . . . . . WhenAvailable

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

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

Application group

We create one application group at the sysplex level.

NFS_HAPLEX: The NFS server should run on one of the two systems at a time. Therefore, we define a SYSPLEX/MOVE group with the NFS server as the only member, as shown in Figure 43 (active applications are represented as shaded boxes).

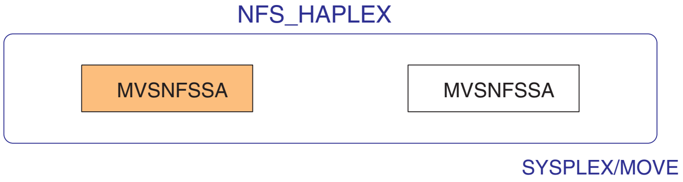


Figure 43. NFS_HAPLEX application group

Definition:

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

```
Select applications:
MVSNFSSA
  ...
```

SAProuter

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

Applications

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

SAP_RTVIPA: This application corresponds to the VIPA associated with the SAProuter.

Definition:

```
Entry Name: SAP_RTVIPA

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

Application Automation Definition
Job Type. . . . . . . TRANSIENT
Transient Rerun . . . YES

Startup
   Parameters. . . . . . ,VIPA='172.20.10.3'
```

SAP_ROUTER: This application corresponds to the SAProuter.

Definition:

```
Entry Name: SAP_ROUTER
Link to Class USS_APPLICATION

Application Information
Application Type. . . USS
Job Name. . . . . . . SAPROUTE

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

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

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

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

Application group

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

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

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

Select applications:
SAP_ROUTER
SAP_RTVIPA

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

SAP_RTPLEX: The SAProuter (and its associated VIPA) should run on one of the two systems at a time. Therefore, we define a SYSPLEX/MOVE application group, as shown in Figure 44 (active applications are represented as shaded boxes).

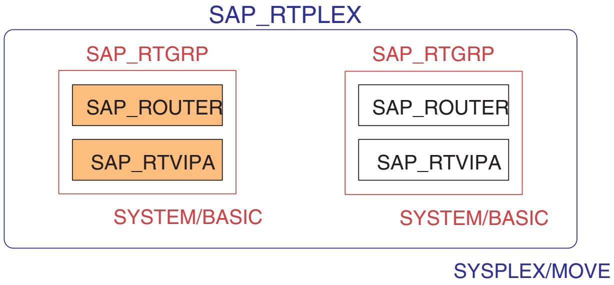


Figure 44. SAP_RTPLEX application group

Definition:

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

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

SAP local application

In this section, we describe the definition of the SAP local application saposcol. This application is started once on every system on which an SAP is running. >In this case, you have to create a SAP_OSCOL application and the application group SAP_LOCAL for each system. This is described below.

Alternatively, in a sysplex environment, it is sufficient to run saposcol once in the sysplex, because it gathers the RMF data from all the LPARs in the sysplex. For this case, configure an additional rfcoscol as described in “Additional rfcoscol” on page 145. Then put both into a basic group and add this group to a sysplex move group. See “Alternative: Sysplex-wide SAPOSCOL” on page 188. You do not need to create the application group SAP_LOCAL in this situation.

Application

We define one application for saposcol.

SAP_OSCOL: This application corresponds to saposcol.

Definition:

```
Entry Name: SAP_OSCOL
Link to Class USS_APPLICATION

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

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

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

USS Control
User ID. . . . . . . root user
Command/Path . . . . /usr/sap/RED/SYS/exe/run/saposcol
```

Application group

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

SAP_LOCAL: This group, as shown in Figure 45 on page 188, combines applications running on each LPAR. In fact, in our environment, this is just the application SAP_OSCOL (active applications are represented as shaded boxes).

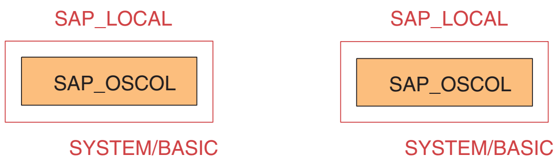


Figure 45. SAP_LOCAL application group

Definition:

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

Select applications:
SAP_OSCOL
```

Alternative: Sysplex-wide SAPOSCOL

An alternative in a sysplex environment is to create one application and application group for the sysplex.

Procedure:

First define an additional RFCOSCOL application:

```
Entry Name : SAPSYS_RFC
Link to Class USS_APPLICATION

Application Information
Application Type. . . USS
Job Name. . . . . . . SAPSYRFC

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

Startup
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/usr/sap/RED/rfc/start_rfcoscol_FCSPLEX &
/usr/sap/RED/rfc/start_rfcoscol_FCSPLEX.log'

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

```
USS Control
User ID. . . . . . . REDADM
Command/Path . . . . ./rfcoscol_FCSPLEX
```

Then define the group that contains both the saposcol and rfcoscol resources:

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

Select applications:
SAP_OSCOL and SAPSYS_RFC
```

Last, create a sysplex move group:

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

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

Defining superior groups

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

RED_SAPPLEX

This SYSPLEX application group combines all resources that belong to the SAP system RED, as shown in Figure 46 on page 190.

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

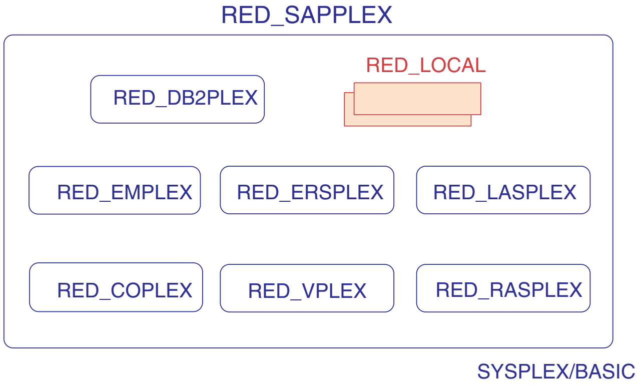


Figure 46. RED_SAPPLEX application group

Definition:

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

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

SAP

This SYSPLEX application group is the top level group of all SAP-related resources, as shown on Figure 47 on page 191.

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

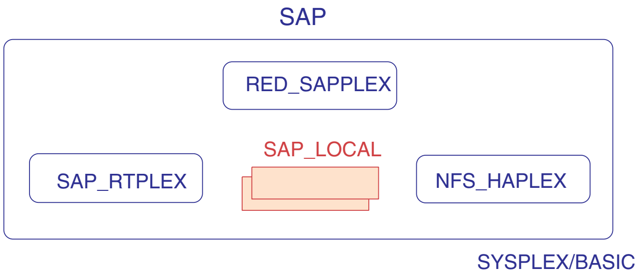


Figure 47. SAP application group

Definition:

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

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

Overall picture

Figure 48 on page 192 gives you the overall picture of all of the groups and applications that we defined in our SA z/OS policy database.

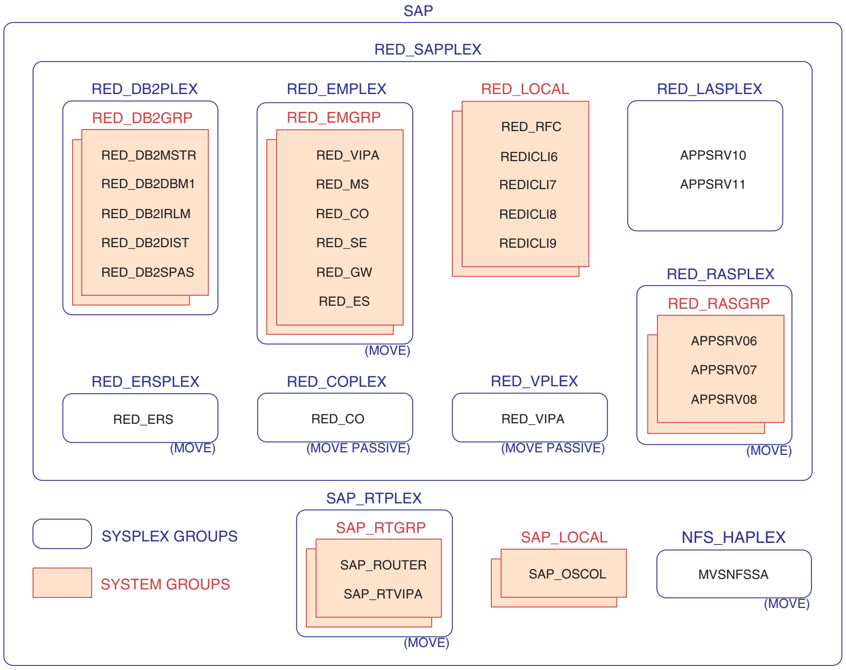


Figure 48. Overview of the resources

Summary tables

The following tables summarize all groups and applications we defined, with a short description, and the page on which you can find the parameters to be entered in the SA z/OS policy.

Classes

The following table summarizes all the classes we used.

Table 21. Summary of the classes

Name	Description	Page
CLASS_DB2_MASTER	Class for DB2 MSTR address space	165
CLASS_RED_DB2_CHILDS	Class for SAP RED DB2 children	165
USS_APPLICATION	Class for all USS applications	164

Applications

Table 22 summarizes all the applications we defined.

Table 22. Summary of the applications

Name	Description	Page
APPSRV06	RED SAP application server on VMLINUX6 (remote)	176

Table 22. Summary of the applications (continued)

Name	Description	Page
APPSRV10	RED SAP application server on SC42 (local)	177
APPSRV11	RED SAP application server on SC04 (local)	178
MVSNFSSA	Network File System server for TCPIPA	183
RED_DB2DBM1	RED DB2 DBM1 address space	166
RED_DB2DIST	RED DB2 DIST address space	167
RED_DB2IRLM	RED DB2 IRLM address space	167
RED_DB2MSTR	RED DB2 MSTER address space	166
RED_DB2SPAS	RED DB2 SPAS address space	167
RED_CO	RED SAP syslog collector	170
RED_ERS	RED SAP enqueue replication server	171
RED_ES	RED SAP enqueue server	169
RED_GW	RED SAP gateway	170
RED_MS	RED SAP message server	169
RED_RFC	RED SAP rfcoscol	182
RED_SE	RED SAP syslog sender	170
RED_VIPA	VIPA related to RED SAP SCS	171
REDICLI6	ICLI server for APPSRV6	181
SAP_OSCOL	saposcol, runs once for all SAPs on one LPAR	187
SAP_ROUTER	SAProuter	185
SAP_RTVIPA	VIPA related to saprouter	185

Application groups

The following table summarizes all application groups we defined.

Table 23. Summary of the application groups

Name	Type	Description	Page
NFS_HAPLEX	SYSPLEX MOVE	All MVSNFSSA applications	184
RED_COPLEX	SYSPLEX MOVE PASSIVE	All RED_CO applications	175
RED_DB2GRP	SYSTEM BASIC	All RED_DB2* applications	168
RED_DB2PLEX	SYSPLEX BASIC	All RED_DB2GRP application groups	168
RED_EMGRP	SYSTEM BASIC	All SCS components	173
RED_EMPLEX	SYSPLEX MOVE	All RED_EMGRP application groups	173
RED_ERSPLEX	SYSPLEX MOVE	All RED_ERS applications	173
RED_LASPLEX	SYSPLEX BASIC	All local application servers	180

Table 23. Summary of the application groups (continued)

Name	Type	Description	Page
RED_LOCAL	SYSTEM BASIC	All REDICLI* + RED_RFC applications	182
RED_RASGRP	SYSTEM BASIC	All remote application servers	179
RED_RASPLEX	SYSPLEX MOVE	All RED_RASGRP application groups	179
RED_SAPPLEX	SYSPLEX BASIC	All resources belonging to SAP RED	189
RED_VPLEX	SYSPLEX MOVE PASSIVE	All RED_VIPA applications	174
SAP	SYSPLEX BASIC	All elements of SAP	190
SAP_LOCAL	SYSTEM BASIC	All SAP_OSCOL applications	187
SAP_RTGRP	SYSTEM BASIC	All SAP_ROUTER + SAP_RTVIPA applications	185
SAP_RTPLEX	SYSPLEX MOVE	All SAP_RTGRP application groups	186

Additions to the Automation Table

The Automation Table must be enhanced to trap some special messages for the high availability solution and route them to SA z/OS. The entries for the IEF403I message trap the UP messages for the TCPVIPA and ICLI server started tasks (if you still need to run them). You need to enhance the entry for the ICLI server started task if you have more than one started task for ICLI servers defined. The entry for the BPXF024I message traps error messages during the startup of application servers. The sample file SA23_SAPMSGUX contains the listed entries, which are the only changes needed for SA V2.3.

```
************************************************************************
* DESCRIPTION: SAMPLE DSIPARM - MSG AUTOMATION TABLE FOR USS           *
************************************************************************
*
%INCLUDE AOFMSGSY
*
1*************************************************************************
*                                                                       *
* IEF403I JOB STARTED                                                   *
*                                                                       *
*************************************************************************
*
IF MSGID = 'IEF403I' & DOMAINID = '&DOMAIN.' & IFRAUSDR = %AOFSIRTASK%
   THEN BEGIN;
*
   IF TOKEN(2)='TCPVIPA' .
      THEN EXEC(CMD('ACTIVMSG UP=YES ')
           ROUTE(ONE %AOFOPGSSOPER%));
*
   IF TOKEN(2)='REDICLI6' .
      THEN EXEC(CMD('ACTIVMSG UP=YES ')
           ROUTE(ONE %AOFOPGSSOPER%));
*
   ALWAYS;
*
END;
*
*************************************************************************
*                                                                       *
* 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;
*
```

Extension for DFS/SMB

This is an extension to “Defining the SAP-related resources” on page 163. We describe here how to add the definitions for DFS/SMB to the SA z/OS policy, to SDF, and 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='SAPRED.SHFS.SAPMNT',SYSNAME=&SYSNAME.
MVS SETOMVS FILESYS,FILESYSTEM='SAPRED.SHFS.TRANS',SYSNAME=&SYSNAME.

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

Application group

We define one application group for DFS/SMB.

SMB_PLEX: DFS/SMB should run on one of the two systems at a time. Therefore, we define a SYSPLEX/MOVE group with DFS/SMB, as shown in Figure 49 (active applications are represented as shaded boxes).

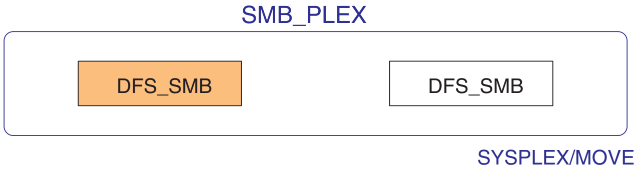


Figure 49. SMB_PLEX application group

```
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 SDF

We add the following entries for DFS_SMB to our sample SDF panel AOFSAP:

```
SF(SC04.DFS_SMB,07,40,52,N, ,)
ST(DFS_SMB   )
SF(SC42.DFS_SMB,07,54,66,N, ,)
ST(DFS_SMB   )
```

An entry DFS_SMB is also added to the SDF tree. One extra line is inserted in the members AOFTSC04 and AOFTSC42:

```
...
010700   2 SAP
010800     3 MVSNFSSA
010810     3 DFS_SMB
010900     3 SAP_ROUTER
...
```

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('TERMMSG FINAL=YES')
           ROUTE(ONE %AOFOPGSSOPER%));
*
   ALWAYS;
```

Chapter 11. Customizing the Tivoli System Automation for Multiplatforms base component

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.

For information on the SA MP end-to-end management automation component, see Chapter 12, "Customizing the Tivoli System Automation for Multiplatforms end-to-end component," on page 221.

SAP on System z is built around IBM DB2 Universal Database (UDB) for OS/390 and 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 following discussion lists the resources and components that need to be considered when implementing automation procedures in an SAP environment. The necessary steps to set up SA MP are discussed in "Setting up SA MP and SAP" on page 204. Extensive testing is required to verify a proper configuration. You can find the verification procedure in "Verification procedure and failover scenarios" on page 279. Appendix F, "Detailed description of the sample Tivoli System Automation for Multiplatforms high availability policy for SAP," on page 327 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-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 *base component* provide high availability and disaster recovery capabilities for Linux, including Linux on System z and AIX server clusters

- the *end-to-end component* 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 base component, while Chapter 12, “Customizing the Tivoli System Automation for Multiplatforms end-to-end component,” on page 221 describes tailoring of the end-to-end component.

Tivoli System Automation for Multiplatforms 2.1 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`

SAP in a high availability environment

Chapter 6, “Architecture for a highly available solution for SAP,” on page 91 and Chapter 7, “Planning and preparing a System z based end-to-end high availability solution,” on page 115 also apply to SAP running on Linux on System z. They discuss the hardware and software architecture and offer planning information, including:

- Network configuration
- File system setup
- DB2 setup
- SAP installation

Chapter 8, “Customizing SAP for high availability,” on page 129 discusses the customization of SAP needed for high availability. It describes the following components:

- ABAP SCS and J2EE SCS
- Integrated Call Level Interface (ICLI) servers (no longer needed for DB2 V8 and SAP NetWeaver '04).
- Application server instances
- SAPOSCOL
- RFCOSCOL
- SAProuter
- SAP Web Dispatcher

This gives you 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 the standalone enqueue server for high availability, which is a prerequisite for removing the single point of failure represented by SAP enqueue processing.

Scope of the sample SA MP high availability policy for SAP

Because the SAP database must reside on z/OS, parts of the SAP system will still be required to run under z/OS and must be kept highly available there. These parts are:

- DB2
- ICLI servers (no longer needed for DB2 V8 and SAP NetWeaver '04)
- RFCOSCOL
- SAPOSCOL

What can be moved to Linux on System z under the control of SA MP are the following components (SAPOSCOL needs to be on z/OS and Linux on System z):

- NFS (file systems)

- ABAP SCS and J2EE SCS
- Application server instances
- SAPOSCOL
- SAProuter
- SAP Web Dispatcher

You need the sample SAP HA policy version 4.2, as a minimum, in order to manage SAP J2EE components. Compared to the sample SAP HA policy version 3.2, the updated version 4.2:

- Supports NetWeaver '04 J2EE-only installations such as EP 6.0 and thus J2EE SCS, J2EE application servers, and the SAP Web Dispatcher
- Supports NetWeaver '04 'add-in' installations such as XI 3.0 and thus ABAP and J2EE in parallel
- Supports more than one SAP system in one SA MP domain
- Stopping of application servers reflects ’pending offline’ as long as ’stopsap’ is running
- Policy parts can be created and removed independently (CentralServices, SAPCommon, AppServer)
- Contains and uses a small J2EE Java application server health-check program
- Contains sample commands to create relationships from SAP policy resources to a high-availability database and NFS server resources
- Supports AIX

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.

With the sample SAP HA policy version 4.2, you can run an additional SAP system within the same SA MP cluster. All resources that are specific to the additional SAP system contain its unique <SAPSID> in the resource name. Additionally, we also recommend using a different prefix for this SAP system.

Figure 50 on page 202 shows an overview of the SAP policy definitions for the ABAP part of an add-in system with SAPSID EP0.

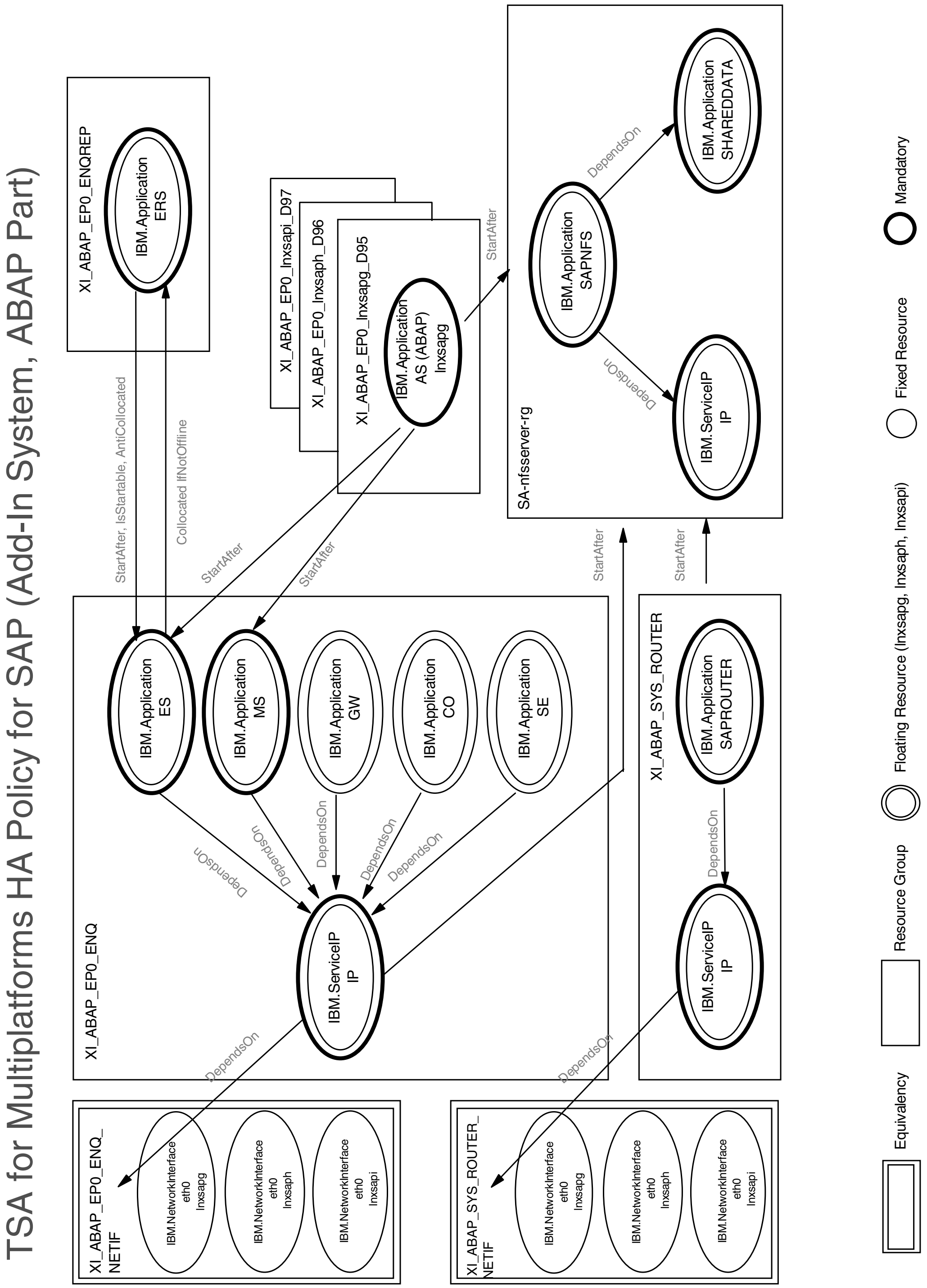


Figure 50. Overview of the SAP policy definitions for add-in system (ABAP part)

Figure 51 on page 203 shows an overview of the SAP policy definitions for the J2EE part of an add-in system with SAPSID EP0.

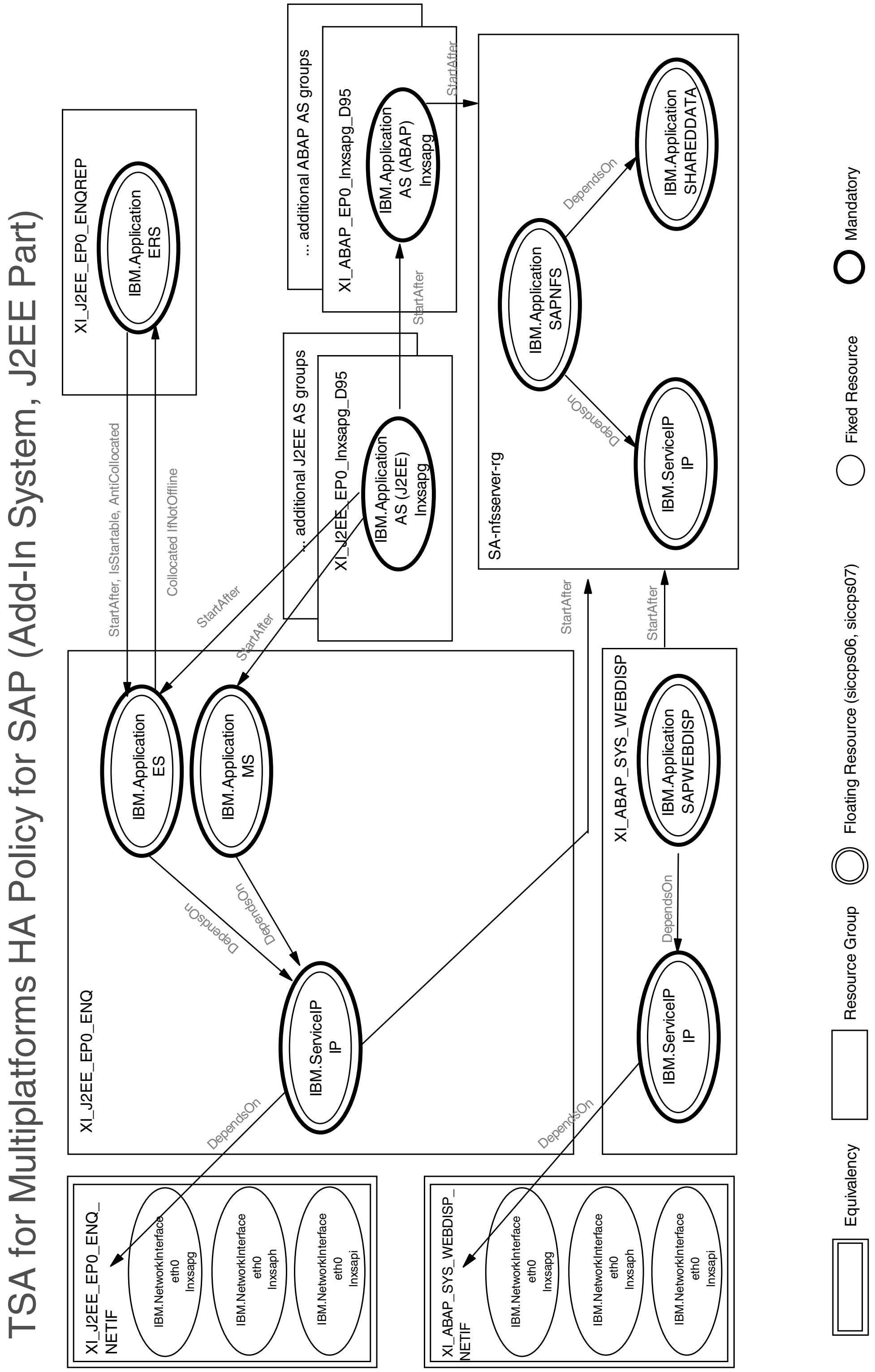


Figure 51. Overview of the SAP policy definitions for add-in system (J2EE part)

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), gateway (GW), syslog collector (CO), syslog sender (SE), and IP address (IP).
- 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 J2EE enqueue group (<J2EE_PREF>_EP0_ENQ) containing the enqueue server (ES), message server (MS), and IP address (IP).
- The J2EE enqueue replicator group (<J2EE_PREF>_EP0_ENQREP) containing the enqueue replication server (ERS)
- One or more J2EE 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 120. 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.

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.

Setting up SA MP and SAP

This section provides a detailed description of the setup of SA MP. 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.
- SAP Application Servers run on each node of the cluster.

- an SAProuter runs anywhere in the cluster.

The following figure 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 lnxsapg.

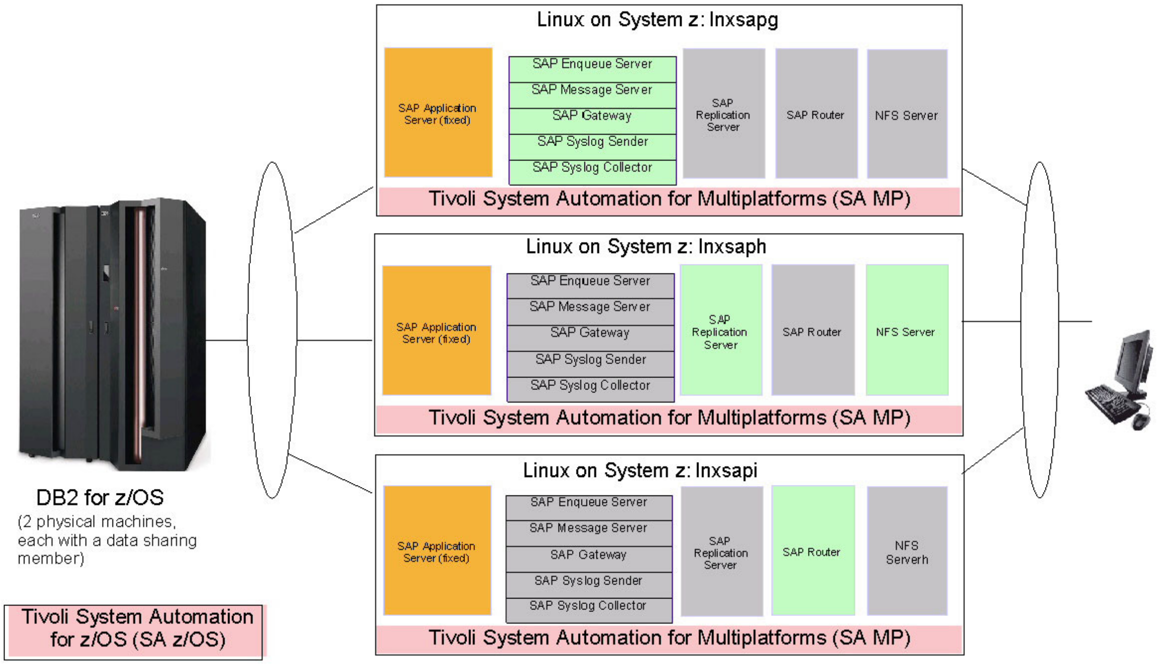

Linux on System z: lnxsapg
SAP Application Server (fixed)
SAP Enqueue Server
SAP Message Server
SAP Gateway
SAP Syslog Sender
SAP Syslog Collector
SAP Replication Server
SAP Router
NFS Server
Tivoli System Automation for Multiplatforms (SA MP)
Linux on System z: lnxsaph
SAP Application Server (fixed)
SAP Enqueue Server
SAP Message Server
SAP Gateway
SAP Syslog Sender
SAP Syslog Collector
SAP Replication Server
SAP Router
NFS Server
Tivoli System Automation for Multiplatforms (SA MP)
Linux on System z: lnxsapi
SAP Application Server (fixed)
SAP Enqueue Server
SAP Message Server
SAP Gateway
SAP Syslog Sender
SAP Syslog Collector
SAP Replication Server
SAP Router
NFS Serverh
Tivoli System Automation for Multiplatforms (SA MP)
DB2 for z/OS
(2 physical machines, each with a data sharing member)
Tivoli System Automation for z/OS (SA z/OS)

Figure 52. Recommended Linux on System z cluster

Establishing the setup

The following steps have to be performed to establish the setup:

1. Install and customize SAP
2. Install SA MP.
3. Set up SA MP to manage the SAP resources.
4. Make NFS highly available via SA MP
5. Install the SA MP high availability policy
6. Customize the SA MP high availability policy
7. Automate SAPOSCOL using SA MP
8. Verify the high availability XI installation

To uninstall, you can perform the step “Removing the HA policy” on page 219.

Installing and customizing SAP

The installation and/or customization of SAP for high availability can be taken from Part 3, “Application server considerations for high availability,” on page 89. Briefly, you need to set up an ABAP or J2EE SCS with the stand-alone enqueue and enqueue replication servers, as well as the application servers on the different nodes of the cluster. For ABAP application server monitoring, you need the SAP supplied program rfcping. Since SAP 6.30, these parts are included in the SAP kernel. If you are using an older kernel, you can obtain them from the SAP Service Marketplace as described in “Software prerequisites” on page 116. For J2EE application server monitoring, you need the GetWebPage.class utility, which is part of the preconfigured SAP policy as of version 4.2.

Extract packages to the SAP directory for executables (..SYS/exe/run). Note that the 6.20 packages are also valid for SAP 4.6 systems.

If you plan to use the SAProuter, you need to set up a routing table (saprouttab).

Installing SA MP

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 49 on page 371. 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.
3. Change to the installation directory (for example `SAM2100Base`) on the CD or in the structure unpacked from the `tar` archive.
4. Install the SA MP software by entering:
   ```
   # ./installSAM
   ```

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 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
   ```

Setting up SA MP 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, lnxsaph, and lnxsapi):
   ```
   # preprpnode lnxsapg lnxsaph lnxsapi
   ```
2. Create the SAP for SA MP cluster domain:
   ```
   # mkrpdomain sap lnxsapg lnxsaph lnxsapi
   ```
3. Start (online) the domain:
   ```
   # startrpdomain sap
   ```
4. Ensure the domain is online:
   ```
   # lsrpdomain
   ```

   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:

   ```
   # lsrpnode
   ```

   You should see output similar to

   ```
   Name     OpState  RSCTVersion
   lnxsapg  Online   2.4.3.2

   lnxsaph  Onine    2.4.3.2

   lnxsapi  Online   2.4.3.2
   ```

For this three node cluster, a reservation disk (or tie breaker disk) is not required.

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_v1.2.pdf" from

```
ftp://ftp.software.ibm.com/software/tivoli/products/sys-auto-linux
```

Note: To set up the NFS server HA policy, a domain (like `sap` in our example below) must exist. See "Setting up SA MP to manage SAP resources" on page 206 for details on how to set up a domain.

Mount the NFS-exported directory at all nodes in the SAP domain, also on 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. We recommend using the automounter.

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 3, "Application server considerations for high availability," on page 89 and Part 4, "Autonomic operation of the high availability solution for SAP," on page 159.

If you have installed and activated the SA MP NFS server HA policy, remember to perform "Step 3: Adapt the dependency file to create dependencies to the NFS server" on page 215 and "Step 4: Run the commands in the dependency file" on page 217 to ensure that the NFS server HA policy is always activated before the SAP policy.

Installing the high availability policy for SAP

The high availability policy for SAP consists of the set of scripts necessary to control the various SAP resources and utilities that simplify management of the cluster. Installation of SA MP 2.1 (or higher) installs the scripts in the

```
/usr/sbin/rsct/sapolicies/sap
```

directory. We recommend downloading the latest version. You can obtain the tar file saphasalinux-version4.2.tar (or later version) that contains it from the SA MP FTP directory:

```
ftp://ftp.software.ibm.com/software/tivoli/products/sys-auto-linux
```

Procedure:

1. Obtain the latest .tar file and copy it to /tmp, for example.
2. Switch to a directory of your choice (for example, the directory where the SAP executables reside):
   ```
   # cd <your install path>
   ```
3. Extract the files with:
   ```
    # tar xvf /tmp/saphalinux-version4.2.tar   (or later version)
   ```
4. A subdirectory
   ```
   ./ha/salinux
   ```
   is created to which the scripts are extracted.
5. 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 ha
   ```
   or
   ```
   # chown –R <sapsid>adm:sapsys /usr/sbin/rsct/sapolicies/sap
   ```
6. Do this on ***each*** cluster node. The following scripts and other files are installed:

Table 24. 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, add-in, or J2EE-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 J2EE instances
GetWebPage.classLinux	Java program to do a primitive health check of the J2EE stack on Linux systems. **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.

Table 24. Components of the SA MP high availability policy for SAP (continued)

File	Description
GetWebPage.classAIX	Java program to do a primitive health check of the J2EE stack on AIX systems. **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.
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
enqt.pf_scs01	Sample profile for SAP enqt (enqueue test) utility
InstallPolicyVnn.doc (nn = version number)	Documents the SAP sample policy installation

SAP NetWeaver '04 supports three different installation setups:

- an ABAP-only installation such as for ECC 5.0
- a J2EE-only installation such as for EP 6.0, or
- a so-called 'add-in' installation such as for XI 3.0.

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.

A ***J2EE-only installation*** consists of a J2EE CI, J2EE SCS instance, and J2EE DI application servers. For high availability the J2EE SCS instance must be installed (or manually adapted) to run with its own virtual host name.

The most complicated installation is the ***'add-in' installation***. It comprises all the components of both the ABAP-only and J2EE-only installations. For high availability, you have an ABAP SCS instance and a J2EE SCS instance, each with its own virtual host name, a J2EE CI, and J2EE and ABAP DIs.

Note: For the System z solution, the ABAP and J2EE SCS instances can also run under z/OS USS.

In order to automate J2EE-only or 'add-in' setups, the sample SA MP policy for SAP (version 4.2 or higher) is needed. This version extends the support of ABAP-only SAP systems to J2EE-only SAP systems and furthermore to the most complex SAP systems, which cover both the ABAP and J2EE stacks.

J2EE-only SAP systems are very similar to ABAP-only SAP systems from the SA MP viewpoint. This is different from SAP systems that support ABAP and J2EE stacks simultaneously, such as SAP XI. Such a so-called 'add-in' system has two

SAP Central Services instances, an ABAP SCS and a J2EE SCS in parallel. And it runs 'add-in' application servers. An 'add-in' application server is physically one instance that runs both the ABAP and the J2EE stacks. Although it is physically one instance, we separate it into its two logical parts, the ABAP application server and the J2EE application server. In other words, within an SA MP domain, one 'add-in' application server instance is automated as two logical application server instances, an ABAP application server instance and a J2EE application server instance. However, there is a close relationship between these two logical application servers. The J2EE instance is always started after the ABAP instance. This StartAfter relationship guarantees that starting the J2EE instance automatically triggers the prior start of the ABAP instance. On the other hand, stopping the J2EE application server does not stop any of the 'add-in' server processes but rather only the monitoring Java program 'java GetWebPage', which does a primitive health check of J2EE application servers.

An 'add-in' SAP system such as XI 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: J2EE resources

- J2EE SAP Central Services instance. It is named

  ```
  SCS<instance number>
  ```

  and comprises the J2EE (standalone) enqueue server, the message server, and its own virtual host name (IP address).
- Two or more J2EE 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 J2EE systems.
- SAPOSCOL, an SAP utility collecting operating system information for an application server machine. It is started by default when the first application server is started on a machine via the startsap script (outside of SA MP). However, it is not stopped automatically when all application servers are stopped. It keeps running and holds a file handle to the SAP executable directory. If the SAP executables are not local and if the NFS server that provides access to the SAP executables is automated within the SA MP domain of application servers, shutdown of the NFS server will hang up while SAPOSCOL is running.

  For example, if you have stopped all SAP resources via SA MP and you want to stop the NFS server, the open file handle from saposcol causes the stop to fail, because an implicit umount of the SAP executable directory is pending. Therefore, we strongly recommend starting and stopping saposcol directly by Tivoli System Automation. In this way, you avoid problems such as those described above. Remove the invocation of saposcol from the standard SAP start scripts, and instead leave the starting and stopping of saposcol solely to Tivoli System Automation.

The sample policy does not cover saposcol by default. Sample resource definitions and other necessary actions are described under "SAPOSCOL" on page 335.

Component D: Dependencies between component resources noted above and the NFS server

For a J2EE-only environment such as Enterprise Portal 6.0 (EP 6.0), there is no component A and no SAProuter resource.

Components A, J and I can be created by using the predefined SA MP policy. In order to do this, you need to adapt two configuration files. The first configuration file is for component A and the optional component I (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 component J 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).

Before you can adapt the configuration files, you need to set up the SAP system in such a way that it runs with an ABAP SCS instance (with standalone ABAP enqueue server) and a J2EE SCS instance (with standalone J2EE enqueue server). The procedure for doing this is described in Chapter 8, "Customizing SAP for high availability," on page 129.

Provided you have this setup, you create the SA MP policy for an 'add-in' SAP system such as XI by executing the following steps:

1. Adapt the sample ABAP and J2EE configuration files to your SAP environment.
2. Execute the 'mksap' script multiple times with appropriate command line options as described in the following.
3. Adapt the dependency file to define dependencies with respect to the NFS server
4. Execute the native SA MP commands to create the dependencies of component A, I and J to the NFS server (if running in the same domain).
5. Save the policy.
6. Verify that your HA setup works as designed. This is described in "Verifying the high availability XI (ABAP+J2EE) installation" on page 219.

These steps are described in detail in the following section.

Customizing the high availability policy for an add-in or J2EE-only SAP system

Step 1: Adapt the sample ABAP and J2EE configuration files:

Note: For a J2EE-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 J2EE resources.

The ABAP_instances.conf file contains the following entries, which you must adapt to your SAP environment.

```
lINSTALL_DIR="/home/ep0adm/ha/salinux"                 # installation directory of SAP policy scripts
CLUSTER="sap"                                          # SA MP domain name for below mentioned nodes; not used in HA scripts
NODES="lnxsapg lnxsaph lnxsapi"                        # list of nodes included in the SA MP domain
PREF="XI_ABAP"                                         # prefix of all SA MP SAP ABAP resources and SAProuter/SAP web dispatcher
# for ABAP instance we recommend using <common_prefix>_ABAP
SAPSID="EP0"                                           # SAP system ID
SAP_ADMIN_USER="ep0adm"                                # SAP administration user ID
ENQSRV_IP="9.152.81.230"                               # ASCS instance IP address
ENQSRV_HOSTNAME="XIAscsIP"                             # (virt.) hostname for ASCS IP address; if not defined, comment our of leave empty
ENQSRV_IP_NETMASK="255.255.248.0"                      # 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; for future use
ENQDIR="ASCS00"                                        # instance directory of the ASCS instance
ASNOS="95 96 97"                                       # list of instance numbers of the SAP App. servers
INSTDIRS="D95 D96 D97"                                 # list of instance directories of the SAP App. servers

# SAProuter...
SAPROUTER_IP="9.152.81.231"                            # SAProuter IP address
SAPROUTER_IP_NETMASK="255.255.248.0"                   # SAProuter IP address' netmask
SAPROUTER_IP_INTERFACE="eth0"                          # interface on which SAProuter IP address is activated on each node as alias
ROUTTAB="/usr/sap/EP0/SYS/profile/saprouttab"          # fully qualified SAProuter routing table

# in our test setup for XI we do not have an SAP web dispatcher...
#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 on which SAP web dispatcher IP address is activated on each node as alias
#SAPWEBDISP_DIR="/usr/sap/EP0/sapwebdisp"              # directory where SAP web dispatcher is installed and where its profile is located
```

Figure 53. 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. You should not need to change anything within the sapctrl_* scripts as long as you use the default SAP installation procedure. Make sure that you have the SAP supplied rfcping program included in the directory containing the SAP executables. Also make sure that you have a routing table (saprouttab) for the SAProuter accessible by all nodes in the cluster. If you want to automate the SAP Web Dispatcher, make sure it is installed and configured correctly. You can adapt the scripts within the indicated areas. Change only the lines within the indicated area. Do not add or remove lines, and do not change the names to the left of the equal signs.

The number of entries in NODES, ASNOS, and INSTDIRS must be the same, because the sample policy assumes the following:

- the ABAP application server with the first instance number (95) and first instance directory (D95) runs on node 1 (lnxsapg)
- the ABAP application server with the second instance number (96) and second instance directory (D96) runs on node 2 (lnxsaph)

and so on.

Ensure that the IP addresses you want to associate with central services and the SAProuter are currently unused and available. We use the IP addresses 9.152.81.230 and 9.152.81.231 (netmask 255.255.248.0) in the examples that follow. In addition, make sure that the SAP profiles are set up to use these addresses. You might want to register the IP addresses in the Domain Name System (DNS) and use the names instead of the addresses. Note that the IP address must be given, whereas the DNS name is only used to identify the ASCS profile.

Using network equivalencies for each IP address removes the limitation that only IP addresses of the same IP subnet as the network interface can be used. If you are using an IP address of a different subnet, you must define the alias interface (such as eth0:1) as a separate interface in the OSPF configuration file. See ″What is the IBM.ServiceIP resource class?″ in the respective *Guide and Reference* (Table 49 on page 371) for details.

The above entries reflect an SAP environment in which no SAP Web Dispatcher is running. If you run an SAP Web Dispatcher in your environment, remove the comment indicator (#) at the beginning of the lines, in order to be able to create SA MP resources for the SAP Web Dispatcher. Comment out the lines for SAProuter if you do not run an SAProuter in your environment.

The J2EE_instances.conf file contains the following entries, which you must adapt to your SAP environment:

```
INSTALL_DIR="/home/ep0adm/ha/salinux"                 # installation directory of SAP policy scripts
CLUSTER="sap"                                         # SA MP domain name for below mentioned nodes; not used in HA scripts
NODES="lnxsapg lnxsaph lnxsapi"                       # list of nodes included in the SA MP domain
PREF="XI_J2EE"                                        # prefix of all SA MP SAP J2EE resources
# for J2EE instance we recommend to use <common prefix>_J2EE
ABAP_PREF="XI_ABAP"                                   # prefix of all SA MP SAP ABAP resources; it is only needed, if the J2EE
                                                      # application server is part of an 'Add-In' application server (like for XI);
                                                      # it must have the same value as PREF in ABAP_instances.conf
SAPSID="EP0"                                          # SAP system ID
SAP_ADMIN_USER="ep0adm"                               # SAP administration user ID
ENQSRV_IP="9.152.81.232"                              # SCS instance IP address
ENQSRV_HOSTNAME="XIScsIP"                             # (virt.) hostname for SCS IP address; if not defined, comment our of leave empty
ENQSRV_IP_NETMASK="255.255.248.0"                     # 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; for future use
ENQDIR="SCS01"                                        # instance directory of the SCS instance
ASNOS="95 96 97"                                      # list of instance numbers of the SAP J2EE App. servers
INSTDIRS="D95 D96 D97"                                # list of instance directories of the J2EE SAP App. servers

# setting J2EE_HTTP_PORTS indicates, that profile is for J2EE
J2EE_HTTP_PORTS="59500 59600 59700"                         # list of J2EE instance HTTP ports (default: 50000+instance number*100 +0)
```

Figure 54. Entries to be adapted in the J2EE_instances.conf file

See Table 43 on page 341 for an explanation of the differences from the ABAP profile.

The number of entries in NODES, ASNOS, and INSTDIRS must be the same, because the sample policy assumes the following:

- the J2EE application server with the first instance number (95) and first instance directory (D95) runs on node 1 (lnxsapg)
- the J2EE application server with the second instance number (96) and second instance directory (D96) runs on node 2 (lnxsaph)

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 component A (ABAP resources) and optional component I (SAP-system-independent resources)

The next step is to run the mksap script to create the SA MP resource groups, resources, relationships and equivalencies for components A (ABAP resources) and I (SAPSID-independent resources). 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]]
mksap creates (parts of) the sample SA MP policy for a highly available SAP environment.
  -h,  Print this help.
  -s,  Use static equivalencies for sample SA MP 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 SA MP policy entries for an SAP application server on NodeName and add them to an existing SA MP policy.
  -t,  Define the type of the sample SA MP policy to be created; valid types: CentralServices|SAPCommon|AppServer;  default: None, create all.
  -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 SA MP policy entries for all nodes specified; default: create entries for all nodes defined in the configuration file.
```

Figure 55. 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.
2. Run (with adapted node names):

   ```
   ./mksap -c ABAP_instances.conf -t AppServer -i ABAP -n lnxsapg,lnxsaph,lnxsapi
   ```

   This step creates the SA MP resources, etc. for an ABAP application server on nodes lnxsapg, lnxsaph and lnxsapi.

   Note: Option -n <node list> is not necessary if an application server resource is to be created for each node listed in the configuration file entry 'NODES'.
3. (Optional) Run the mksap script to create the SA MP resources, etc. for component I (SAP-system-independent resources). Perform this step also for a J2EE-only system such as EP 6.0 if you want to run an SAP Web Dispatcher under SA 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 component J (J2EE resources)

1. As the root user, run the mksap script again to create the SA MP resources, etc. for component J (J2EE resources). In this case, use the configuration file J2EE_instances.conf. If your add-in installation is pure Unicode, first run:

   ```
   ./mksap -c J2EE_instances.conf -t CentralServices -i ADDIN
   ```

   This creates the SA MP resources, etc. for the J2EE SCS.

Before you proceed to create the ‘add-in’ J2EE application server resources, make sure that the corresponding ABAP application server resources have already been created.

2. Run:

```
./mksap -c J2EE_instances.conf -t AppServer -i ADDIN
```

This creates the SA MP resources, etc. for a J2EE application server on each node defined in the configuration file entry ‘NODES’, in our case on nodes lnxsapg, lnxsaph and lnxsapi. It also generates the StartAfter relationships between the J2EE 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.

The foregoing applies to a pure Unicode installation. It is also possible to install the ABAP part to use the ASCII character set and the J2EE part to use Unicode. The SAP ABAP executables are then located in

```
./usr/sap/<sapsid>/SYS/exe/run
```

and the SAP J2EE executables in

```
./usr/sap/<sapsid>/SYS/exe/runU
```

If your add-in setup configuration is such a mixed setup, you must use the ’-l ASCII’ option in order to create the J2EE resources and make them use the correct exe/runU directory:

```
./mksap -c J2EE_instances.conf -t CentralServices -i ADDIN -l ASCII
./mksap -c J2EE_instances.conf -t AppServer -i ADDIN -l ASCII
```

Note: For a J2EE-only system such as Enterprise Portal 6.0 (EP 6.0) you only need to run:

```
./mksap -c J2EE_instances.conf -t CentralServices -i J2EE
./mksap -c J2EE_instances.conf -t AppServer  -i J2EE
```

Step 3: Adapt the dependency file to create dependencies to the NFS server:
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 resource names. You can list all SA MP resources with the command

```
/usr/sbin/rsct/sapolicies/bin/getstatus
```

Alternatively, you can use `lssap` to display all SAP resources. In our XI environment, all SAP resources names start with 'XI' because our ‘PREF’ entry in both configuration files starts with “XI”. Therefore we get a list of all resources which begin with ″XI″ by calling:

```
        ./lssap -p XI
```

This produces an output like:

```
.---------------------------------.
| SAP resources 09/12/05 14:19:59 |
'---------------------------------'
 XI_ABAP_EP0_ENQ Offline (Offline)
 '- XI_ABAP_EP0_ENQ_ES Offline
 '- XI_ABAP_EP0_ENQ_MS Offline
 '- XI_ABAP_EP0_ENQ_GW Offline
 '- XI_ABAP_EP0_ENQ_CO Offline
 '- XI_ABAP_EP0_ENQ_SE Offline
 '- XI_ABAP_EP0_ENQ_IP Offline
 XI_ABAP_EP0_ENQREP Offline (Offline)
 '- XI_ABAP_EP0_ENQREP_ERS Offline
 XI_ABAP_EP0_lnxsapg_D95 Offline (Offline)
 '- XI_ABAP_EP0_lnxsapg_D95_AS Offline
 XI_ABAP_EP0_lnxsaph_D96 Offline (Offline)
 '- XI_ABAP_EP0_lnxsaph_D96_AS Offline
 XI_ABAP_EP0_lnxsapi_D97 Offline (Offline)
 '- XI_ABAP_EP0_lnxsapi_D97_AS Offline
 XI_J2EE_EP0_ENQ Offline (Offline)
 '- XI_J2EE_EP0_ENQ_ES Offline
 '- XI_J2EE_EP0_ENQ_MS Offline
 '- XI_J2EE_EP0_ENQ_IP Offline
 XI_J2EE_EP0_ENQREP Offline (Offline)
 '- XI_J2EE_EP0_ENQREP_ERS Offline
 XI_J2EE_EP0_lnxsapg_D95 Offline (Offline)
 '- XI_J2EE_EP0_lnxsapg_D95_AS Offline
 XI_J2EE_EP0_lnxsaph_D96 Offline (Offline)
 '- XI_J2EE_EP0_lnxsaph_D96_AS Offline
 XI_J2EE_EP0_lnxsapi_D97 Offline (Offline)
 '- XI_J2EE_EP0_lnxsapi_D97_AS Offline
 XI_ABAP_SYS_ROUTER Offline (Offline)
 '- XI_ABAP_SYS_ROUTER_SAPROUTER Offline
 '- XI_ABAP_ROUTER_IP Offline
```

Both the above output and that of the

```
/usr/sbin/rsct/sapolicies/bin/getstatus
```

command can be used to adapt the resource names in both sample files to your environment. Also, you can find a description of the naming conventions for SAP resources in “Tivoli System Automation for Multiplatforms” on page 120.

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

In detail, the file contains the following:

```
# All NFS and SAP resources must be offline before running this script!
# Version 4.2
# Step 1:  Replace static NFS Server equivalency with a dynamic one.
#          Adapt names and interface(s) to your environment!
rmequ  SA-nfsserver-nieq-1
mkequ -D 'Name like "eth0"' SA-nfsserver-nieq-1 IBM.NetworkInterface
rmrel SA-nfsserver-ip-on-nieq-1
mkrel -S IBM.ServiceIP:SA-nfsserver-ip-1 -p DependsOn -G IBM.Equivalency:SA-nfsserver-nieq-1 SA-nfsserver-ip-on-nieq-1

# Step 2:  Add relationships to start SAP after NFS server
#          Adapt names to your environment!
#          If neither NFS nor SAP Central Services (ABAP or J2EE) are running within the SA MP domain, then you do not need them. Comment out.
mkrel -S IBM.ServiceIP:XI_ABAP_EP0_ENQ_IP -p StartAfter -G IBM.ResourceGroup:SA-nfsserver-rg XI_ABAP_EP0_ENQ_IP_StartAfter_SA-nfsserver-server
mkrel -S IBM.ServiceIP:XI_J2EE_EP0_ENQ_IP -p StartAfter -G IBM.ResourceGroup:SA-nfsserver-rg XI_J2EE_EP0_ENQ_IP_StartAfter_SA-nfsserver-server

# Step 3:  Add relationship to start SAProuter/SAP Web Dispatcher after NFS Server.
#          Adapt names to your environment!
#          If neither NFS nor SAProuter/SAP Web Dispatcher are running within the SA MP domain, then you do not need this. Comment out.
#mkrel -S IBM.ServiceIP:XI_ABAP_SYS_ROUTER_IP -p StartAfter -G IBM.ResourceGroup:SA-nfsserver-rg XI_ABAP_SYS_ROUTER_IP_StartAfter_SA-nfsserver-server
#mkrel -S IBM.ServiceIP:XI_ABAP_SYS_WEBDISP_IP -p StartAfter -G IBM.ResourceGroup:SA-nfsserver-rg XI_ABAP_SYS_WEBDISP_IP_StartAfter_SA-nfsserver-server

# Step 4:  Add StartAfter relationship of Application server to the NFS server.
mkrel -S IBM.Application:XI_ABAP_EP0_lnxsapg_D95_AS:lnxsapg -p StartAfter -G IBM.ResourceGroup:SA-nfsserver-rg XI_ABAP_EP0_lnxsapg_D95_AS_StartAfter_NFS
mkrel -S IBM.Application:XI_ABAP_EP0_lnxsaph_D96_AS:lnxsaph -p StartAfter -G IBM.ResourceGroup:SA-nfsserver-rg XI_ABAP_EP0_lnxsaph_D96_AS_StartAfter_NFS
mkrel -S IBM.Application:XI_ABAP_EP0_lnxsapi_D97_AS:lnxsapi -p StartAfter -G IBM.ResourceGroup:SA-nfsserver-rg XI_ABAP_EP0_lnxsapi_D97_AS_StartAfter_NFS
```

Figure 56. sap_StartAfterRel_nfs_etc. script

Step 4: Run the commands in the dependency file:
Before you can run the commands within the file, you must stop all resources referenced therein. Then execute the commands with

```
./sap_StartAfterRel_nfs_etc
```

The resource groups and resources must be offline, because relationships can only defined for offline resources and resource groups. Verify via the

```
lsrel –Ab
```

command that all relationships defined in the file 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.

Step 5: Save the policy:
Run

```
sampolicy –s <policy-file>.xml
```

to save the policy, where `<policy-file>` is a name you assign..

Automating SAPOSCOL via SA MP

See “SAPOSCOL” on page 335 for a detailed discussion of this topic.

Starting the SAP system

You can now start your entire SAP system by issuing the commands:

```
# chrg -o online -s "Name like 'XI%'"
# chrg -o online -s "Name like 'SAP%'"
```

After a short delay, you should see the following output when calling:

```
# ./lssap -p % .
```

```
.---------------------------------.
| SAP resources 03/08/06 16:22:19 |
'---------------------------------'
 SAP_SYS_SAPOSCOL_lnxsapg Online (Online)
 '- SAP_SYS_SAPOSCOL_EXE_lnxsapg Online on {lnxsapg.boeblingen.de.ibm.com}
 XI_J2EE_EP0_lnxsapg_D95 Online (Online)
 '- XI_J2EE_EP0_lnxsapg_D95_AS Online on {lnxsapg.boeblingen.de.ibm.com}
 XI_J2EE_EP0_lnxsaph_D96 Online (Online)
 '- XI_J2EE_EP0_lnxsaph_D96_AS Online on {lnxsaph.boeblingen.de.ibm.com}
 XI_J2EE_EP0_lnxsapi_D97 Online (Online)
 '- XI_J2EE_EP0_lnxsapi_D97_AS Online on {lnxsapi.boeblingen.de.ibm.com}
 SAP_SYS_SAPOSCOL_lnxsaph Online (Online)
 '- SAP_SYS_SAPOSCOL_EXE_lnxsaph Online on {lnxsaph.boeblingen.de.ibm.com}
 SAP_SYS_SAPOSCOL_lnxsapi Online (Online)
 '- SAP_SYS_SAPOSCOL_EXE_lnxsapi Online on {lnxsapi.boeblingen.de.ibm.com}
 XI_J2EE_EP0_ENQREP Online (Online)
 '- XI_J2EE_EP0_ENQREP_ERS Online on {lnxsaph.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_lnxsaph_D96 Online (Online)
 '- XI_ABAP_EP0_lnxsaph_D96_AS Online on {lnxsaph.boeblingen.de.ibm.com}
 XI_ABAP_EP0_lnxsapi_D97 Online (Online)
 '- XI_ABAP_EP0_lnxsapi_D97_AS Online on {lnxsapi.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-nfsserver-rg Online (Online)
 '- SA-nfsserver-data-work Online on {lnxsapg.boeblingen.de.ibm.com}
 '- SA-nfsserver-data-varlibnfs Online on {lnxsapg.boeblingen.de.ibm.com}
 '- SA-nfsserver-ip-1 Online on {lnxsapg.boeblingen.de.ibm.com}
 '- SA-nfsserver-server Online on {lnxsapg.boeblingen.de.ibm.com}
```

Figure 57. 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 2.1 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 the high availability XI (ABAP+J2EE) installation

To verify your HA setup, perform the steps listed in Chapter 14, "Verification and problem determination on Linux on System z," on page 279. 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 J2EE 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
```

If you have created SA MP groups, resources and relationships for SAPOSCOL starting with "SAP", you can do the following to delete them:

1. `# cp ABAP_instances.conf removeSAPresources.conf`
2. `# vi removeSAPresources.conf` and change the PREF value to "SAP"
3. `# ./rmsap -c removeSAPresources.conf` and enter 'no' and then 'yes'

Two-node scenario using 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 49 on page 371)

Chapter 12. Customizing the Tivoli System Automation for Multiplatforms end-to-end component

This chapter describes the strategic implementation and design of the end-to-end automation solution for the heterogeneous SAP on System z environment. The strategic tool to automate a heterogeneous solution environment is the IBM Tivoli System Automation for Multiplatforms end-to-end management automation component. We provide guidance and recommendations based on the sample high availability SAP on System z environments already described in Chapter 10, “Customizing Tivoli System Automation for z/OS,” on page 161 and Chapter 11, “Customizing the Tivoli System Automation for Multiplatforms base component,” on page 199.

SAP on System z is built around IBM DB2 Universal Database (UDB) 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, such as the 64-bit operating systems Linux for System z, Linux for System x, and AIX. The environment is therefore heterogeneous by nature. The SA MP end-to-end component is ideal for automating such an environment.

Overview of end-to-end automation management

End-to-end automation can be used to automate the operation of resources within heterogeneous environments (referred to as 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 domain is connected to the end-to-end automation manager by an end-to-end automation adapter.

The main elements of the end-to-end component are the end-to-end automation manager and the operations console. The operations 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

For details about these elements, see the SA MP publications in Table 49 on page 371.

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

As previously noted, the SAP on System z solution runs only the database under z/OS but application servers on non-z/OS systems as follows:

- 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 an overview of the sample SAP on System z HA solution:

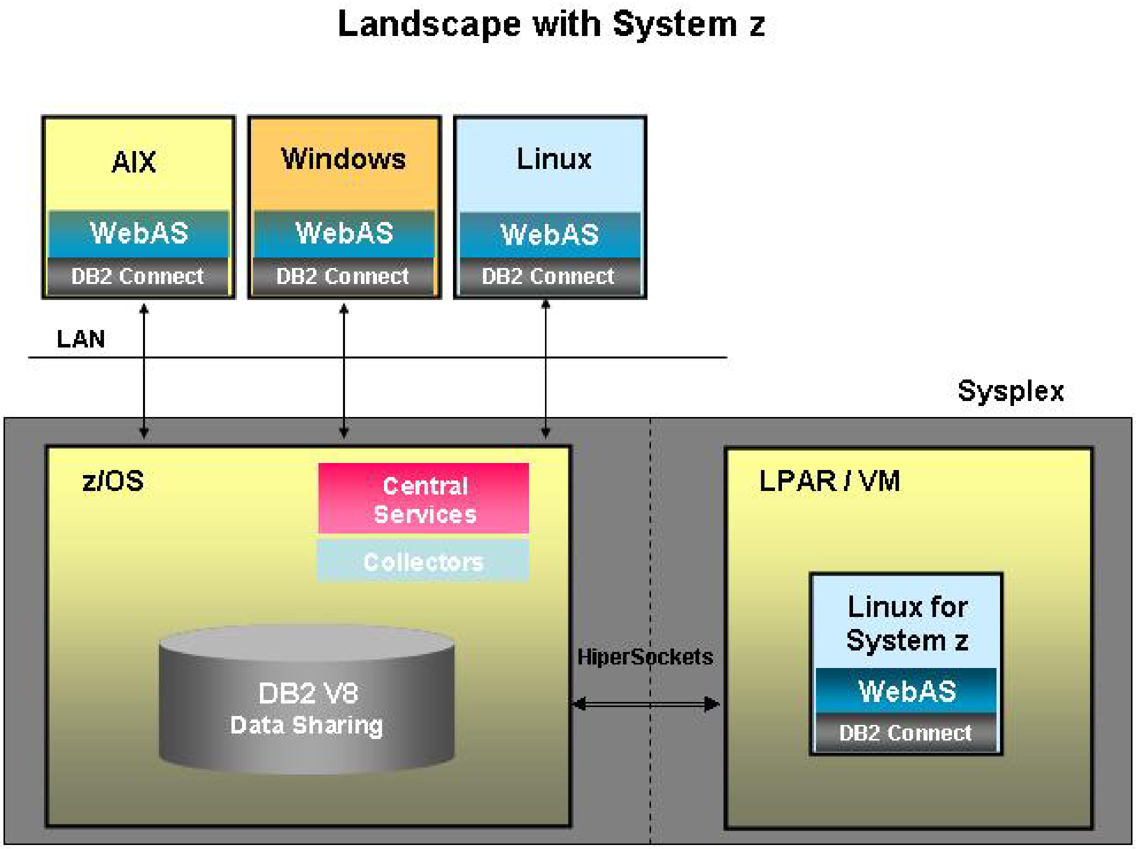


Figure 58. Landscape with sample SAP on System z HA solution

This solution is by design heterogeneous. SA MP's end-to-end component is ideally suited for managing such a heterogeneous environment. It allows easy and reliable starting and stopping of a heterogeneous SAP system. Additionally, it gives you instant problem detection across cluster boundaries. And finally, it 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:

- DB2 database with at least two data sharing members
- SAP Central Services
- necessary infrastructure such as an NFS server and SAPOSCOL for the z/OS cluster.

The application servers and other critical utilities that more appropriately reside in application server nodes run on ‘remote’ systems. The remote nodes are managed by SA MP. The utilities are the SAProuter, SAP Web Dispatcher, and SAPOSCOL, for example.

Setting up the end-to-end component

To install the end-to-end component, see the appropriate SA MP publication in Table 49 on page 371.

The sample end-to-end policy presented in the following is based on the following SAP system setup, which we have recommended for our SAP on System z HA solution:

- A three LPAR z/OS cluster, running on two physical machines, is implemented on System z hardware. SA z/OS manages:
  - 3 DB2 data sharing members, one in each LPAR, which do *not* move
  - SAP ABAP 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:
  - 3 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)
  - 3 SAPOSCOLs, one on each node, which do *not* move

The SAP system ID is EP0, and the <sapsid>adm user ID is **ep0adm**, with home directory `/home/ep0adm`.

The figures that follow reflect the z/OS cluster and the SA z/OS resource groups and resources for the previously described sample high availability environment of the SAP on System z solution. Understanding this structure is necessary in order to understand the references to the SA z/OS first-level domain that are contained in the end-to-end sample policy.

The first figure shows the top level SAP_X group, which contains a group for SAP system (SAPSID)-specific resources (SAPEP0_X) and one for all SAP-system-independent resources (SAPSYS_X):

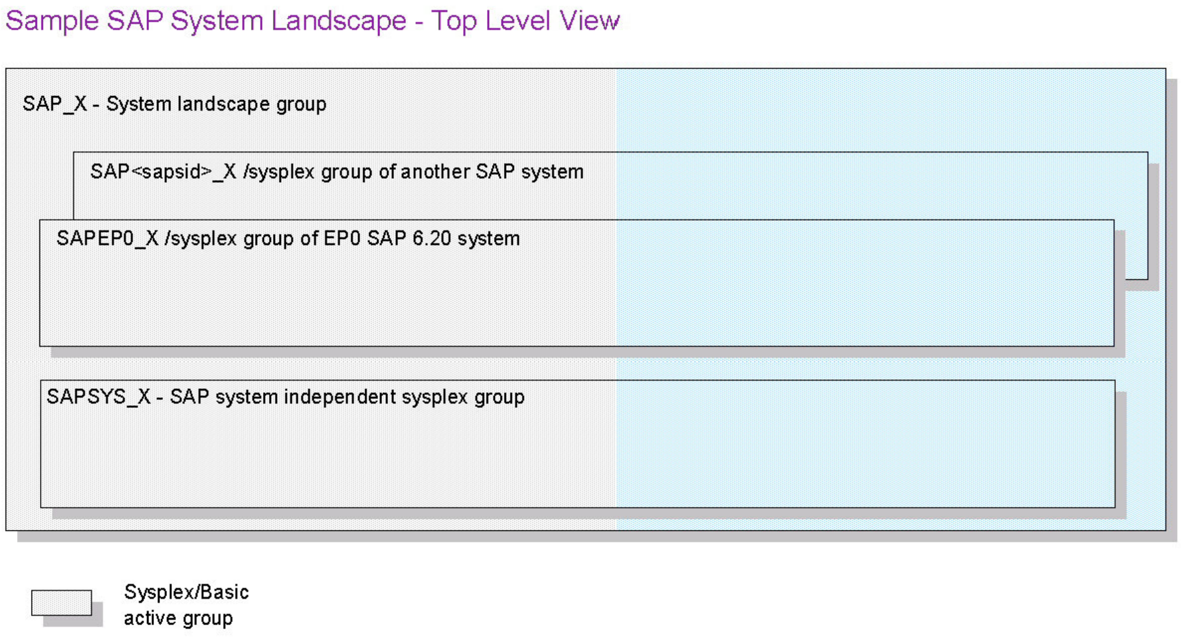

Sample SAP System Landscape - Top Level View
SAP_X - System landscape group
SAP<sapsid>_X /sysplex group of another SAP system
SAPEP0_X /sysplex group of EP0 SAP 6.20 system
SAPSYS_X - SAP system independent sysplex group
Sysplex/Basic
active group

Figure 59. z/OS cluster and SA z/OS resources (overview)

The next figure shows the details of the SAPSID-specific resources (SAPEP0_X group). The SAPSID-specific group comprises the groups for the DB2 database (SAPEP0_DBX), the ASCS (referred to here as SAPEP0_ENQX) and necessary utilities SAPEP0_UTIL. The utilities are the ICLI server resources, which are needed for DB2 prior to Version 8, and local z/OS application server resources. The group for the DB2 database (SAPEP0_DBX) in turn contains the DB2 resources: MSTR, DBM1, IRLM, etc. address spaces.

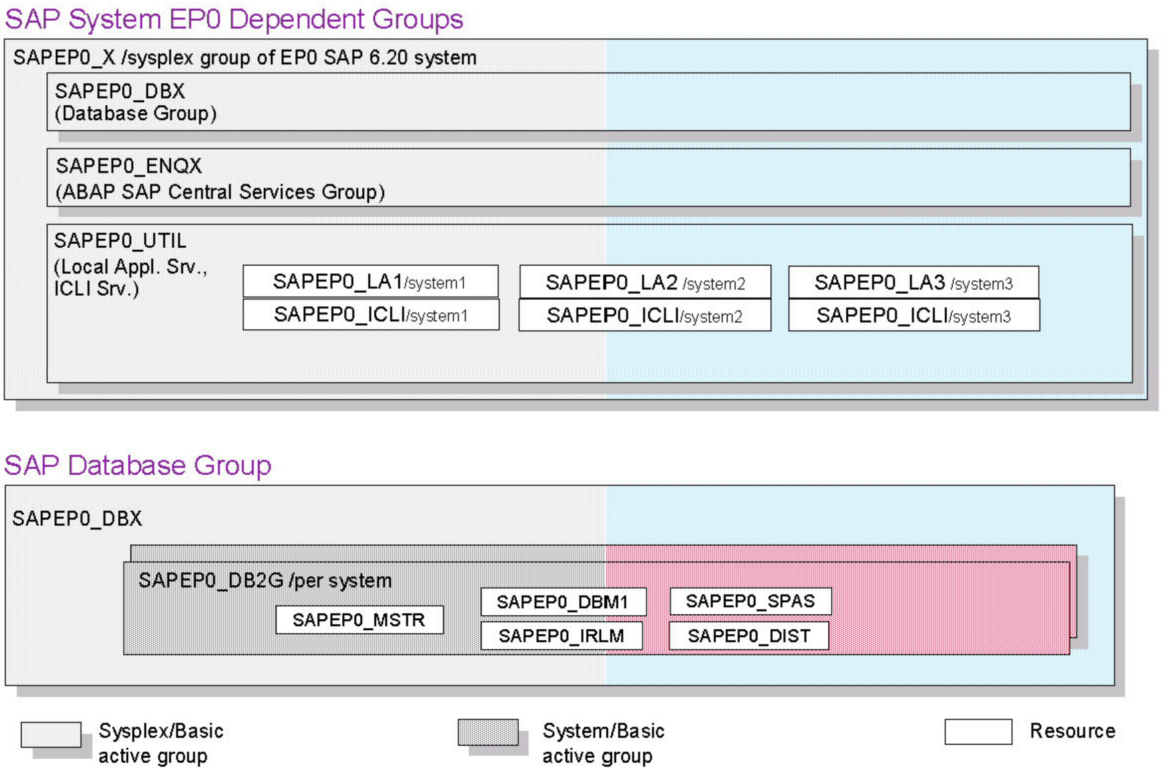


Figure 60. SA z/OS resources (Part 1)

The next figure shows details for the ASCS resources group SAPEP0_ENQX and the SAP-system-independent resources group SAPSYS_X. For details of the RED_EMPLEX and RED_ERSPLEX groups, see Chapter 10, "Customizing Tivoli System Automation for z/OS," on page 161 and in particular "SAP Central Services and the enqueue replication server" on page 169.

The SAP-system-independent resources group SAPSYS_X comprises the highly available NFS server, SAPOSCOL (collecting sysplex-wide data) and SAProuter resources. Here we run the SAProuter under USS in order to be 'close to' the local z/OS application servers.

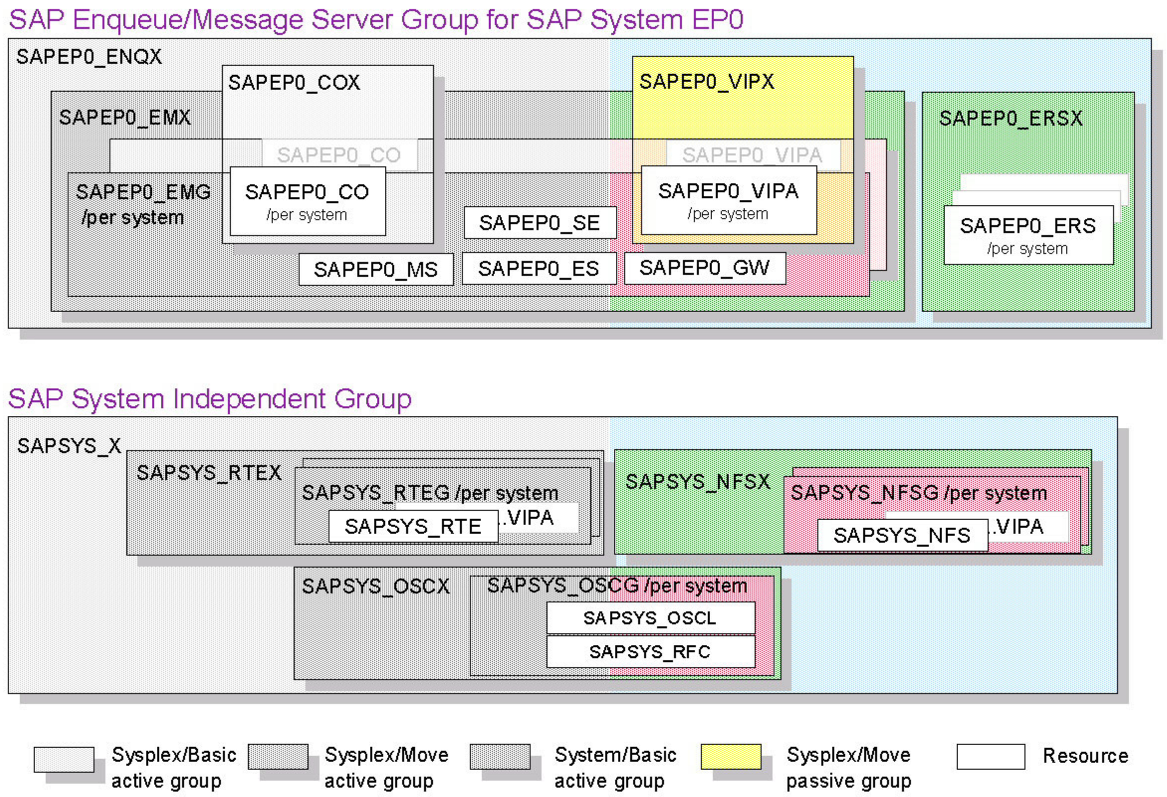


Figure 61. SA z/OS resources (Part 2)

The next figure shows the Linux on System z cluster and the SA MP resource groups and resources for the previously described sample high availability environment of the SAP on System z solution. Understanding this structure is necessary in order to understand the references to the SA MP first-level domain that are contained in the end-to-end sample policy. The policy comprises:

- 3 fixed application servers, one on each node
- 1 SAProuter utility, movable between the nodes
- 1 end-to-end adapter, movable between the nodes
- 3 fixed SAPOSCOLs, one on each node

A light background indicates a floating resource that is currently active. For example, SAProuter is active on node `lnxsapi`.

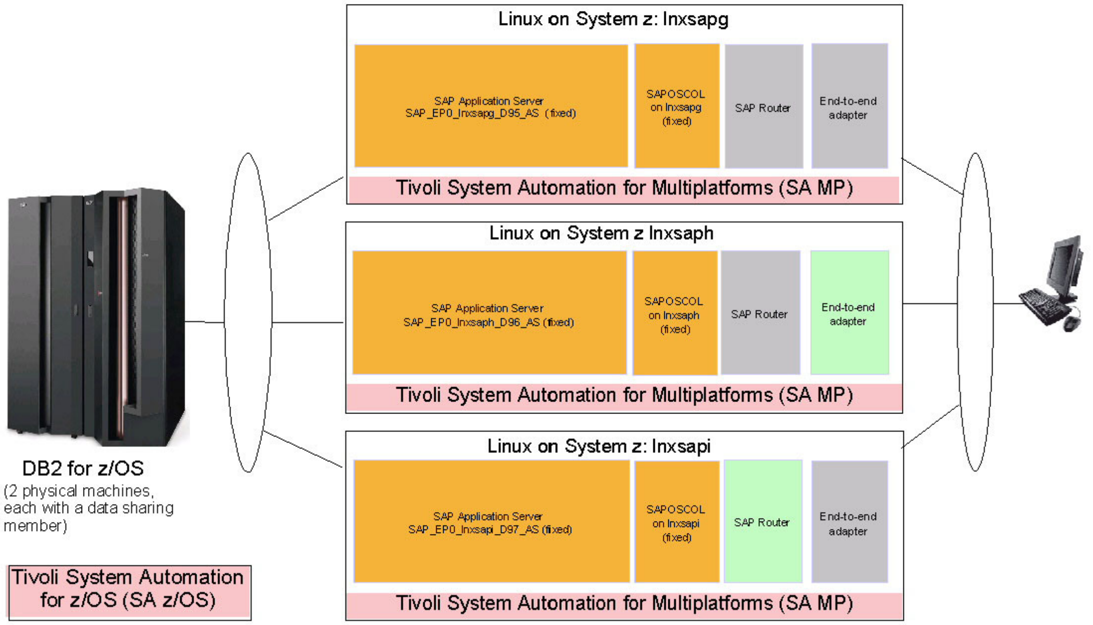


Figure 62. Linux on System z cluster

Based on the first-level domains for SA z/OS and SA MP, the sample end-to-end policy contains resource references:

- “SAPEP0 AppServer D95 on lnxsapg” for application server group “SAP_EP0_lnxsapg_D95” and similarly for each additional application server group
- “SAPSYS SAPRouter Grp” to the SAProuter group “SAP_SYS_ROUTER” running under Linux on System z

- “SAPEP0 ABAP SCS SPGrp” to the ABAP Central Services group “SAPEP0_ENQX” and a resource reference “SAPEP0 ABAP ERS SPGrp” to the corresponding enqueue replication server group ″SAPEP0_ERSX” running under USS
- “SAPSYS SAPOSCOL SPGrp” to the sysplex-wide saposcol group “SAPSYS_OSCX” and a resource reference “SAPEP0 UTIL Grp” to the Utilities group “SAPEP0_UTIL”.
- "SAPSYS SAPOSCOL on lnxsapg″ to the saposcol resource group ″SAP_SYS_SAPOSCOL_LNXSAPG″ running on machine lnxsapg. There are similar resource references for each SAPOSCOL resource running on each other application server.
- “SAPEP0 DB SPGrp” to the DB2 group “SAPEP0_DBX” and a resource reference “SAPSYS NFS SPGrp” to the NFS server group “SAPSYS_NFSX” running under native z/OS.

The application server references are associated in an end-to-end group “SAPEP0 AppServer Grp”, which has StartAfter relations to the ABAP Central Services “SAPEP0 ABAP SCS SPGrp” and NFS Server “SAPSYS_NFSX”. Together with all other SAP EP0-system-specific resource references, this group forms the end-to-end group “SAPEP0 All Resources Grp”. All other resource references that are independent of SAP system EP0 form the end-to-end group “SAP Infrastructure Grp”. This grouping allows you to add another SAP<sapsid> system by adding another “SAP<sapsid> All Resources Grp” and being able to reuse the infrastructure group. Optionally, you can put these high level groups into the end-to-end group “SAP Domain”.

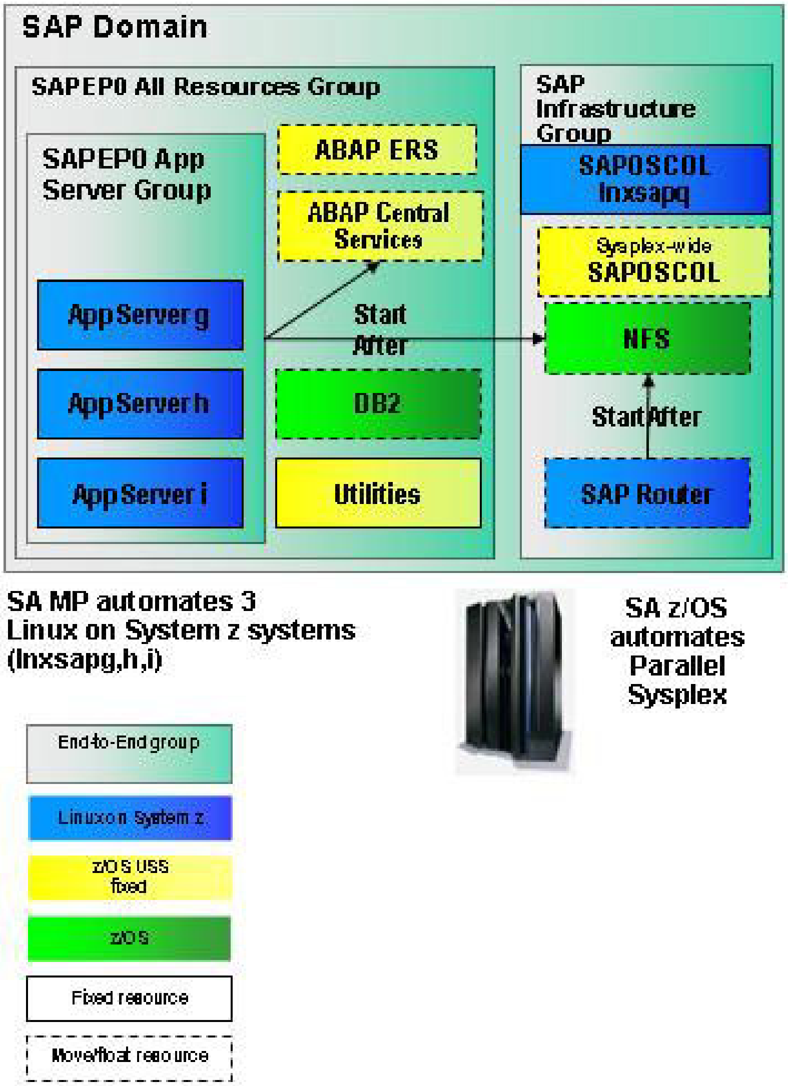


Figure 63. Overview of the end-to-end policy definitions

Defining and installing the end-to-end high availability policy for SAP

Defining and installing an end-to-end policy is described in *IBM Tivoli System Automation for Multiplatforms End-to-End Automation Management User's Guide and Reference* (see Table 49 on page 371). In general, you can define an end-to-end policy XML file from scratch or copy the following sample XML file (Figure 65 on page 232) and adapt it to your environment. Before you can activate the policy, you must put it in the policy pool directory of the end-to-end automation manager machine:

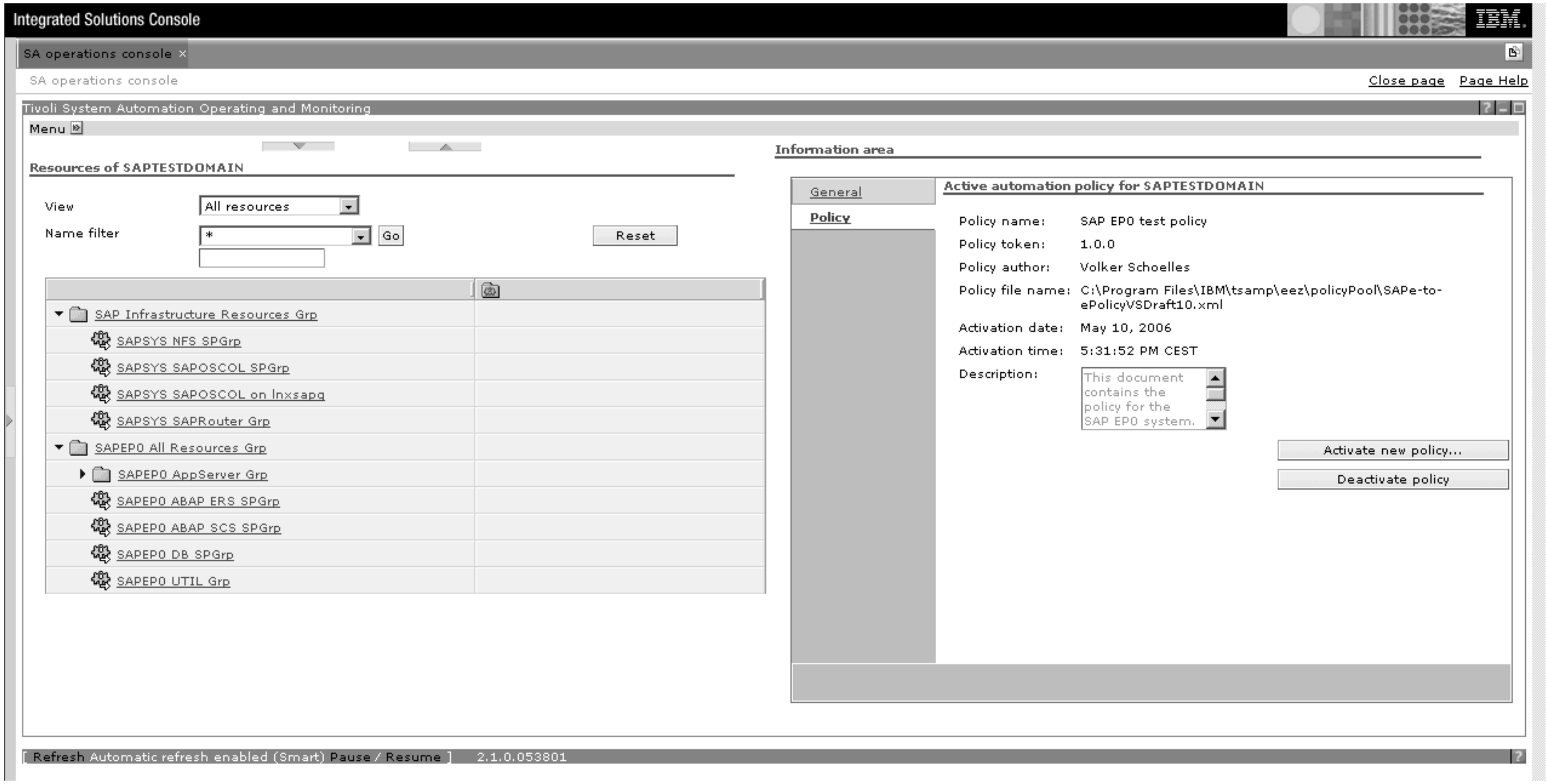
```
/etc/opt/IBM/tsamp/eez/policyPool
```

under Linux and AIX or

```
C:\Program Files\IBM\tsamp\eez\policyPool
```

under Windows. 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 follows:


Integrated Solutions Console
IBM.
SA operations console ×
SA operations console
Close page
Page Help
Tivoli System Automation Operating and Monitoring
Menu
Go
Reset
Activate new policy...
Deactivate policy
Refresh Automatic refresh enabled (Smart) Pause / Resume
2.1.0.053801

Resources of SAPTESTDOMAIN

View: All resources

Name filter: *

- SAP Infrastructure Resources Grp
  - SAPSYS NFS SPGrp
  - SAPSYS SAPOSCOL SPGrp
  - SAPSYS SAPOSCOL on lnxsapg
  - SAPSYS SAPRouter Grp
- SAPEP0 All Resources Grp
  - SAPEP0 AppServer Grp
  - SAPEP0 ABAP ERS SPGrp
  - SAPEP0 ABAP SCS SPGrp
  - SAPEP0 DB SPGrp
  - SAPEP0 UTIL Grp

Information area

General

Policy

Active automation policy for SAPTESTDOMAIN

Policy name:	SAP EP0 test policy
Policy token:	1.0.0
Policy author:	Volker Schoelles
Policy file name:	C:\Program Files\IBM\tsamp\eez\policyPool\SAPe-to-ePolicyVSDraft10.xml
Activation date:	May 10, 2006
Activation time:	5:31:52 PM CEST
Description:	This document contains the policy for the SAP EP0 system.

Figure 64. Sample policy as depicted by the SA MP user interface

The corresponding XML file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>

<AutomationPolicy version="1.0"
    xmlns="http://www.ibm.com/TSA/Policy.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.ibm.com/TSA/Policy.xsd EEZPolicy.xsd ">
          <PolicyInformation>
               <PolicyName>SAP EP0 test policy</PolicyName>
               <AutomationDomainName>SAPTESTDOMAIN </AutomationDomainName>
               <PolicyToken>1.0.0 </PolicyToken>
               <PolicyAuthor>Volker Schoelles</PolicyAuthor>
               <PolicyDescription>
               This document contains the policy for the SAP EP0 system.
               The SAP EP0 system runs with a 6.40 kernel and DB2 V8 on z/OS.
               All non z/OS Appl. server use DB2 Connect as SQL interface.

               Last Update: 10th of May 2006 by Volker Schoelles
               Change History:
               ------------------------------------------------------
               Date       Name             Description
               ------------------------------------------------------
               03/15/05   Volker Schoelles Initial Policy
               08/18/05   Volker Schoelles Moved SAPRouter to TSA MP
               10/14/05   Volker Schoelles Only 1 SAPOSCOL move group
               11/02/05   Volker Schoelles Added AppServer on lnxsaph
               04/06/06   Volker Schoelles Added ERS reference
               05/10/06   Volker Schoelles Added SAPOSCOL on lnxsapg
               ------------------------------------------------------
               </PolicyDescription>
          </PolicyInformation>

<ResourceReference name="SAPEP0 DB SPGrp">
         <Description>This resource reference refers to SAP EP0 system's Database resources on FCSPLEX</Description>
         <Owner>VS</Owner>
         <InfoLink>In case of a p,roblem: Please contact Mr. Schmitt (x-1234) from z/OS DB2 support.</InfoLink>
         <DesiredState>Offline</DesiredState>
<!--  The default DesiredState is offline. It will be set to online by the group to which this Res.Ref belongs. -->
                <ReferencedResource>
                <AutomationDomain>FCSPLEX</AutomationDomain>
                <Name>SAPEP0_DBX</Name>
                <Class>APG</Class>
                </ReferencedResource>
         </ResourceReference>

<!--  SAP application server can have a StartAfter relationship to the SCS resource... -->

<ResourceReference name="SAPEP0 ABAP SCS SPGrp">
         <Description>This resource reference refers to SAP EP0 system's ABAP Central Services Group on FCSPLEX</Description>
         <Owner>VS</Owner>
         <InfoLink></InfoLink>
                <ReferencedResource>
                <AutomationDomain>FCSPLEX</AutomationDomain>
                <Name>SAPEP0_ENQX</Name>
                <Class>APG</Class>
                </ReferencedResource>
         </ResourceReference>

<ResourceReference name="SAPEP0 ABAP ERS SPGrp">
         <Description>This resource reference refers to SAP EP0 system's ABAP Enqueue Replication Server Group on FCSPLEX</Description>
         <Owner>VS</Owner>
         <InfoLink></InfoLink>
                <ReferencedResource>
                <AutomationDomain>FCSPLEX</AutomationDomain>
                <Name>SAPEP0_ERSX</Name>
                <Class>APG</Class>
                </ReferencedResource>
         </ResourceReference>


<!--  Non z/OS SAP application server resource references... -->

<ResourceReference name="SAPEP0 AppServer D95 on lnxsapg">
         <Description>This resource reference refers to the Application Server Group on lnxsapg for D95 AppServer instance </Description>
         <Owner>VS</Owner>
         <InfoLink>In case of a problem: Please contact Mr. Mueller (x-4321) from SAP Basis support for Linux.</InfoLink>
                <ReferencedResource>
                <AutomationDomain>sap</AutomationDomain>
                <Name>SAP_EP0_lnxsapg_D95</Name>
                <Class>IBM.ResourceGroup</Class>
```

Figure 65. XML file defining the sample SA MP automation policy (Part 1 of 3)

```
                </ReferencedResource>
         </ResourceReference>

<ResourceReference name="SAPEP0 AppServer D97 on lnxsaph">
         <Description>This resource reference refers to the Application Server Group on lnxsaph for D97 AppServer instance </Description>
         <Owner>VS</Owner>
         <InfoLink>In case of a problem: Please contact Mr. Mueller (x-4321) from SAP Basis support for Linux.</InfoLink>
                <ReferencedResource>
                <AutomationDomain>sap</AutomationDomain>
                <Name>SAP_EP0_lnxsaph_D97</Name>
                <Class>IBM.ResourceGroup</Class>
                </ReferencedResource>
         </ResourceReference>

<ResourceReference name="SAPEP0 UTIL Grp">
         <Description>This resource reference refers to SAP EP0 system's Utility resources on FCSPLEX</Description>
         <Owner>VS</Owner>
         <InfoLink></InfoLink>
                <ReferencedResource>
                <AutomationDomain>FCSPLEX</AutomationDomain>
                <Name>SAPEP0_UTIL</Name>
                <Class>APG</Class>
                </ReferencedResource>
         </ResourceReference>

<ResourceReference name="SAPSYS SAPRouter Grp">
         <Description>This resource reference refers to the SAProuter on TSA MP domain sap</Description>
         <Owner>VS</Owner>
         <InfoLink></InfoLink>
                <ReferencedResource>
                <AutomationDomain>sap</AutomationDomain>
                <Name>SAP_SYS_ROUTER</Name>
                <Class>IBM.ResourceGroup</Class>
                </ReferencedResource>
         </ResourceReference>

<ResourceReference name="SAPSYS SAPOSCOL SPGrp">
         <Description>This resource reference refers to the SAPOSCOL on z/OS sxsplex</Description>
         <Owner>VS</Owner>
         <InfoLink></InfoLink>
                <ReferencedResource>
                <AutomationDomain>FCSPLEX</AutomationDomain>
                <Name>SAPSYS_OSCX</Name>
                <Class>APG</Class>
                </ReferencedResource>
         </ResourceReference>

<ResourceReference name="SAPSYS NFS SPGrp">
         <Description>This resource reference refers to the SAP NFS Server on zOS sysplex</Description>
         <Owner>VS</Owner>

        <InfoLink></InfoLink>
                <ReferencedResource>
                <AutomationDomain>FCSPLEX</AutomationDomain>
                <Name>SAPSYS_NFSX</Name>
                <Class>APG</Class>
                </ReferencedResource>
         </ResourceReference>

<!--  Non z/OS SAP SAPOSCOL resource references... -->

<ResourceReference name="SAPSYS SAPOSCOL on lnxsapg">
         <Description>This resource reference refers to the SAPOSCOL resource group on lnxsapg machine </Description>
         <Owner>VS</Owner>
         <InfoLink>In case of a problem: Please contact Mr. Maier (x-6543) from SAP Basis support for Linux.</InfoLink>
                <ReferencedResource>
                <AutomationDomain>sap</AutomationDomain>
                <Name>SAP_SYS_SAPOSCOL_LNXSAPG</Name>
                <Class>IBM.ResourceGroup</Class>
                </ReferencedResource>
         </ResourceReference>

<!--  Here start the end-to-end group definitions...  -->

<ResourceGroup name="SAPEP0 AppServer Grp" >
         <DesiredState>Offline</DesiredState>
<!--  The default DesiredState is offline. It will be set to online by the group to which this Res.Group belongs. -->
         <Description>This is the resource group for all AppServer of SAP system EP0 zSeries Linux cluster sap</Description>
         <Owner>VS</Owner>
         <InfoLink>http://www.ibm.com/help</InfoLink>
```

Figure 65. XML file defining the sample SA MP automation policy (Part 2 of 3)

```
        <Members><ResourceReference name="SAPEP0 AppServer D95 on lnxsapg"/>
                <ResourceReference name="SAPEP0 AppServer D97 on lnxsaph"/></Members>
        <!-- Additional Members can be specified like SAP_EP0_lnxsaph_D97 -->
        </ResourceGroup>

<ResourceGroup name="SAPEP0 All Resources Grp" >
        <DesiredState>Online</DesiredState>
        <Description>This is the resource group for all resources of SAP system EP0</Description>
        <Owner>VS</Owner>
        <InfoLink>http://www.ibm.com/help</InfoLink>
        <Members><ResourceReference name="SAPEP0 DB SPGrp"/>
                <ResourceReference name="SAPEP0 ABAP SCS SPGrp"/>
                <ResourceReference name="SAPEP0 ABAP ERS SPGrp"/>
                <ResourceReference name="SAPEP0 UTIL Grp"/>
                <ResourceGroup name="SAPEP0 AppServer Grp"/></Members>
        </ResourceGroup>

<ResourceGroup name="SAP Infrastructure Resources Grp" >
        <DesiredState>Online</DesiredState>
        <Description>This is the resource group for all infrastructure resources of a SAP system</Description>
        <Owner>VS</Owner>
        <InfoLink>http://www.ibm.com/help</InfoLink>
        <Members><ResourceReference name="SAPSYS SAPRouter Grp"/>
                <ResourceReference name="SAPSYS SAPOSCOL SPGrp"/>
                <ResourceReference name="SAPSYS SAPOSCOL on lnxsapg"/>
                <ResourceReference name="SAPSYS NFS SPGrp"/></Members>
        </ResourceGroup>

<!--  Here start the end-to-end relationship definitions...  -->

<Relationship>
        <Source><ResourceGroup name="SAPEP0 AppServer Grp"/></Source>
        <Type>StartAfter</Type>
        <Target><ResourceReference name="SAPSYS NFS SPGrp"/></Target>
        </Relationship>

<Relationship>
        <Source><ResourceGroup name="SAPEP0 AppServer Grp"/></Source>
        <Type>StartAfter</Type>
        <Target><ResourceReference name="SAPEP0 ABAP SCS SPGrp"/></Target>
        </Relationship>

<Relationship>
        <Source><ResourceReference name="SAPSYS SAPRouter Grp"/></Source>
        <Type>StartAfter</Type>
        <Target><ResourceReference name="SAPSYS NFS SPGrp"/></Target>
        </Relationship>

</AutomationPolicy>
```

Figure 65. XML file defining the sample SA MP automation policy (Part 3 of 3)

Part 5. Verification and problem determination

Chapter 13. Verification and problem determination on z/OS

Verification procedures and failover scenarios

This chapter describes the test scenarios we designed and ran to test the SA z/OS[9] policy.

Our z/OS test environment employed DB2 V7. We therefore had to run the ICLI client/server as the remote SQL interface. All descriptions in this chapter therefore include the ICLI server as a resource managed by SA z/OS. With SAP NetWeaver '04 and DB2 V8, the ICLI client/server is no longer supported. Instead, DB2's DDF address space is supported on z/OS and DB2 Connect on remote application servers. Since the DDF address space is part of the standard z/OS DB2 automation policy, you will no longer need ICLI server resources.

Overview of the test scenarios

Before defining and running test scenarios to verify the SA OS/390 policy, we made the following assumptions:

- The z/OS and network configuration had been done.
- The high availability solution had been installed.
- The SA OS/390 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 25. Examples of test scenarios

	Planned outages	**Unplanned outages**
Component	• Shutdown of a DB2 subsystem for maintenance • Stop of an SAP application server for kernel upgrade	• Failure of a TCP/IP stack • Failure of the enqueue server
LPAR	• Shutdown of an LPAR for hardware upgrade • Shutdown of an LPAR for re-IPLing	• Power outage • Unrecoverable operating system failure

Test scenarios to verify the SA OS/390 policy

We built a list of test scenarios, including planned and unplanned outages, to verify the SA OS/390 policy.

Planned outage scenarios:

[9] At the time of this test, the product was designated as System Automation for OS/390.

- 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 application server on z/OS
  - Start and stop of an 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 ICLI server[1]
  - An application server on z/OS
  - An application server on Linux on System z
  - A DB2 subsystem
  - The NFS server
  - The syslog collector
  - A syslog sender
  - The SAP gateway
  - Saprouter
  - Saposcol
  - Rfcoscol
- Failure of a network component:
  - A TCP/IP stack on z/OS
  - OSPF (OMPROUTE)
  - A network adapter on 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
  - An ICLI server[1]
- 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 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.

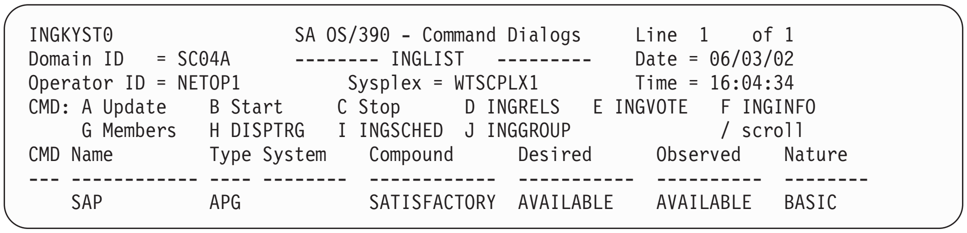
Verification of resource status

Before each test, we used the following checklist to review the status of all the SAP-related resources defined in SA OS/390:

1. Do all the resources monitored by SA OS/390 have a compound status SATISFACTORY?

   Tip: The NetView command INGLIST SAP/APG displays the status of the application group SAP. If the compound status is SATISFACTORY, then we know that all resources belonging to that group have a compound state SATISFACTORY. Otherwise, we can drill down the tree of resources using option G (Members).

   The following is a sample output of the NetView command INGLIST SAP/APG, showing the application group SAP with a compound status of SATISFACTORY:

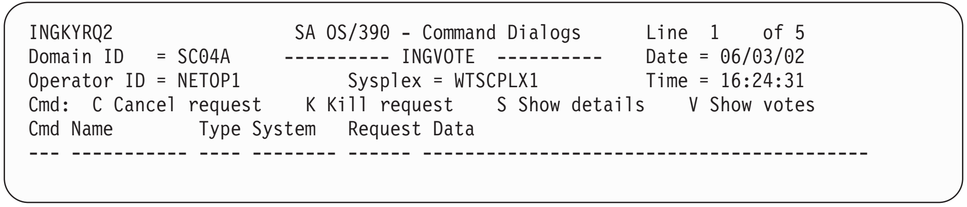


```
INGKYST0                 SA OS/390 - Command Dialogs     Line  1    of 1
Domain ID   = SC04A      -------- INGLIST   ---------    Date = 06/03/02
Operator ID = NETOP1          Sysplex = WTSCPLX1         Time = 16:04:34
CMD: A Update    B Start     C Stop      D INGRELS   E INGVOTE   F INGINFO
     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 OS/390?

   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 OS/390?

   Note: There is no command to display all the excludes in SA OS/390 at once. Individual INGINFO commands must be issued against every application group defined as SYSPLEX/MOVE groups.

   In our configuration, we used the following commands:

```
INGINFO RED_EMPLEX
INGINFO RED_ERSPLEX
INGINFO NFS_HAPLEX
INGINFO RED_RASPLEX
INGINFO SAP_RTPLEX
INGINFO RED_COPLEX
INGINFO RED_VPLEX
```

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:

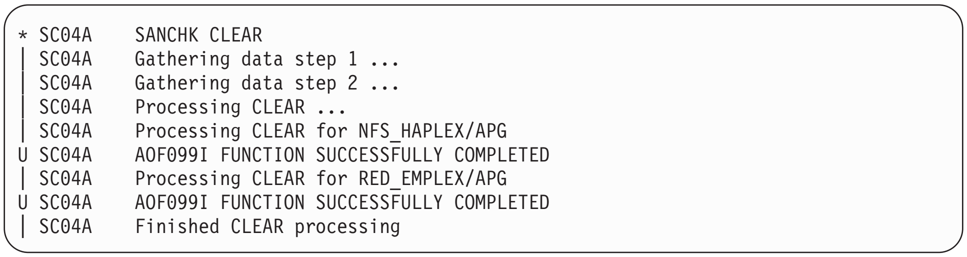
```
INGGROUP RED_EMPLEX/APG ACTION=INCLUDE SYSTEMS=SC42
```

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 OS/390. The code source for this procedure can be found in “SANCHK” on page 312. 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
```

We can also use the REXX procedure SANCHK with the option CLEAR to remove all the excludes:



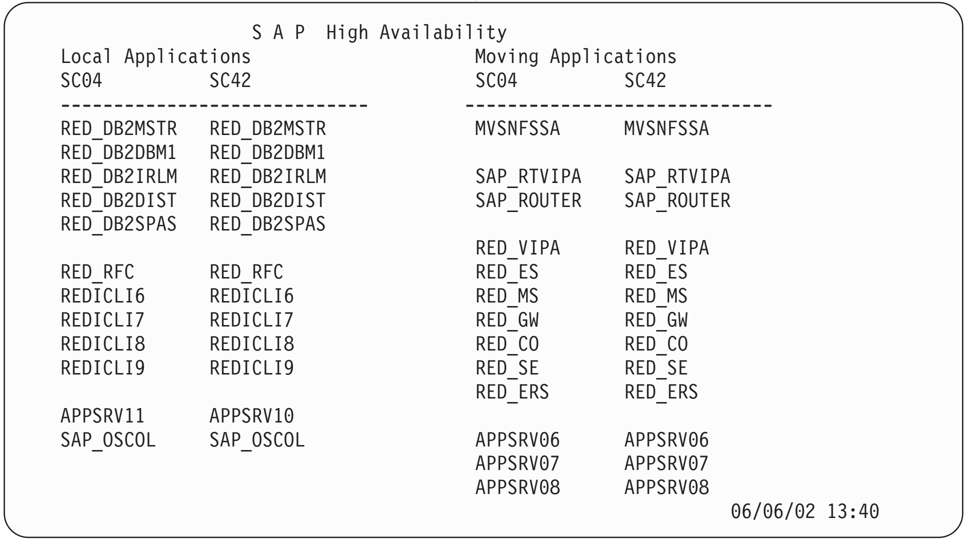
```
* SC04A    SANCHK CLEAR
| SC04A    Gathering data step 1 ...
| SC04A    Gathering data step 2 ...
| SC04A    Processing CLEAR ...
| SC04A    Processing CLEAR for NFS_HAPLEX/APG
U SC04A    AOF099I FUNCTION SUCCESSFULLY COMPLETED
| SC04A    Processing CLEAR for RED_EMPLEX/APG
U SC04A    AOF099I FUNCTION SUCCESSFULLY COMPLETED
| SC04A    Finished CLEAR processing
```

4. Where are the enqueue server, message server, enqueue replication server and NFS server running before the test?

   Note: We customized an SDF panel to monitor all the SAP-related resources and to see on which system they are running (for more information, refer to “Sending UNIX messages to the syslog” on page 163.

   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_DB2SPAS   RED_DB2SPAS
                                      RED_VIPA      RED_VIPA
 RED_RFC       RED_RFC                RED_ES        RED_ES
 REDICLI6      REDICLI6               RED_MS        RED_MS
 REDICLI7      REDICLI7               RED_GW        RED_GW
 REDICLI8      REDICLI8               RED_CO        RED_CO
 REDICLI9      REDICLI9               RED_SE        RED_SE
                                      RED_ERS       RED_ERS
 APPSRV11      APPSRV10
 SAP_OSCOL     SAP_OSCOL              APPSRV06      APPSRV06
                                      APPSRV07      APPSRV07
                                      APPSRV08      APPSRV08
                                                          06/06/02 13:40
```

5. Are the NFS file systems mounted on the remote application server accessible?

   Tip: We either logon to the remote application server and display the available file systems (using the UNIX command *df*), or we use the SAP transaction AL11 to check that we can access the files in the SAP directories.

Preparation for the test (unplanned outage only)

During the unplanned outage scenarios, we want to verify the impact of the failure for end users and for any workload that would be running on the system. Therefore, before each test, we execute the following preparation steps:

1. Log on to all the application servers.
2. Create an SAP workload.

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:

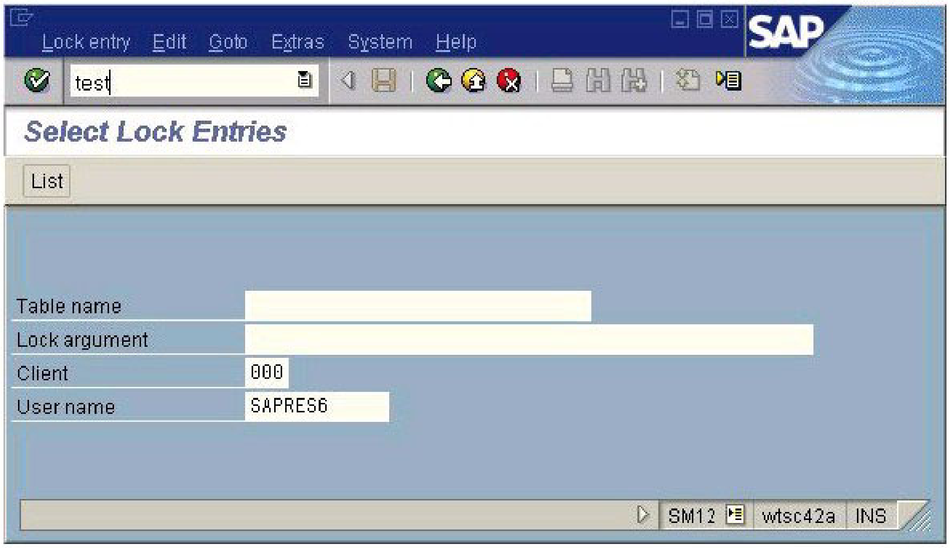


Table name

Lock argument

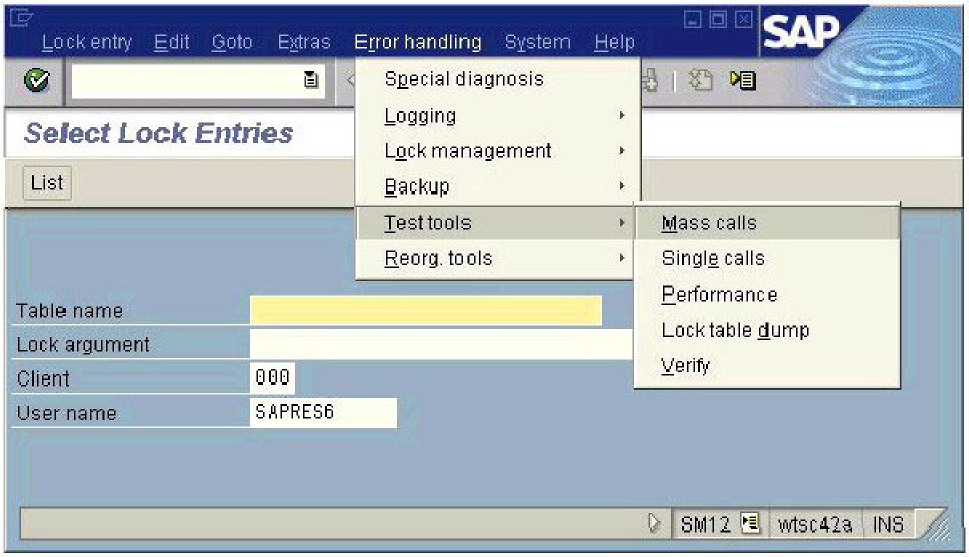
Client 000

User name SAPRES6

Figure 66. SM12 primary panel

A new selection appears in the menu bar: “Error handling”.

Click **Error handling** → **Test tools** → **Mass calls**



Select Lock Entries

Table name

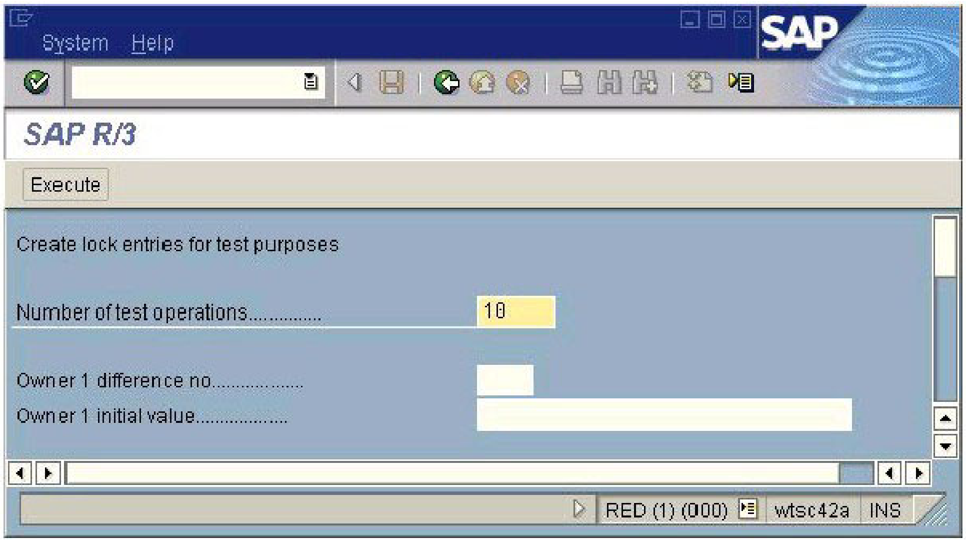
Lock argument

Client 000

User name SAPRES6

Figure 67. 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:



SAP R/3

Create lock entries for test purposes

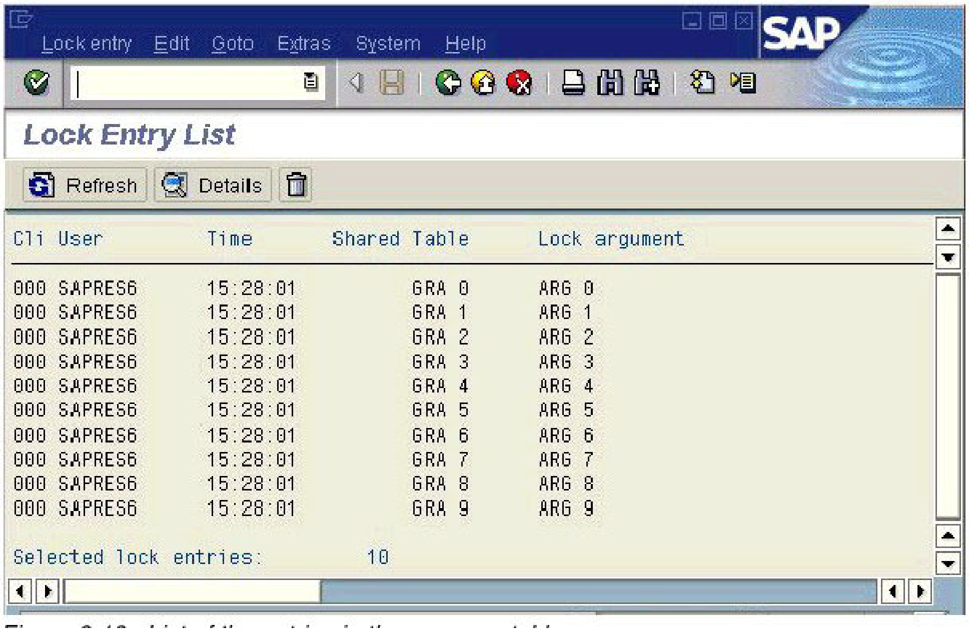
Number of test operations.............. 10

Owner 1 difference no..................

Owner 1 initial value..................

Figure 68. 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:



```
Cli User          Time       Shared Table      Lock argument

000 SAPRES6       15:28:01          GRA 0      ARG 0
000 SAPRES6       15:28:01          GRA 1      ARG 1
000 SAPRES6       15:28:01          GRA 2      ARG 2
000 SAPRES6       15:28:01          GRA 3      ARG 3
000 SAPRES6       15:28:01          GRA 4      ARG 4
000 SAPRES6       15:28:01          GRA 5      ARG 5
000 SAPRES6       15:28:01          GRA 6      ARG 6
000 SAPRES6       15:28:01          GRA 7      ARG 7
000 SAPRES6       15:28:01          GRA 8      ARG 8
000 SAPRES6       15:28:01          GRA 9      ARG 9

Selected lock entries:         10
```

Figure 69. 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 OS/390 must be used for the following tasks:
  - Starting and stopping of resources
  - Moving of resources
  - Excluding resources on specific systems
  - Initiating SA OS/390 vote requests against resources
- To simulate a failure or an unplanned outage of resources, an external action must be taken, such as:
  - Kill a UNIX process ID
  - Cancel or stop an address space
  - Reset an LPAR
  - Stop a network adapter or power down a switch
  - Pull a cable

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 240).

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)
- Looking at the ICLI message log ICLI.<pid>.err1, if you still run the ICLI server in your HA environment
- Displaying the DB2 threads using the DB2 command -DIS THREAD(*)

Note: A trace file called enquelog 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/enquelog
```

.

```
 Start: Thu May 30 11:34:57 2002: enqueue server started
RepAct: Thu May 30 11:41:22 2002: replication activated
RepDea: Thu May 30 14:15:20 2002: replication deactivated
  Stop: Thu May 30 14:15:36 2002: enqueue server stopped: normal shutdown
 Start: Thu May 30 14:16:20 2002: enqueue server started
RepAct: Thu May 30 14:21:39 2002: replication activated
```

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 267)

Planned outage test scenarios

This section describes the planned outage test scenarios we chose to perform in order to verify the SA OS/390 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 240, 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 OS/390. 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 26. Stop of the entire SAP system with SA OS/390

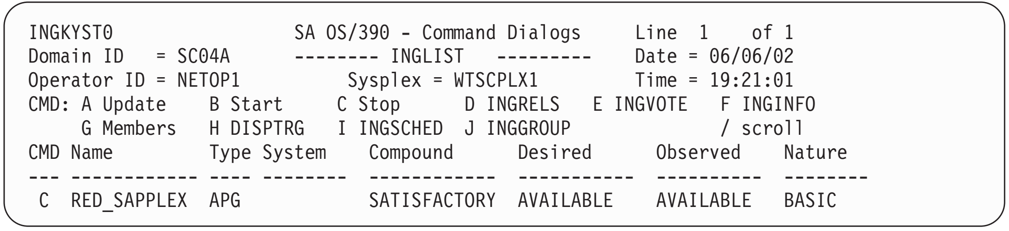
Purpose	Scope: The entire SAP RED system Action: Planned stop using SA OS/390
Expected behavior	All RED-related resources should come down properly. The NFS server, SAProuter, and saposcol should stay up.
Setup	SC42 and SC04 must be up, including all required z/OS resources and SAP-related resources.
Execution	Issue a STOP request in SA OS/390 against the application group RED_SAPPLEX.
Results	All RED-related resources came down properly. The NFS server, SAProuter, and saposcol stayed up.

Table 27 summarizes the execution of the start phase.

Table 27. Start of the entire SAP system with SA OS/390

Purpose	Scope: The entire SAP RED system Action: Planned start using SA OS/390
Expected behavior	All RED-related resources should come up properly.
Setup	SC42 and SC04 must be up, with all required z/OS resources, but all RED-related resources are stopped.
Execution	Kill the STOP request in SA OS/390 against the application group RED_SAPPLEX.
Results	All RED-related resources came up properly.

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.

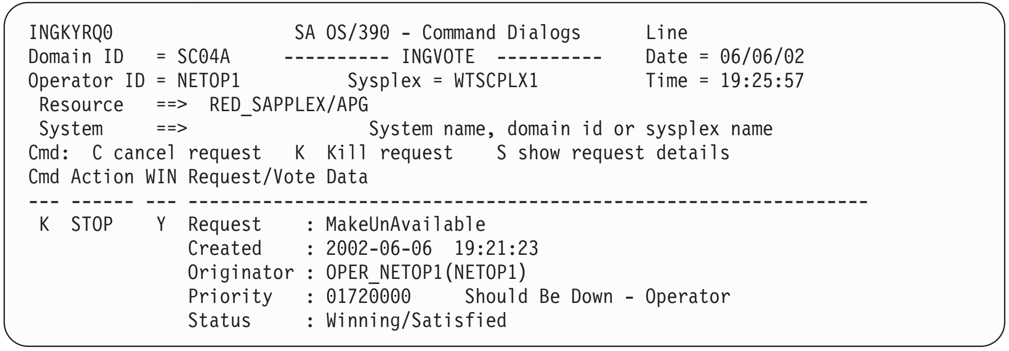
Note: Because of our SA OS/390 definitions, only the monitor for the remote application server running on Linux stopped. The application server itself stayed idle until the system was up again, and then it reconnected.

If we wanted to stop the remote application server, we needed to issue a STOP request with the option FORCE on the application group RED_RASPLEX before stopping the group RED_SAPPLEX.

On our SDF customized panel we checked the status of all the RED-related resources. All the resources went from an UP status to a STOPPING status, and finally to an AUTODOWN status. The NFS server, saposcol, and SAProuter were still running.

Note: The SA OS/390 resource APPSRV06 appears with an AUTODOWN status although the remote application server APPSRV06 is still running on Linux. Only the monitor has stopped.

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 28 summarizes the startup of the first LPAR.

Table 28. 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, ICLI, application server, rfcoscol, 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, ICLI, application server, rfcoscol, and saposcol, plus NFS server and enqueue server, but not enqueue replication server.

Table 29 summarizes the startup of the second LPAR.

Table 29. 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, ICLI, application server, rfcoscol, and saposcol, plus enqueue replication server.
Setup	The first LPAR must be up with all required z/OS resources and SAP-related resources: database server, ICLI, application server, rfcoscol, 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, ICLI, application server, rfcoscol, 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 30 summarizes the execution of the shutdown phase.

Table 30. 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
Expected behavior	The NFS server should move to the other LPAR. The enqueue server should move to the other LPAR. The enqueue replication server should stop or move to another LPAR if more than two LPARs are available. The application server on the remaining LPAR should reconnect to the message server and enqueue server. The LPAR should come down properly to the point where we can enter the following command to remove the LPAR from the sysplex: */V XCF,<sysname>,OFFLINE*
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: • The enqueue server running on SC04. • The enqueue replication server running on SC42. • The NFS server running on SC04.

Table 30. Shutdown of the LPAR where the ES and NFS servers are running (continued)

Execution	Move the SAP critical components running on SC04 to SC42 (NFS server, enqueue server, and SAProuter). Stop the remaining SAP-related resources on SC04 (application server, rfcoscol, saposcol, ICLI servers, and database server). Issue a STOP request in SA OS/390 against the system group SC04 using the NetView command SHUTSYS ALL.
Verifications	Check that the application server APPSRV10 on SC42 reconnects to the message server and enqueue server.
Results	The NFS server moved from SC04 to SC42. The enqueue server moved from SC04 to SC42. The enqueue replication server stopped (the application group RED_ERSPLEX has a status INHIBITED). The application server APPSRV10 on SC42 reconnected to the message server and enqueue server. SC04 came down properly to the point where we can enter: */V XCF,SC04,OFFLINE*

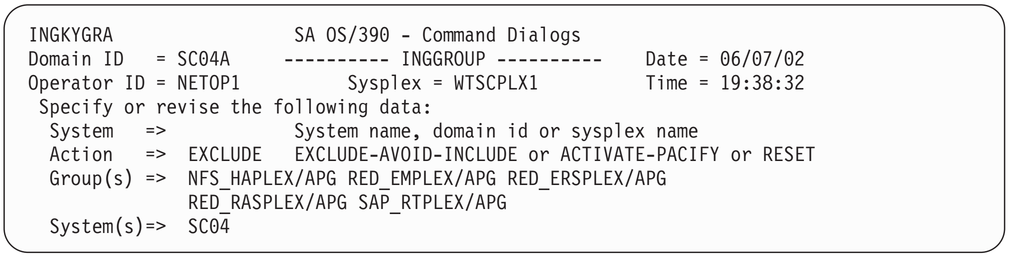
Table 31 summarizes the execution of the restart phase.

Table 31. Restart of the LPAR where the ES and NFS servers are running

Purpose	Scope: One LPAR Action: Restart after planned shutdown of the LPAR where the enqueue server and the NFS server are running (in our case SC04)
Expected behavior	SC04 should come up with all required address spaces including database server, ICLI, application server, rfcoscol, and saposcol. The enqueue server and the NFS server should stay on SC42. The enqueue replication server should restart to SC04.
Setup	SC42 must be up, including all required z/OS resources and SAP-related resources: database server, ICLI, application server, rfcoscol, 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, ICLI, application server, rfcoscol, and saposcol. The enqueue server and the NFS server stayed on SC42. The enqueue replication server was restarted on SC04.

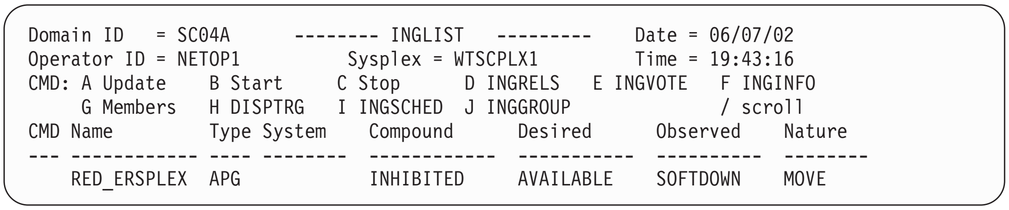
All the SAP-related resources are in UP status. The NFS server and the enqueue server are running on SC04. The enqueue replication server is running on SC42.

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 OS/390 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_EMPLEX/APG RED_ERSPLEX/APG
               RED_RASPLEX/APG SAP_RTPLEX/APG
  System(s)=>  SC04
```

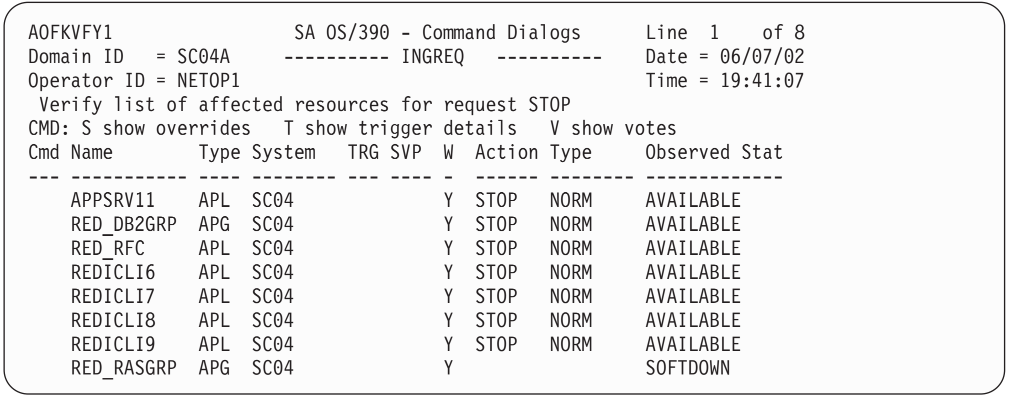
Note that we also excluded SC04 for the resource RED_ERSPLEX. If we had had a third system, the enqueue replication server would have moved to that system. In our configuration, the enqueue replication server stopped and the application group RED_ERSPLEX remained in an INHIBITED status:



```
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 application server APPSRV11, the ICLI servers, saposcol, rfcoscol, and the DB2 subsystem.

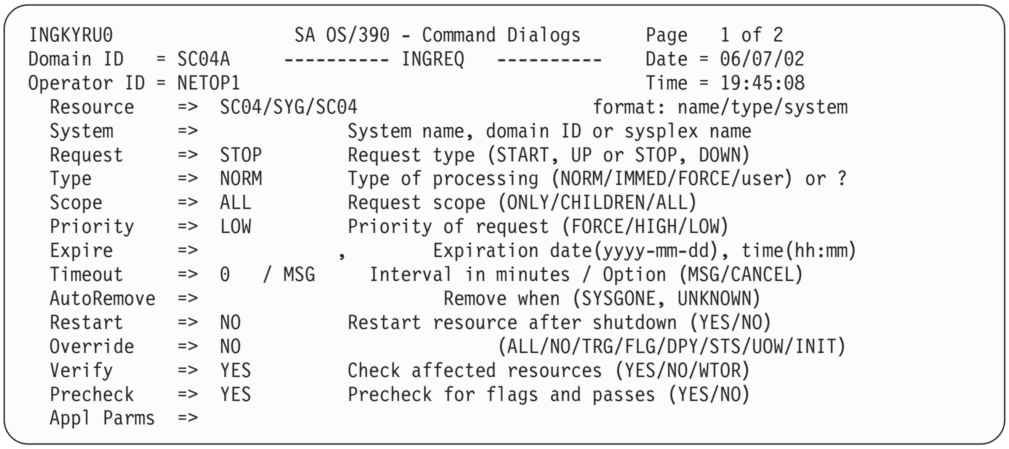
Because of all the dependencies defined in the SA OS/390 policy, issuing a STOP request against the application group RED_DB2GRP on SC04 not only stops the DB2 subsystem, but if the parameter scope is set to ALL (default value), it also stops all the children: the application server APPSRV11, the ICLI servers, and rfcoscol. SA OS/390 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
    REDICLI6    APL  SC04              Y  STOP   NORM     AVAILABLE
    REDICLI7    APL  SC04              Y  STOP   NORM     AVAILABLE
    REDICLI8    APL  SC04              Y  STOP   NORM     AVAILABLE
    REDICLI9    APL  SC04              Y  STOP   NORM     AVAILABLE
    RED_RASGRP  APG  SC04              Y                  SOFTDOWN
```

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 Parms  =>
```

SC04 came down to the point where we could enter the following MVS command to remove SC04 from the sysplex:

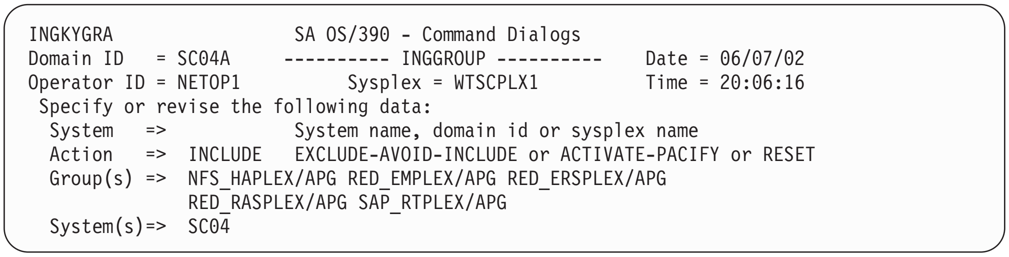
```
/V XCF,SC04,OFFLINE
```

We checked that the 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 OS/390 was started automatically and restarted all the resources on the system, including the DB2 subsystem, the ICLI servers, the application server APPSRV11, rfcoscol, and saposcol.

The enqueue replication server was not restarted because we still had the exclude of SC04 on the application group RED_ERSPLEX. To restart it, we removed this exclude (and all the outstanding excludes) using the NetView command INGGROUP:



```
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_EMPLEX/APG RED_ERSPLEX/APG
               RED_RASPLEX/APG SAP_RTPLEX/APG
  System(s)=>  SC04
```

As described in “Verification of resource status” on page 240, 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/enquelog
```

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 OS/390 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 240, 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 OS/390. We also wanted to measure the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 32. Failure of the enqueue server

Purpose	Scope: Enqueue server Action: Unplanned outage
Expected behavior	SA OS/390 should show a PROBLEM/HARDDOWN status for the resource RED_ES and restart 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: • The enqueue server running on SC42. • The enqueue replication server running on SC04. • The NFS server running on SC42.

Table 32. Failure of the enqueue server (continued)

Preparation	Log on to all the application servers. Create a workload on one application server (APPSRV11 on SC04). Create entries in the enqueue table.
Execution	Use the UNIX command *kill -9* to kill the enqueue server process externally (from SA OS/390).
Verifications	Check that the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Look for error messages in the enqueue log file , in the dev_enqserv file, in the developer traces dev_disp and dev_wx, and in the system log (SM21).
Results	SA OS/390 showed a PROBLEM/HARDDOWN status for RED_ES on SC42 and restarted 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 242, we logged on to all the application servers, created a workload on APPSRV11 (five parallel tasks), and generated 10 lock entries in the enqueue table.

Then we simulated the failure: we killed the enqueue server process from SA OS/390, using the UNIX command *kill -9 <pid>*:

```
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 ttyp0002  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.

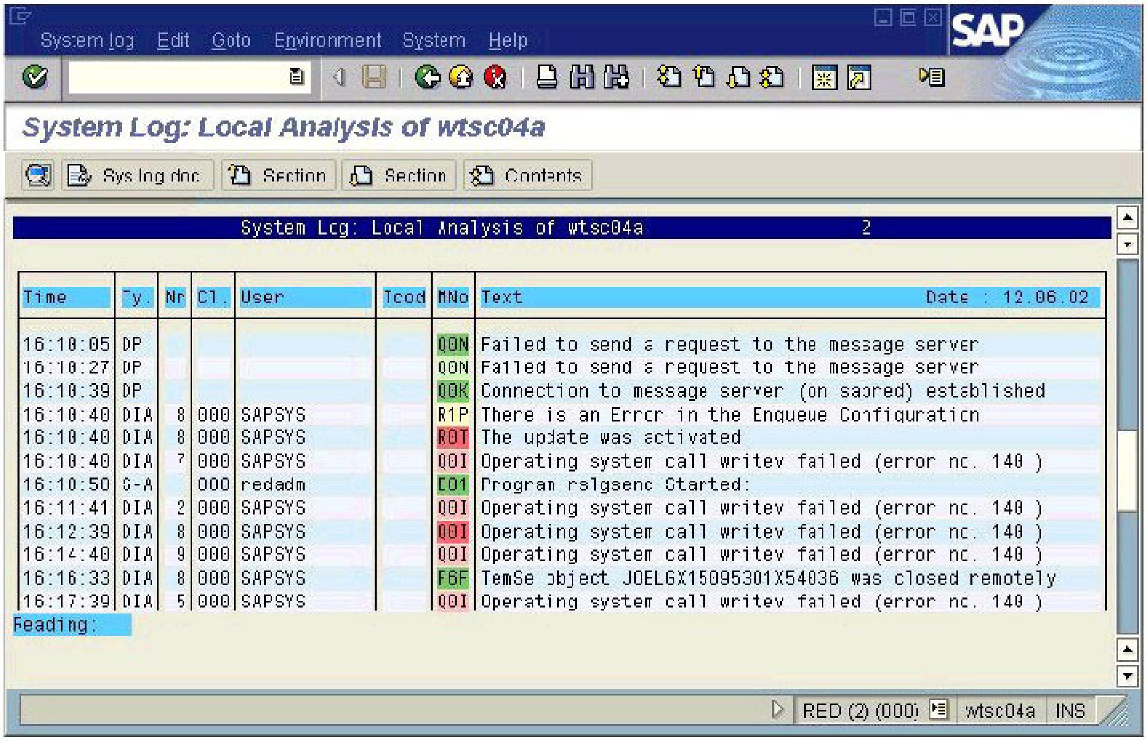
When the enqueue server restarts on SC04, it reads the enqueue replication table from shared memory and rebuilds the enqueue table. Using the transaction SM12, we verified that the 10 lock entries we had generated were still in the enqueue table.

Looking at the enqueue server log file (enquelog), 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.



System Log: Local Analysis of wtsc04a

```
                   System Log: Local Analysis of wtsc04a                   2

Time     |Ty. |Nr|Cl.|User         |Tcod|MNo|Text                                  Date : 12.06.02

16:10:05 |DP  |  |   |             |    |Q0N|Failed to send a request to the message server
16:10:27 |DP  |  |   |             |    |Q0N|Failed to send a request to the message server
16:10:39 |DP  |  |   |             |    |Q0K|Connection to message server (on sapred) established
16:10:40 |DIA | 8|000|SAPSYS       |    |R1P|There is an Error in the Enqueue Configuration
16:10:40 |DIA | 8|000|SAPSYS       |    |R0T|The update was activated
16:10:40 |DIA | 7|000|SAPSYS       |    |Q0I|Operating system call writev failed (error no. 140 )
16:10:50 |G-A |  |000|redadm       |    |E01|Program rslgsend Started:
16:11:41 |DIA | 2|000|SAPSYS       |    |Q0I|Operating system call writev failed (error no. 140 )
16:12:39 |DIA | 8|000|SAPSYS       |    |Q0I|Operating system call writev failed (error no. 140 )
16:14:40 |DIA | 9|000|SAPSYS       |    |Q0I|Operating system call writev failed (error no. 140 )
16:16:33 |DIA | 8|000|SAPSYS       |    |F6F|TemSe object JOELGX15095301X54036 was closed remotely
16:17:39 |DIA | 5|000|SAPSYS       |    |Q0I|Operating system call writev failed (error no. 140 )
Reading:
```

Figure 70. SAP system log (SM21)

Note that the system log shows a 6-minute interval before complete reconnection of the application server. This was due to a bug in TCP/IP (probably related to our multiple-stack environment). After we 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 had a third system, SA OS/390 would have restarted the enqueue replication server on that system.

We used the NetView command SETSTATE to tell SA OS/390 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 OS/390. We also wanted to measure the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 33. Failure of the message server

Purpose	Scope: Message server Action: Unplanned outage
Expected behavior	SA OS/390 should try to restart the message server in place until the critical threshold is reached (5 failures in 10 minutes). If the critical threshold is reached, SA OS/390 should show a PROBLEM/HARDDOWN status for the resource RED_MS and the entire 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: • The enqueue server running on SC42. • The enqueue replication server running on SC04. • The NFS server running on SC42.
Preparation	Log on to all the application servers. Create a workload on one application server (APPSRV11 on SC04). Create entries in the enqueue table.
Execution	Use the UNIX command *kill -9* to kill the message server process externally (out of SA OS/390).
Verifications	Check that the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Look for error messages in the developer trace dev_disp and in the system log (SM21).
Results	SA OS/390 restarted the message server in place, on SC42. The failure was transparent to the SAP workload.

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 242, we logged on to all the application servers, created a workload on APPSRV11 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

Then we simulated the failure: we killed the message server process out of SA OS/390, using the UNIX command *kill -9 <pid>*:

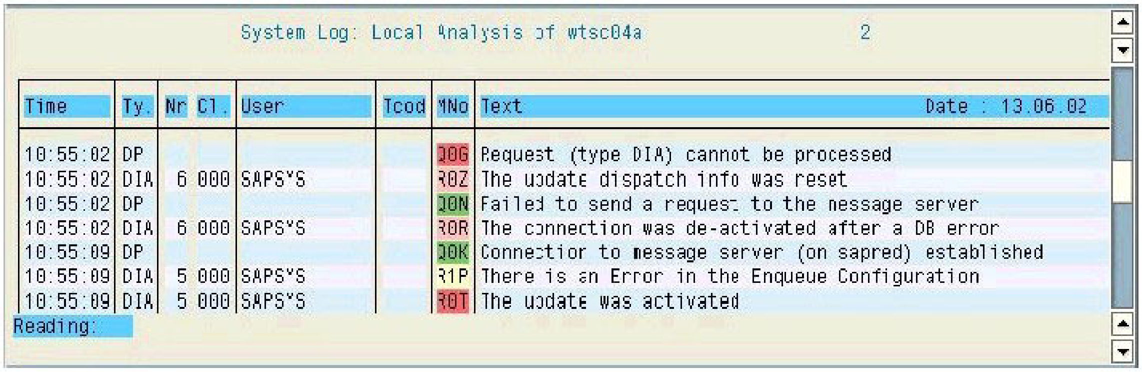
```
SC42>ps -ef | grep ms.sapRED_ASCS00
  redadm   34408866     854437  - 09:47:44 ?         0:00 ms.sapRED_ASCS00 pf=/usr
/sap/RED/SYS/profile/RED_ASCS00
     DFS     854747   51186380  - 10:54:55 ttyp0003  0:00 grep ms.sapRED_ASCS00
SC42>kill -9 34408866
```

Because the critical threshold was not reached, SA OS/390 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.



```
                  System Log: Local Analysis of wtsc04a              2

Time     Ty. Nr Cl. User        Tcod MNo Text                              Date : 13.06.02

10:55:02 DP                          Q0G Request (type DIA) cannot be processed
10:55:02 DIA  6 000 SAPSYS           R0Z The update dispatch info was reset
10:55:02 DP                          Q0N Failed to send a request to the message server
10:55:02 DIA  6 000 SAPSYS           R0R The connection was de-activated after a DB error
10:55:09 DP                          Q0K Connection to message server (on sapred) established
10:55:09 DIA  5 000 SAPSYS           R1P There is an Error in the Enqueue Configuration
10:55:09 DIA  5 000 SAPSYS           R0T The update was activated
Reading:
```

Figure 71. SAP system log (SM21)

Failure of the ICLI server

In this scenario, we wanted to simulate the failure of the ICLI server and test the behavior of SA OS/390. We also wanted to measure the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 34. Failure of the ICLI server

Purpose	Scope: ICLI server Action: Unplanned outage

Table 34. Failure of the ICLI server (continued)

Expected behavior	SA OS/390 should try to restart the ICLI server until the critical threshold is reached. When that happens, SA OS/390 should show a PROBLEM/HARDDOWN status and the ICLI server will not be restarted. Running transactions should be rolled back. Work processes should reconnect either to the same database server, or failover to the standby database server.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: • The enqueue server running on SC04. • The enqueue replication server running on SC42. • The NFS server running on SC04.
Preparation	Log on to the remote application server. Create a workload on the remote application server.
Execution	Cancel the address space REDICLI6 on SC42.
Verifications	Check if the workload is still running (SM66). Look for error messages in the system log (SM21) and in the developer traces dev_wx. Use transaction DB2 and the DB2 command -DIS THREAD(*) to determine where the application server is connected.
Results	Because the critical threshold was not reached, SA OS/390 restarted the ICLI server REDICLI6 in place, on SC42. Running transactions were rolled back. Because the ICLI server was restarted before failover timeout detection, work processes could reconnect to the database server on SC42.

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.

We logged on to the application server APPSRV06 running on VMLINUX6.

We displayed the current DB host using the SAP transaction DB2. (On the first panel of transaction DB2, we clicked *Installation parameters* → *Database analysis* → *Switch DB host* . We selected *Refresh* → *Execute)*. The following shows that, before the failure, APPSRV06 is connected to wtsc42a, its primary DB host. The standby DB host is wtsc04a.

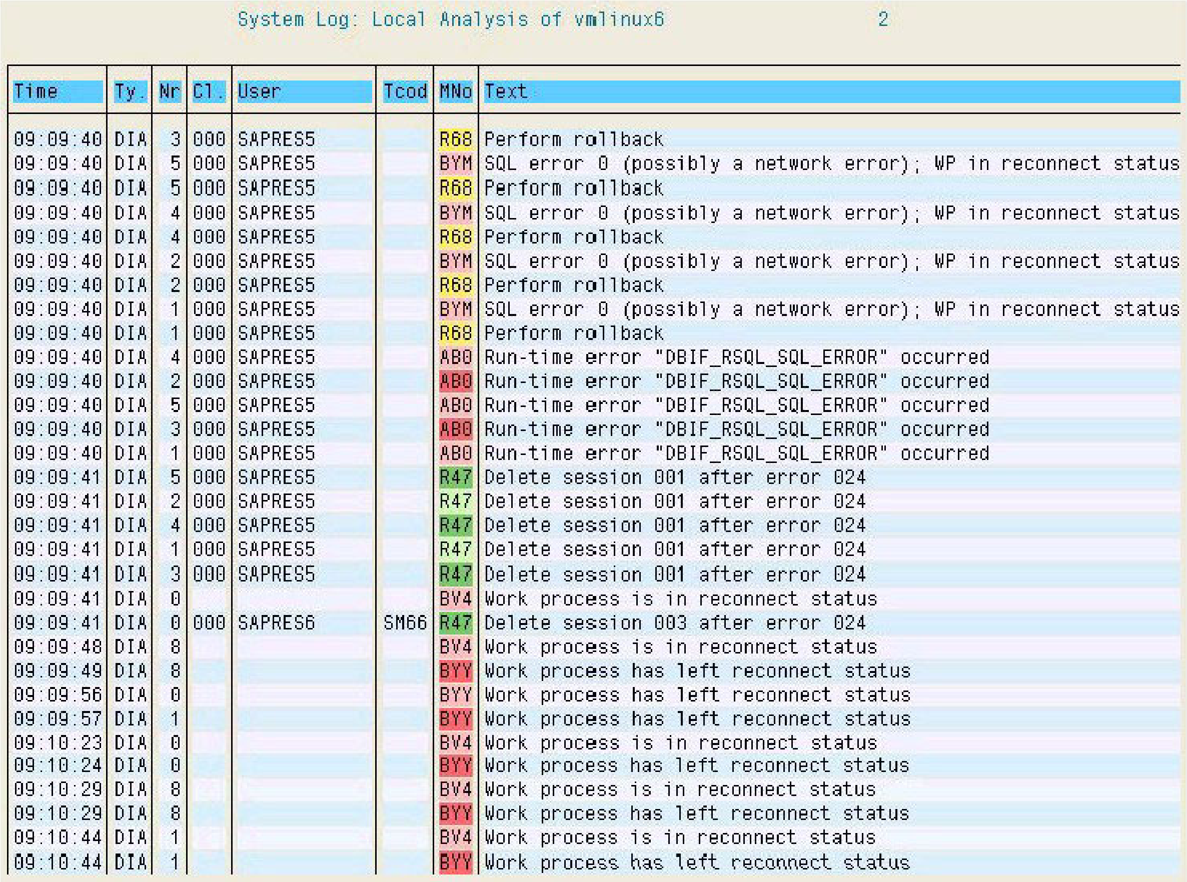
```
Settings:
Primary DB host                          wtsc42a
Standby DB host                          wtsc04a
Present DB host                          wtsc42a

Operation:
Operation completed successfully.
New DB host                              wtsc42a
```

We started the workload on APPSRV06 (5 parallel tasks). Then we simulated the failure by cancelling the ICLI server address space. Because the critical threshold was not reached, SA OS/390 immediately restarted the ICLI server in place, on SC42:

```
09:09:36.64 SAPRES6  00000290 -RO SC42,C REDICLI6
...
09:09:40.56 STC09771 00000090 $HASP395 REDICLI6 ENDED
...
09:09:41.42 AWRK0342 00000290 S REDICLI6
...
09:09:42.71 STC10710 00000090 $HASP373 REDICLI6 STARTED
```

The following log output shows the messages written in the SAP system log (SM21) during the interval of the test.



System Log: Local Analysis of vmlinux6 2

Time	Ty.	Nr	Cl.	User	Tcod	MNo	Text
09:09:40	DIA	3	000	SAPRES5		R68	Perform rollback
09:09:40	DIA	5	000	SAPRES5		BYM	SQL error 0 (possibly a network error); WP in reconnect status
09:09:40	DIA	5	000	SAPRES5		R68	Perform rollback
09:09:40	DIA	4	000	SAPRES5		BYM	SQL error 0 (possibly a network error); WP in reconnect status
09:09:40	DIA	4	000	SAPRES5		R68	Perform rollback
09:09:40	DIA	2	000	SAPRES5		BYM	SQL error 0 (possibly a network error); WP in reconnect status
09:09:40	DIA	2	000	SAPRES5		R68	Perform rollback
09:09:40	DIA	1	000	SAPRES5		BYM	SQL error 0 (possibly a network error); WP in reconnect status
09:09:40	DIA	1	000	SAPRES5		R68	Perform rollback
09:09:40	DIA	4	000	SAPRES5		AB0	Run-time error "DBIF_RSQL_SQL_ERROR" occurred
09:09:40	DIA	2	000	SAPRES5		AB0	Run-time error "DBIF_RSQL_SQL_ERROR" occurred
09:09:40	DIA	5	000	SAPRES5		AB0	Run-time error "DBIF_RSQL_SQL_ERROR" occurred
09:09:40	DIA	3	000	SAPRES5		AB0	Run-time error "DBIF_RSQL_SQL_ERROR" occurred
09:09:40	DIA	1	000	SAPRES5		AB0	Run-time error "DBIF_RSQL_SQL_ERROR" occurred
09:09:41	DIA	5	000	SAPRES5		R47	Delete session 001 after error 024
09:09:41	DIA	2	000	SAPRES5		R47	Delete session 001 after error 024
09:09:41	DIA	4	000	SAPRES5		R47	Delete session 001 after error 024
09:09:41	DIA	1	000	SAPRES5		R47	Delete session 001 after error 024
09:09:41	DIA	3	000	SAPRES5		R47	Delete session 001 after error 024
09:09:41	DIA	0				BV4	Work process is in reconnect status
09:09:41	DIA	0	000	SAPRES6	SM66	R47	Delete session 003 after error 024
09:09:48	DIA	8				BV4	Work process is in reconnect status
09:09:49	DIA	8				BYY	Work process has left reconnect status
09:09:56	DIA	0				BYY	Work process has left reconnect status
09:09:57	DIA	1				BYY	Work process has left reconnect status
09:10:23	DIA	0				BV4	Work process is in reconnect status
09:10:24	DIA	0				BYY	Work process has left reconnect status
09:10:29	DIA	8				BV4	Work process is in reconnect status
09:10:29	DIA	8				BYY	Work process has left reconnect status
09:10:44	DIA	1				BV4	Work process is in reconnect status
09:10:44	DIA	1				BYY	Work process has left reconnect status

Figure 72. SAP system log (SM21)

The five running transactions receive a DB2 SQL error 0 and are rolled back. The work processes are put in a reconnect status. The running sessions are lost and need to be restarted by the user. Within seconds, the work processes reestablish the connection and leave the reconnect status.

The transaction DB2 shows that the current DB host is still wtsc42a. We checked with the DB2 command -DIS THREAD(*) that all the threads were connected to SC42. Connection information for each work process can be found in the developer trace files, dev_wx.

Note: During our test, we observed that the work processes could reconnect to the primary database server. This was because the ICLI server was restarted before failover time-out detection. However, especially in the case of a heavy workload, you could experience a failover to the standby database server.

Failure of the NFS server

In this scenario, we wanted to simulate the failure of the NFS server and test the behavior of SA OS/390. We also wanted to measure the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 35. Failure of the NFS server

Purpose	Scope: NFS server Action: Unplanned outage
Expected behavior	SA OS/390 should restart the NFS server. Existing NFS mounts should be reestablished. The failure should be transparent to the SAP workload.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: • The enqueue server running on SC42. • The enqueue replication server running on SC04. • The NFS server running on SC42.
Preparation	Log on to all the application servers. Create a workload on a remote application server (APPSRV06). Create entries in the enqueue table.
Execution	Cancel the address space MVSNFSSA on SC42.
Verifications	Check that the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Check that the file systems are accessible (AL11). Look for error messages in the system log (SM21).
Results	SA OS/390 restarted the NFS server. Existing NFS mounts were reestablished. The failure was transparent to the SAP workload.

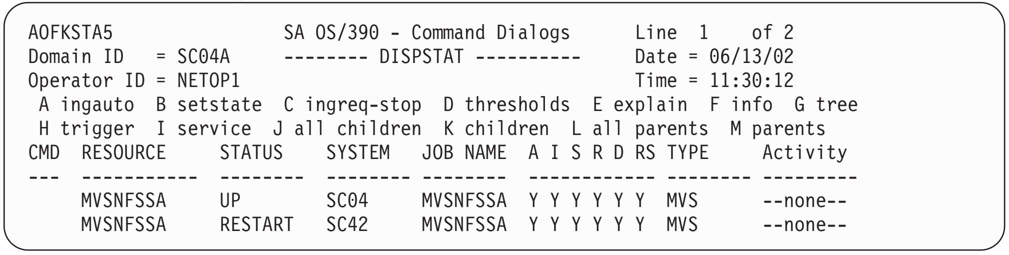
Before the test, all SAP-related resources are in UP status. The NFS and enqueue servers are running on SC42, and the enqueue replication server is running on SC04.

As described in “Preparation for the test (unplanned outage only)” on page 242, we logged on to all the application servers, created a workload on the remote application server APPSRV06 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

Then we simulated the failure by cancelling the address space of the NFS server on SC42 using the following command:

```
/C MVSNFSSA
```

Because, at the time of the test, the effective preference of SC04 was higher than that of SC42, SA OS/390 immediately restarted the NFS 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 OS/390. We also wanted to measure the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 36. Failure of a TCP/IP stack

Purpose	Scope: TCP/IP stack Action: Unplanned outage
Expected behavior	SA OS/390 should try to restart the TCP/IP stack until the critical threshold is reached. If the critical threshold is reached, SA OS/390 should show a PROBLEM/HARDDOWN status and the TCP/IP stack will not be restarted. The NFS server should fail and SA OS/390 should restart it. SCS should fail and SA OS/390 should restart it on the LPAR where the enqueue replication server is running. SA OS/390 should try to restart the enqueue replication server on a different LPAR. The application server running on the LPAR where the failure occurs should fail and SA OS/390 should restart it. For the remote application server connected to the database server running on the LPAR where the failure occurs, running transactions should be rolled back and work processes should reconnect either to the same database server, or failover to the standby database server. For the application server running on the other LPAR, the failure should be transparent.

Table 36. Failure of a TCP/IP stack (continued)

Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: • The enqueue server running on SC42. • The enqueue replication server running on SC04. • The NFS server running on SC42.
Preparation	Log on to all the application servers. Create a workload on APPSRV11 running on SC04 and on APPSRV06 running on VMLINUX6 and connected to SC42. Create entries in the enqueue table.
Execution	Cancel the address space TCPIPA on SC42.
Verifications	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	SA OS/390 could restart the TCP/IP stack on SC42. The NFS server failed and SA OS/390 restarted it on SC04. SCS failed and SA OS/390 restarted it on SC04. SA OS/390 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. APPSRV10 running on SC42 failed. SA OS/390 restarted it. 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. 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 242, we logged on to all the application servers, created a workload on APPSRV11 (5 parallel tasks) as well as on APPSRV06 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

We simulated the failure by stopping TCPIPA on SC42 using the following command:

```
/P TCPIPA
```

Because the critical threshold was not reached, SA OS/390 immediately restarted TCPIPA on SC42:

```
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 application server APPSRV10 running on SC42. The ICLI servers, however, stayed up and reconnected as soon as TCP/IP was back.

SA OS/390 immediately restarted the NFS server on SC04.

SA OS/390 restarted SCS on the LPAR where the enqueue replication server was running, that is, SC04. The enqueue replication server stopped and SA OS/390 tried to restart it on SC42.

During our test, although SA OS/390 had successfully restarted 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 OS/390 policy, as last shutdown command for the resource RED_ES.

SA OS/390 immediately restarted the SAProuter on SC04.

The application server APPSRV10 running on SC42 went down and was immediately restarted by SA OS/390. All the sessions connected to this application server were, of course, lost and needed to be restarted.

The application server APPSRV06 running on VMLINUX6 lost the connection to the database server on SC42. The five running transactions received a DB2 SQL error 0 and were rolled back. The work processes were put in a reconnect status. The running sessions were lost and needed to be restarted by the users. Within seconds, the work processes reestablished the connection and left the reconnect status.

The transaction DB2 showed that the current DB host was still wtsc42a. We used the DB2 command -DIS THREAD(*) to check that all the threads are connected to SC42. Connection information for each work process can be found in the developer trace files dev_wx.

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 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 OS/390. We also wanted to measure the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 37. 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 OS/390 should restart the NFS server on another LPAR. SA OS/390 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 application server connected to the database server running on the failing LPAR, running transactions should be rolled back and work processes should failover to the standby database server. For the application server running on the other LPAR, the failure should be transparent.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: • The enqueue server running on SC42. • The enqueue replication server running on SC04. • The NFS server running on SC42.

Table 37. Failure of the LPAR where the ES and NFS servers are running (continued)

Preparation	Log on to all the application servers. Create a workload on APPSRV11 running on SC04 and on APPSRV06 running on VMLINUX6 and connected to the database server on SC42. Create entries in the enqueue table.
Execution	System reset at the HMC for SC42.
Verifications	Check if the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Look for error messages in the enqueue log file, in the developer traces dev_disp and dev_wx, and in the system log (SM21).
Results	ARM restarted the failing DB2 subsystem D7X1 on SC04 with the option RESTART(LIGHT). The DB2 subsystem went down after successful startup. SA OS/390 restarted the NFS server on SC04. SA OS/390 restarted 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 242, we logged on to all the application servers, created a workload on APPSRV11 (5 parallel tasks) as well as on APPSRV06 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

We simulated the failure by doing a system reset at the HMC.

We used the NetView command INGLIST */*/SC42 to display the status of the resources on SC42. They all appeared with a status INHIBITED/SYSGONE. 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 OS/390 restarted the NFS server on SC04.

SA OS/390 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 OS/390 would have restarted the enqueue replication server on that LPAR.

The application server APPSRV06 running on VMLINUX6 lost the connection to the database server on SC42. The five running transactions received a DB2 SQL error 0 and were rolled back. The work processes were put in a reconnect status. The running sessions were lost and needed to be restarted. The work processes did a failover to the standby database server, reestablished the connection and left the reconnect status.

The transaction DB2 showed that the current DB host was now wtsc04a, as shown in 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.

```
Settings:
Primary DB host                          wtsc42a
Standby DB host                          wtsc04a
Present DB host                          wtsc04a

Operation:
Operation completed successfully.
New DB host                              wtsc04a
```

For the application server APPSRV11 running on SC04, the failure is transparent—the workload is still running (SM66) and the lock entries that we generated are still in the enqueue table (SM12). The developer trace dev_disp shows that the dispatcher lost its connection with the message server 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  ThVBHdlMsgDown: 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  ThVBHdlMsgUp: 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: recv (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).
  - *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.

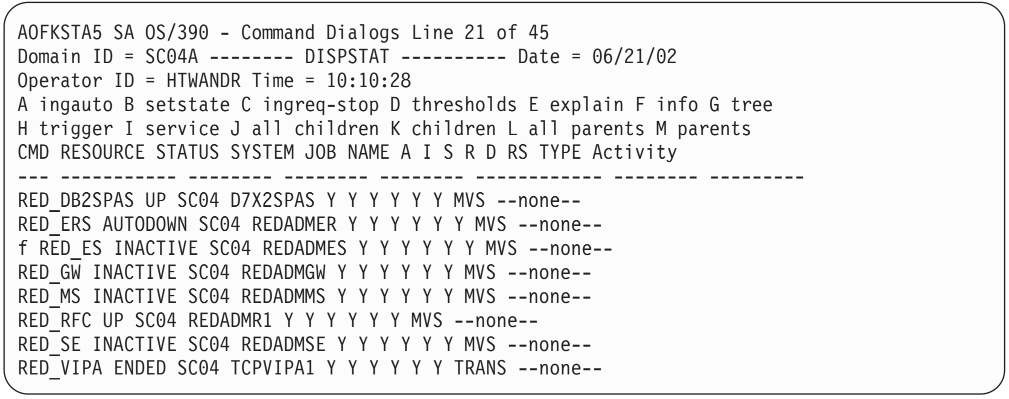
Problem determination in SA z/OS

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

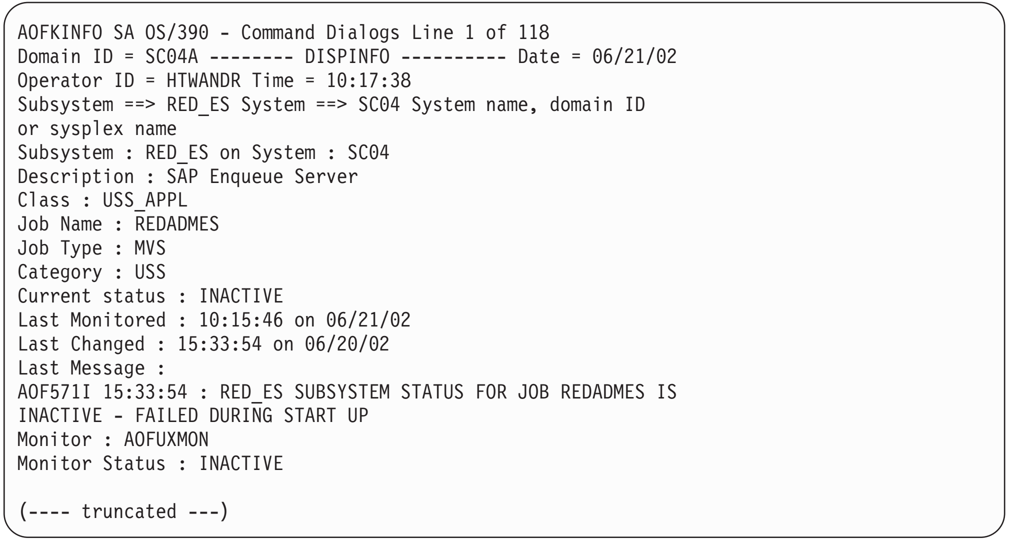
SDF or NMC: The first indication of an unusual situation is often the dynamic display of SDF or NMC. This display shows the status of the resource in question. You can use the help function to learn more about the meaning of the status color of each resource. You can also use the EXPLAIN command on the 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_DB2SPAS UP SC04 D7X2SPAS Y Y Y Y Y Y MVS --none--
RED_ERS AUTODOWN SC04 REDADMER Y Y Y Y Y Y MVS --none--
f RED_ES INACTIVE SC04 REDADMES Y Y Y Y Y Y MVS --none--
RED_GW INACTIVE SC04 REDADMGW Y Y Y Y Y Y MVS --none--
RED_MS INACTIVE SC04 REDADMMS Y Y Y Y Y Y MVS --none--
RED_RFC UP SC04 REDADMR1 Y Y Y Y Y Y MVS --none--
RED_SE INACTIVE SC04 REDADMSE Y Y Y Y Y Y MVS --none--
RED_VIPA ENDED SC04 TCPVIPA1 Y Y Y Y Y Y TRANS --none--
```

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
REDICLI6 APL SC04 SATISFACTORY AVAILABLE AVAILABLE
REDICLI7 APL SC04 SATISFACTORY AVAILABLE AVAILABLE
REDICLI8 APL SC04 SATISFACTORY AVAILABLE AVAILABLE
REDICLI9 APL SC04 SATISFACTORY AVAILABLE AVAILABLE
RESOLVER APL SC04 SATISFACTORY AVAILABLE AVAILABLE
RMF APL SC04 SATISFACTORY AVAILABLE AVAILABLE
```

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 268).

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 "fancy lights" on SDF or NMC.

Typically this situation occurs because there is a lock in the system, which can have various causes. In this section, we describe these causes and show how you can determine where the problem lies:

- A pending vote
  - Use the INGVOTE command to look for open votes.
- Missing supporting applications
  - Check the relationships of the failing application. Are there any unresolved dependencies?
- Pending excludes or avoids against groups
  - Use the INGGROUP command or the SANCHK REXX to find excludes and avoids
- Auto flags in the 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 268).
    - 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
    ```

    .
  - Check the home directory and the instance work directory for core files or CEEDUMP files indicating an abnormal termination of a UNIX process.

    Such files are also written if a DLL was not found due to an incorrect LIBPATH environment variable, or a module could not be loaded because of a missing STEPLIB definition.
- **SAP enqueue server, message server, gateway or syslog collector does not come up**

Problem determination in this case is similar to the application server case.

- Check messages in the startsap_EM00.*.log file. This file contains the output of the startup command invoked by SA.
- Check the SAP development traces in the work directory of Central Services. List the files in chronological order to see which ones have been written last.

In our configuration, they are located in the directory

```
/usr/sap/RED/EM00/work
```

- For the enqueue server, browse the 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:

```
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, (<*MVSNFS*> stands for the jobname of the NFS server):

```
F <MVSNFS>,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 <MVSNFS>,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*.

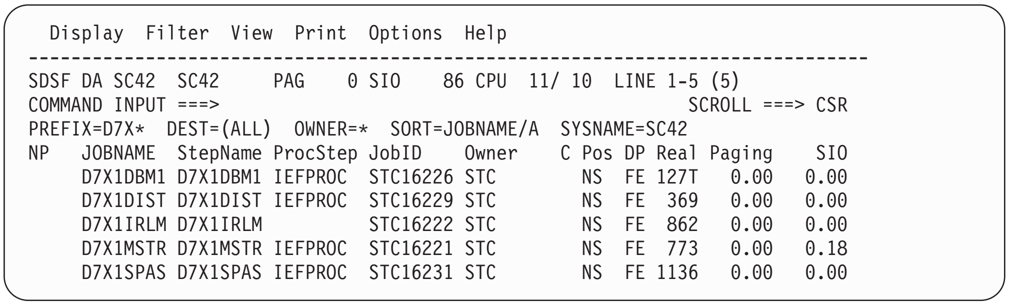
Check that DB2 is running

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 268).



```
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF DA SC42  SC42     PAG    0 SIO    86 CPU  11/ 10  LINE 1-5 (5)
COMMAND INPUT ===>                                            SCROLL ===> CSR
PREFIX=D7X*  DEST=(ALL)  OWNER=*  SORT=JOBNAME/A  SYSNAME=SC42
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     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 73. Results of SDSF DA command

Check the SAP database connections

- Use the DB2 Display Thread command to show the connections to DB2 from the SAP application server on USS, or the ICLI server for remote application servers. The following is the command we used:

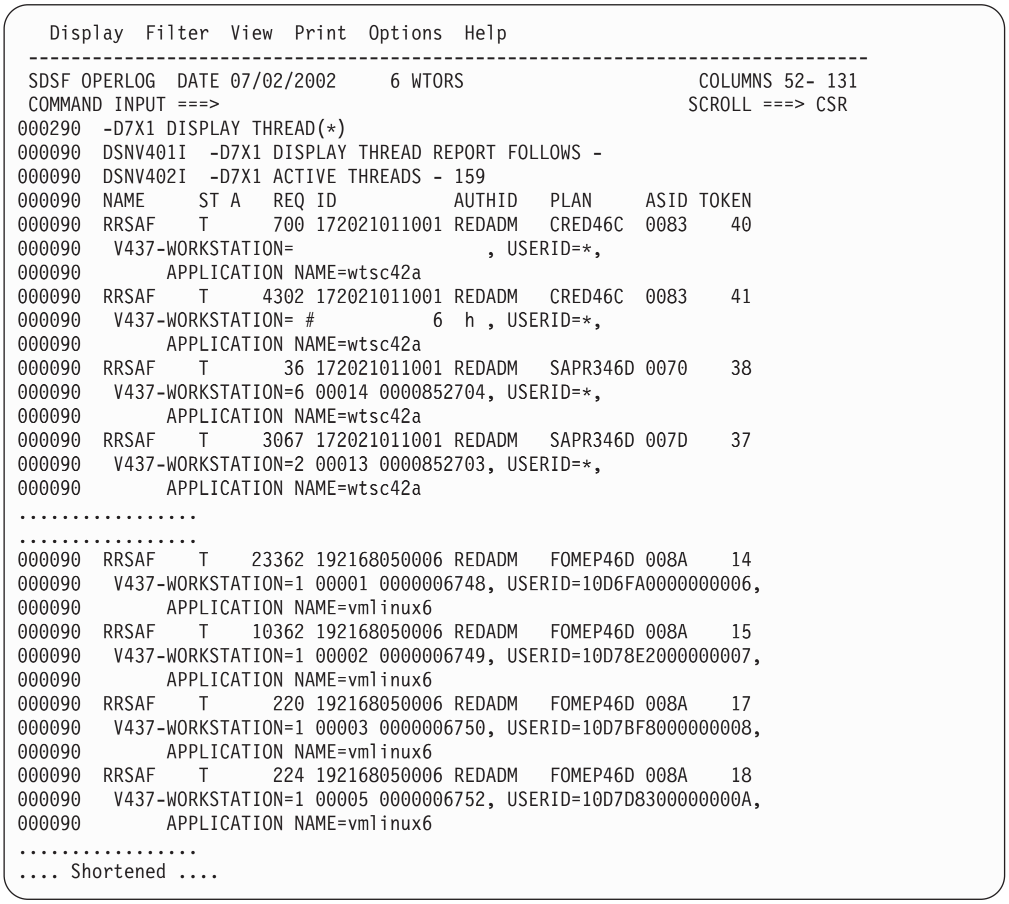
```
-D7X1 DISPLAY THREAD(*)
```

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).

For remote application servers (Linux6 is a remote application server in our configuration), you can check the ICLI server message file for messages indicating that the ICLI client on the application server tried to connect to the ICLI server.

If these messages are present, then look for messages indicating why the ICLI server could not connect with DB2. In our configuration, the ICLI message files are located in the directory /usr/sap/RED/icli/icli6.

If there are no DB2 connections from the USS application servers, or the remote application servers haven't tried to connect to the ICLI servers, refer to "Where to check for application problems" on page 272.



```
   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=wtsc42a
000090  RRSAF    T     4302 172021011001 REDADM   CRED46C  0083    41
000090   V437-WORKSTATION= #           6  h , USERID=*,
000090       APPLICATION NAME=wtsc42a
000090  RRSAF    T       36 172021011001 REDADM   SAPR346D 0070    38
000090   V437-WORKSTATION=6 00014 0000852704, USERID=*,
000090       APPLICATION NAME=wtsc42a
000090  RRSAF    T     3067 172021011001 REDADM   SAPR346D 007D    37
000090   V437-WORKSTATION=2 00013 0000852703, USERID=*,
000090       APPLICATION NAME=wtsc42a
.................
.................
000090  RRSAF    T    23362 192168050006 REDADM   FOMEP46D 008A    14
000090   V437-WORKSTATION=1 00001 0000006748, USERID=10D6FA0000000006,
000090       APPLICATION NAME=vmlinux6
000090  RRSAF    T    10362 192168050006 REDADM   FOMEP46D 008A    15
000090   V437-WORKSTATION=1 00002 0000006749, USERID=10D78E2000000007,
000090       APPLICATION NAME=vmlinux6
000090  RRSAF    T      220 192168050006 REDADM   FOMEP46D 008A    17
000090   V437-WORKSTATION=1 00003 0000006750, USERID=10D7BF8000000008,
000090       APPLICATION NAME=vmlinux6
000090  RRSAF    T      224 192168050006 REDADM   FOMEP46D 008A    18
000090   V437-WORKSTATION=1 00005 0000006752, USERID=10D7D8300000000A,
000090       APPLICATION NAME=vmlinux6
.................
.... Shortened ....
```

Figure 74. 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 38. 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 38. High availability test scenarios (continued)

High availability scenario	Effect
Coupling Facility takeover	No impact
Channel path (CHPID) failure	No impact
ICLI failure	No impact / SQL 0000
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

Chapter 14. Verification and problem determination on Linux on System z

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 39 on page 280 and Table 40 on page 283 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 39. Planned outages

Scenario	Action	Command	Expected Result
Normal operation	Precondition: All groups offline		
	Start an SAP system and related components (SAProuter)	`chrg -o online -s "Name like 'XI_%'"`	ROUTER, ENQ and D95 groups start on lnxsapg. ENQREP and D96 groups start on lnxsaph. D97 group starts on lnxsapi.
	Stop SAP system EP0	`chrg -o offline -s "Name like 'XI_%_EP0%'"`	ENQ, ENQREP, D95, D96, and D97 groups stop.
	Start SAP system EP0	`chrg -o online -s "Name like 'XI_%_EP0%'"`	ENQ and D95 groups start on lnxsapg. ENQREP and D96 groups start on lnxsaph. D97 group starts on lnxsapi.
	Stop entire SAP	`chrg -o offline -s "Name like 'XI_%'"`	All groups stop.

Table 39. 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_EP0_ENQREP`
		`chrg -o offline XI_J2EE_EP0_ENQREP`
			ENQREP groups stop

		`chrg -o online XI_ABAP_EP0_ENQREP`
		`chrg -o online XI_J2EE_EP0_ENQREP`
			ENQREP groups start on lnxsapg

		`chrg -o offline XI_ABAP_EP0_ENQ`
		`chrg -o offline XI_J2EE_EP0_ENQ`
			ENQ groups stop, but MS, ES, and IP stay online because of relation from ERS andAS. ENQ group status is Pending Offline.

		`chrg -o online XI_ABAP_EP0_ENQ`
		`chrg -o online XI_J2EE_EP0_ENQ`
			Offline resources of ABAP ENQ groups restart on lnxsaph.
			Note: This is *not* the way to move the ENQ groups! You need to initiate a failover by stopping ES outside SA control.

Table 39. 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).	`rgreq -o move -n lnxsaph XI_ABAP_LOP_ENQ`	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 *cannot* recreate the enqueue table data. Finally, the ERS starts on lnxsaph.
			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 *cannot* recreate the enqueue table data. **Note:** This is *not* the way to move the ENQ groups! You need to initiate a failover by stopping ES outside System Automation control.
		`stoprsrc -s "Name= 'XI_ABAP_EP0_ENQ_ES'" IBM.Application`	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).	`chrg -o offline XI_J2EE_EP0_lnxsapg_D95`
		`chrg -o offline XI_ABAP_EP0_lnxsapg_D95`
			D95 groups stop

		`chrg -o online XI_ABAP_EP0_lnxsapg_D95`
		`chrg -o online XI_J2EE_EP0_lnxsapg_D95`
			D95 groups restart on lnxsapg.

	Stop and restart z/OS LPAR, DB2, or ICLI server[1] in order to apply maintenance.	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. **Note:** SAP currently does not support an SAP failover mechanism for J2EE application servers.	

Table 40. Unplanned outages

Scenario	Simulation Action/Command	Expected Result
Precondition: ROUTER, ENQ, and D95 groups online on lnxsapg. ENQREP and D96 online on lnxsaph. D97 online on lnxsapi.		
Failure of the ABAP enqueue server	lnxsapg: `killall -9 es.sapEP0_ASCS00`	ABAP ENQ group stops and restarts on lnxsaph. ABAP ERS terminates. ABAP ENQREP group stops and restarts on lnxsapg. See also “Testing an unplanned outage of an ABAP SCS” on page 284.
Failure of the enqueue replication server.	lnxsapg: `killall -9 ers.sapEP0_ASCS00`	ABAP ENQREP group stops and restarts on lnxsapg.
Failure of the message server	lnxsaph: `killall -9 ms.sapEP0_ASCS00`	ABAP MS restarts on lnxsaph.
Failure of an ABAP or add-in application server	lnxsapg: `killall -9 dw.sapEP0_D95`	D95_AS restarts on lnxsapg.
Failure of the J2EE enqueue server	lnxsapg: `killall -9 es.sapEP0_SCS01`	J2EE ENQ group stops and restarts on lnxsaph J2EE ERS terminates J2EE ENQREP group stops and restarts on lnxsapg See also “Testing an unplanned outage of a J2EE SCS” on page 284.
Failure of the J2EE enqueue replication server	lnxsapg: `killall -9 ers.sapEP0_SCS01`	J2EE ENQREP group stops and restarts on lnxsapg
Failure of the J2EE message server	lnxsaph: `killall -9 ms.sapEP0_SCS01`	J2EE MS restarts on lnxsaph
Failure of a J2EE-only application server	lnxsapg: `killall -9 jc.sapEP0_J95`	J95 AS restarts on lnxsapg
Failure of the node where ES is running	lnxsaph: `reboot`	softdog kills lnxsaph because IP is a critical resource. ENQ groups start on lnxsapg. ERS terminates. ENQREP groups stop and restart on lnxsapi. D97 group starts on lnxsaph as soon as lnxsaph is back in the cluster.
Failure of the node where ERS is running.	lnxsapi: `reboot`	lnxsapi reboots (no critical resource online there). ENQREP group starts immediately on lnxsaph. D97 group starts on lnxsapi as soon as lnxsapi is back in the cluster.

Table 40. Unplanned outages (continued)

Scenario	Simulation Action/Command	Expected Result
Failure of an ICLI server[1], DB2 member on z/OS. 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 J2EE application servers.	

Testing an unplanned outage of an ABAP SCS

Note: This test does not apply to a J2EE-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 242. 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 245.

Testing an unplanned outage of a J2EE SCS

Check which machines your J2EE 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 J2EE enqueue server process via:

```
killall -9 es.sapEP0_SCS01
```

SA MP should then restart the J2EE enqueue server on the machine where the J2EE replication server is active. Run both 'enqt' commands again and check that the EnqId and enqueue table entries are the same as before the J2EE SCS failover. If so, you have verified that the enqueue table replication is working properly.

Part 6. 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:

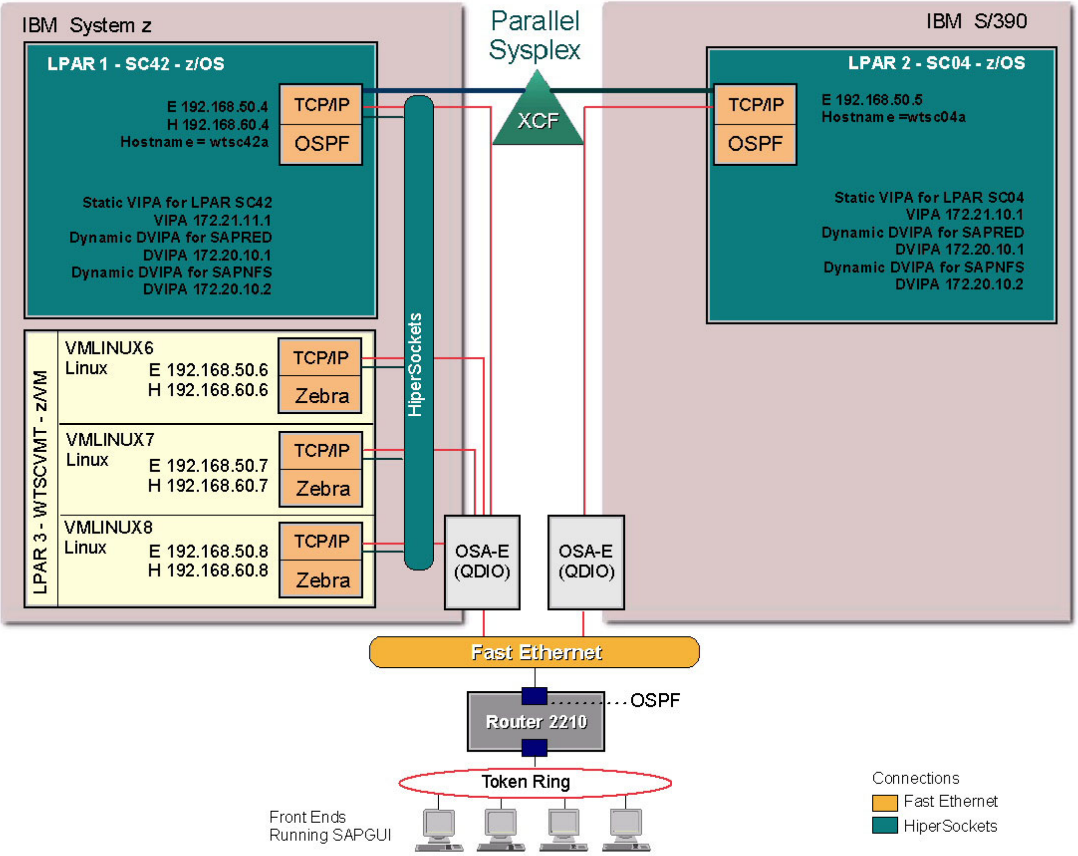


Figure 75. Networking configuration for the high availability solution for SAP

In our z/OS test environment, DB2 V7 was running. We therefore had to run the ICLI client/server package as the remote SQL interface. All descriptions in this chapter therefore contain the ICLI server as a resource managed by SA z/OS. With SAP NetWeaver '04 and DB2 V8, the ICLI client/server is no longer supported. Instead, DB2's DDF address space is supported on z/OS and DB2 Connect on remote application servers. Since the DDF address space is a standard DB2 for z/OS address space, you will no longer need specific ICLI server definitions.

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:

- Static virtual IP address (VIPA) definitions for:
  - ICLI servers

    Note: Defining a static VIPA is also necessary when DB2′s DDF address space is used.
- 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(UDS) ENTRYPOINT(BPXTUINT)
  NETWORK DOMAINNAME(AF_UNIX)
          DOMAINNUMBER(1)
          MAXSOCKETS(1000)
          TYPE(UDS)

  FILESYSTYPE TYPE(INET)
          ENTRYPOINT(EZBPFINI)
  NETWORK DOMAINNAME(AF_INET)
          DOMAINNUMBER(2)
          MAXSOCKETS(25000)
          TYPE(INET)
          ENTRYPOINT(EZBPFINI)

  FILESYSTYPE TYPE(NFS) ENTRYPOINT(GFSCINIT)
              ASNAME(NFSCLNT)
              PARM ('DISABLELLA(Y)')
/*******************************************************************/
```

ICLI server started task

This procedure is one of four used by both LPARs.

```
//REDICLI6 EXEC PGM=BPXBATCH,TIME=NOLIMIT,REGION=0M,
//        PARM='PGM /usr/sbin/fome46ds -PLAN FOMEP46D -LOGDIR /usr/sap/
//             RED/icli/icli6 -PORT 5006'
//STDENV   DD PATH='/usr/sap/RED/icli/iclienv'
//STEPLIB  DD DISP=SHR,DSN=DB7X7.SDSNLOAD
//STDERR   DD PATH='/usr/sap/RED/icli/icli6/redicli6.&SYSNAME..err',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRWXU)
//STDOUT   DD PATH='/usr/sap/RED/icli/icli6/redicli6.&SYSNAME..out',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRWXU)
//SYSUDUMP DD SYSOUT=*
//SYSMDUMP DD SYSOUT=*
```

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 RIPV2
PATHMTUDISCOVERY
SYSPLEXRouting
DYNAMICXCF 192.168.40.4 255.255.255.0 2
IGNORERedirect
REASSEMBLytimeout   15 ; In seconds
STOPONclawerror
TTL    60             ; In seconds, but actually Hop count
SACONFIG  COMMUNITY MVSsub1
ENABLED
AGENT 161
; 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 TCPSENDBFRSIZE 16K TCPRCVBUFRSIZE 16K SENDGARBAGE FALSE
;
; UDPCONFIG: Provides settings for the UDP layer of TCP/IP
;
UDPCONFIG UNRESTRICTLOWPORTS
;
---------------------------------------------------------------------
-
;
; Reserve low ports for servers
;
TCPCONFIG        RESTRICTLOWPORTS
UDPCONFIG        RESTRICTLOWPORTS
;
;
---------------------------------------------------------------------
--
;
; AUTOLOG the following servers
AUTOLOG 5
FTPD JOBNAME FTPDA   ; FTP Server
PMAPA                ; Portmap Server
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 NCPROUT             ; NCPROUTE Server
   750 TCP MVSKERB             ; Kerberos
   750 UDP MVSKERB             ; Kerberos
   751 TCP ADM@SRV             ; Kerberos Admin Server
   751 UDP ADM@SRV             ; Kerberos Admin Server
  2000 TCP IOASRV              ; OSA/SF Server
  2049 UDP MVSNFSSA            ; Our NFS Server
; --------------------------------------------------
;
; 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
```

```
172.21.11.1     STAVIPA1L
192.168.60.4    HIPERLEE
192.168.50.4    OSA2880LNK
;
---------------------------------------------------------------------
--
;
; IP routing information for the host.  All static IP routes should
; be added here.
;
GATEWAY
  192.168.50  =     OSA2880LNK   1500      0
  192.168.60  =     HIPERLEE     32768     0
;
DEFAULTNET 192.168.50.75 OSA2880LNK 1500        0
;
---------------------------------------------------------------------
--
; Turn off all tracing.  If tracing is to be used, change the
following
; line.  To trace the configuration component, for example, change
; the line to ITRACE ON CONFIG 1

ITRACE OFF
;
;
---------------------------------------------------------------------
-
; The ASSORTEDPARMS NOFWD will prevent the forwarding of IP packets
; between different networks.  If NOFWD is not specified, IP packets
; will be forwarded between networks when this host is a gateway.
;
; Even though RESTRICTLOWPORTS was specified on TCPCONFIG and
UDPCONFIG,
; ASSORTEDPARMS default would have been to reset RESTRICTLOWPORTS to
off
; So it is respecified here.
; If the TCPCONFIG and UDPCONFIG followed ASSORTEDPARMS,
RESTRICTLOWPORT
; would not have to be done twice.

ASSORTEDPARMS
; NOFWD
  RESTRICTLOWPORTS
ENDASSORTEDPARMS
; Start all the defined devices.
;
START OSA2880
START IUTIQDEE
```

OMPROUTE started task - SC42:

```
//OMPROUTA PROC
//OMPROUTE EXEC PGM=BPXBATCH,REGION=4096K,TIME=NOLIMIT,
// PARM='PGM /usr/lpp/tcpip/sbin/omproute'
//*      'ENVAR("_CEE_ENVFILE=DD:STDENV")/')
//*
//* PARM=('POSIX(ON)',
//*       'ENVAR("_CEE_ENVFILE=DD:STDENV")/-t1')
//*
//*        Provide environment variables to run with the
//*        desired stack and configuration.  As an example,
//*        the file specified by STDENV could have these
//*        three lines in it:
//*
//*           RESOLVER_CONFIG=//'SYS1.TCPPARMS(TCPDATA2)'
//*           OMPROUTE_FILE=/u/usernnn/config.tcpcs2
//*           OMPROUTE_DEBUG_FILE=/tmp/logs/omproute.debug
//*
//*        For information on the above environment variables,
//*        refer to the IP CONFIGURATION GUIDE.
//*
//STDENV   DD DSN=TCPIPA.&SYSNAME..OMPROUTA.ENVVARS,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
```

ENVVARS - SC42:

```
RESOLVER_CONFIG=//'TCPIPA.SC42.TCPPARMS(TCPDATA)'
OMPROUTE_FILE=//'TCPIPA.SC42.TCPPARMS(OMPROUTA)'
OMPROUTE_DEBUG_FILE=/tmp/omprouta.debug
```

Define the OMPROUTA procedure to RACF. At a TSO command prompt, enter the following commands:

```
rdefine started omprouta.* 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.

```
Area    area_number=0.0.0.0
             stub_Area=no
             Authentication_type=none;
ROUTERID=172.21.11.1;
OSPF_Interface IP_Address=172.21.11.1
               Subnet_mask=255.255.255.0
               Router_Priority=1
               Name=STAVIPA1L
               MTU=1500;
OSPF_Interface IP_Address=192.168.60.4
               Subnet_mask=255.255.255.0
               Router_Priority=1
               Name=HIPERLEE
               MTU=16384;
OSPF_Interface IP_Address=192.168.50.4
               Subnet_mask=255.255.255.0
               Router_Priority=0
               Name=OSA2880LNK
               MTU=1500;
Ospf_interface IP_Address=192.168.40.4
               Name=DYNXCF
               Router_Priority=1
               Subnet_mask=255.255.255.0;
OSPF_Interface IP_Address=172.20.10.0
               Subnet_Mask=255.255.255.0
               Router_Priority=1
               Name=VRANGE;
AS_Boundary_routing
  Import_RIP_Routes=YES
  Import_Direct_Routes=no
  Import_Static_Routes=no;
```

Linux on System z network settings for the test setup

In this section, we describe the network settings for Linux on System z.

Zebra setup - OSPF

```
! -*- ospf -*-
!
! ospfd.conf sample configuration file
!
hostname ospfd
password zebra
!enable password please-set-at-here
!
!router zebra
!network 192.168.1.0/24 area 0
interface hsi1
 ip ospf cost 5
 ip ospf priority 5
interface eth2
 ip ospf cost 10
 ip ospf priority 0
router ospf
 network 192.168.50.0/24 area 0
 network 192.168.60.0/24 area 0
!
log stdout
```

Zebra setup - Zebra

```
! -*- zebra -*-
!
! zebra.conf sample configuration file
!
hostname Router
password zebra
enable password zebra
!
! Interface's description.
!
!interface lo
interface eth2
interface hsi1

!
! Static default route sample.
!
!ip route 0.0.0.0/0 203.181.89.241
!

log file /var/log/zebra.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
```

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,NFSPRFX=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=512M,TIME=1440,
//         PARM='POSIX(ON) ALL31(ON) / -p TCPIPA -c 172.20.10.2'
//*
//GFSAMAIN EXEC PGM=&MODULE,REGION=0M,TIME=1440,COND=(4,LT),
//         PARM=(,
//         'ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIPA")/')
//SYSTCPD  DD   DISP=SHR,DSN=&TCPIP..;&SYSNAME..TCPPARMS(&TCPDATA.)
//STEPLIB  DD   DISP=SHR,DSN=&SYSNFS..NFSLIB
//SYSPRINT DD   SYSOUT=*
//OUTPUT   DD   SYSOUT=*
//SYSERR   DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//NFSATTR  DD   DISP=SHR,DSN=&NFSPRFX..SAPRED.PARMS(ATTRIB)
//EXPORTS  DD   DISP=SHR,DSN=&NFSPRFX..SAPRED.PARMS(EXPORTS)
//NFSLOG1  DD   DISP=SHR,DSN=&NFSPRFX..SAPRED.SERVER.LOG1
//NFSLOG2  DD   DISP=SHR,DSN=&NFSPRFX..SAPRED.SERVER.LOG2
//FHDBASE  DD   DISP=SHR,DSN=&NFSPRFX..SAPRED.FHDBASE1
//FHDBASE2 DD   DISP=SHR,DSN=&NFSPRFX..SAPRED.FHDBASE2
//NFSXLAT  DD   DISP=SHR,DSN=&NFSPRFX..SAPRED.XLAT
```

NFS export file

Following is our export file content:

```
########################################################################
#                                                                      #
#  OS/390 Network File System Server EXPORTS                           #
#                                                                      #
########################################################################
#
/hfs/sapmnt/RED/profile    -access=vmlinux6
/hfs/sapmnt/RED/global     -access=vmlinux6
/hfs/sapmnt/RED/AIX/exe    -access=vmlinux6:vmlinux7:vmlinux8:erprisc2
/hfs/sapmnt/RED/Linux/exe  -access=vmlinux6
/hfs/sap/trans              -access=vmlinux6
```

NFS attribute file

Following is our attribute file content:

```
space(100,10), blks
norlse
recfm(fb), blksize(0), lrecl(80)
dsorg(ps)
dsntype(pds)
dir(25)
keys(64,0)
recordsize(512,4K)
nonspanned
shareoptions(3,3)
attrtimeout(120), readtimeout(90), writetimeout(30)
text
CRLF
blankstrip
mapleaddot
maplower
retrieve
nofastfilesize
setownerroot
executebitoff
xlat(oemvs311)
nofileextmap
sidefile(OS390NFS.SAPRED.NFS.MAPPING)
security(saf,exports,saf)
pcnfsd
leadswitch
mintimeout(1)
nomaxtimeout
logout(604800)                  # 60 * 60 * 24 * 7
nfstasks(8,16,8)
restimeout(48,0)
cachewindow(112)
hfs(/hfs)
logicalcache(16M)
bufhigh(32M)
percentsteal(20)
readaheadmax(16K)
maxrdforszleft(32)
smf(none)
sfmax(20)
nochecklist
fn_delimiter(,)
```

Mount commands on Linux /etc/fstab

```
sapnfs:/hfs/sapmnt/RED/global,text,xlat(oemvs311)  /sapmnt/RED/global nfs
intr,rsize=8192,wsize=8192
sapnfs:/hfs/sapmnt/RED/profile,text,xlat(oemvs311) /sapmnt/RED/profile nfs
intr,rsize=8192,wsize=8192
sapnfs:/hfs/sapmnt/RED/Linux/exe,text,xlat(oemvs311)  /sapmnt/RED/Linux/exe
nfs intr,rsize=8192,wsize=8192
sapnfs:/hfs/sap/trans,text,xlat(oemvs311)          /usr/sap/trans nfs
intr,rsize=8192,wsize=8192
```

Appendix C. ARM policy

This appendix shows the Automated Restart Manager (ARM) policy.

ARM policy JCL

```
//ARMPOL   JOB (999,POK),'SAPRES6',CLASS=A,MSGCLASS=T,
//         NOTIFY=&SYSUID,REGION=4M
/*JOBPARM SYSAFF=SC42
//*-----------------------------------------------------------------*//
//S1      EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *

   DATA TYPE(ARM)

     DEFINE POLICY NAME(ARM01) REPLACE(YES)

     RESTART_GROUP(DB7XGRP)
       TARGET_SYSTEM(SC42,SC04)
       ELEMENT(DB7XUD7X1)
         RESTART_ATTEMPTS(3,120)
         RESTART_TIMEOUT(60)
         READY_TIMEOUT(900)
         TERMTYPE(ALLTERM)
         RESTART_METHOD(ELEMTERM,PERSIST)
         RESTART_METHOD(SYSTERM,STC,'-D7X1 STA DB2,LIGHT(YES)')
       ELEMENT(DB7XUD7X2)
         RESTART_ATTEMPTS(3,120)
         RESTART_TIMEOUT(60)
         READY_TIMEOUT(900)
         TERMTYPE(ALLTERM)
         RESTART_METHOD(ELEMTERM,PERSIST)
         RESTART_METHOD(SYSTERM,STC,'-D7X2 STA DB2,LIGHT(YES)')

/*
```

ARM policy

Appendix D. Basic setup of Tivoli NetView and Tivoli System Automation for z/OS

This appendix contains the following:

- Definitions for the AOFSAP SDF screen
- The sample REXX exec SANCHK

Status Display Facility definition

This section contains the sample SDF panel AOFSAP, the modified SDF tree definition member AOFTSC04, and the modified SDF start screen AOFPSYST.

At a minimum, you might want to use the AOFSAP screen as a base for your own screen developments.

AOFPSYST

```
/***START OF COPYRIGHT NOTICE******************************************/00010000
/*                                                                    */00020000
/* Proprietary Statement:                                             */00030000
/*                                                                    */00040000
/*      5685-151 5655-137                                             */00050000
/*      Licensed Materials - Property of IBM                          */00060000
/*      (C) COPYRIGHT IBM CORP. 1990, 2000   All Rights Reserved.     */00070000
/*                                                                    */00080000
/*      US Government Users Restricted Rights -                       */00090000
/*      Use, duplication or disclosure restricted by                  */00100000
/*      GSA ADP Schedule Contract with IBM Corp.                      */00110000
/*                                                                    */00120000
/*      STATUS= HKYS100                                               */00130000
/*                                                                    */00140000
/**END OF COPYRIGHT NOTICE*********************************************/00150000
/*********************************************************************/ 00160000
/* Change Code Vrsn Date    Who  Description                         */ 00170000
/* ----------- ---- ------- ---  ----------------------------------- */ 00180000
/* $L0=FEATURE,SA21,06JUL00,MIK: Rework for V2R1                     */ 00190000
/*                                                                   */ 00200000
/* ***************************************************************** */ 00210000
/*                                                                   */ 00220000
/* Main system monitoring panel                                      */ 00230000
/*                                                                   */ 00240000
/*   - Repeat definitions for each system added                      */ 00250000
/*     - Remember to put each system on a different line             */ 00260000
/*                                                                   */ 00270000
/*   - Works with definitions from AOFPXXXX and AOFTXXXX             */ 00280000
/*                                                                   */ 00290000
P(SYSTEM,24,80)                                                         00300000
TF(01,02,10,WHITE,NORMAL)                                               00310000
TT(SYSTEM)                                                              00320000
TF(01,23,58,WHITE,NORMAL)                                               00330000
TT(SA OS/390 - SUPPORT SYSTEMS)                                         00340000
/*                                                                   */ 00350000
/* First column is system name                                       */ 00360000
/*                                                                   */ 00370000
TF(03,05,10,T,U)                                                        00380000
TT(System)                                                              00390000
SF(SC04,05,05,10,N,,SC04)                                               00400000
ST(SC04)                                                                00410000
SF(SC42,07,05,10,N,,SC42)                                               00420000
ST(SC42)                                                                00430000
/*                                                                   */ 00500000
/* Second column is the worst subsystem                              */ 00510000
/*                                                                   */ 00520000
TF(03,14,24,T,U)                                                        00530000
TT(Subsystems)                                                          00540000
SF(SC04.APPLIC,05,14,24,N,,SC04,Q1)                                     00550000
SF(SC42.APPLIC,07,14,24,N,,SC42,Q1)                                     00560000
/*                                                                   */ 00600000
/* Third column is the worst WTOR                                    */ 00610000
/*                                                                   */ 00620000
TF(03,27,34,T,U)                                                        00630000
TT(WTORs)                                                               00640000
SF(SC04.WTOR,05,27,34,N,,SC04,1)                                        00650000
SF(SC42.WTOR,07,27,34,N,,SC42,1)                                        00660000
/*                                                                   */ 00700000
/* Fourth column is the worst gateway                                */ 00710000
/*                                                                   */ 00720000
```

```
TF(03,37,45,T,U)                                                        00730000
TT(Gateways)                                                            00740000
SF(SC04.GATEWAY,05,37,45,N,,SC04,1)                                     00750000
SF(SC42.GATEWAY,07,37,45,N,,SC42,1)                                     00760000
/*                                                                   */ 00800000
/* Fifth column is a set of C I D O indicators for each of the       */ 00810000
/* product automation packages.                                      */ 00820000
/*                                                                   */ 00830000
/* - Each system requires 8 entries here...                          */ 00840000
/*                                                                   */ 00850000
TF(03,48,55,T,U)                                                        00860000
TT(Products)                                                            00870000
/*                                                                   */ 00880000
/* Indicators for SC04                                               */ 00890000
/*                                                                   */ 00900000
SF(SC04.CICS,05,48,48,N,,SC04C,)                                        00910000
ST(C)                                                                   00920000
SF(SC04.IMS,05,50,50,N,,SC04I,)                                         00930000
ST(I)                                                                   00940000
SF(SC04.DB2,05,52,52,N,,SC04D,)                                         00950000
ST(D)                                                                   00960000
SF(SC04.OPCERR,05,54,54,N,,SC04O,)                                      00970000
ST(O)                                                                   00980000
/*                                                                   */ 00990000
/* Indicators for SC42                                               */ 01000000
/*                                                                   */ 01010000
SF(SC42.CICS,07,48,48,N,,SC42C,)                                        01020000
ST(C)                                                                   01030000
SF(SC42.IMS,07,50,50,N,,SC42I,)                                         01040000
ST(I)                                                                   01050000
SF(SC42.DB2,07,52,52,N,,SC42D,)                                         01060000
ST(D)                                                                   01070000
SF(SC42.OPCERR,07,54,54,N,,SC42O,)                                      01080000
ST(O)                                                                   01090000
/*                                                                   */ 01430000
/* Sixth column is a set of P V M B T U indicators for each of the   */ 01440000
/* product automation packages.                                      */ 01450000
/*                                                                   */ 01460000
/* - Each system requires 12 entries here...                         */ 01470000
/*                                                                   */ 01480000
TF(03,58,68,T,U)                                                        01490000
TT(System)                                                              01500000
/*                                                                   */ 01510000
/* Indicators for SC04                                               */ 01520000
/*                                                                   */ 01530000
SF(SC04.SPOOL,05,58,58,N,,SC04,)                                        01540000
ST(P)                                                                   01550000
SF(SC04.MVSCOMP,05,60,60,N,,SC04,)                                      01560000
ST(V)                                                                   01570000
SF(SC04.MESSAGES,05,62,62,N,,SC04O,)                                    01580000
ST(M)                                                                   01590000
SF(SC04.BATCH,05,64,64,N,,SC04O,)                                       01600000
ST(B)                                                                   01610000
SF(SC04.ONLINE,05,66,66,N,,SC04O,)                                      01620000
ST(T)                                                                   01630000
SF(SC04.TSOUSERS,05,68,68,N,,SC04O,)                                    01640000
ST(U)                                                                   01650000
/*                                                                   */ 01660000
/* Indicators for SC42                                               */ 01670000
/*                                                                   */ 01680000
```

```
SF(SC42.SPOOL,07,58,58,N,,SC42,)                                        01690000
ST(P)                                                                   01700000
SF(SC42.MVSCOMP,07,60,60,N,,SC42,)                                      01710000
ST(V)                                                                   01720000
SF(SC42.MESSAGES,07,62,62,N,,SC42O,)                                    01730000
ST(M)                                                                   01740000
SF(SC42.BATCH,07,64,64,N,,SC42O,)                                       01750000
ST(B)                                                                   01760000
SF(SC42.ONLINE,07,66,66,N,,SC42O,)                                      01770000
ST(T)                                                                   01780000
SF(SC42.TSOUSERS,07,68,68,N,,SC42O,)                                    01790000
ST(U)                                                                   01800000

/* ---------- The following 2 lines are for SAP HA ------------------*/ 01810000

SF(SC42.SAP,15,24,74,N,,AOFSAP,)                                        01820004

ST(S A P  High Availability)                                            01830003

/*                                                                   */ 01840002
/*                                                                   */ 02260000
/* PFKey Definitions...                                              */ 02270000
/*                                                                   */ 02280000
TF(24,01,47,T,NORMAL)                                                   02290000
TT(1=HELP 2=DETAIL 3=RETURN   6=ROLL   8=NEXT SCR)                      02300000
TF(24,48,79,T,NORMAL)                                                   02310000
TT(  10=LEFT 11=RIGHT 12=TOP)                                           02320000
EP                                                                      02330000
```

AOFSAP

```
/* **START OF COPYRIGHT NOTICE************************************** */ 00010030
/*                                                                   */ 00020030
/* Proprietary Statement:                                            */ 00030030
/*      5655-137                                                     */ 00050030
/*      Licensed Materials - Property of IBM                         */ 00060030
/*      (C) COPYRIGHT IBM CORP. 1990, 2000   All Rights Reserved.    */ 00070030
/*                                                                   */ 00080030
/*      US Government Users Restricted Rights -                      */ 00090030
/*      Use, duplication or disclosure restricted by                 */ 00100030
/*      GSA ADP Schedule Contract with IBM Corp.                     */ 00110030
/*                                                                   */ 00120030
/*      STATUS= HKYS100                                              */ 00130030
/*                                                                   */ 00140030
/* *END OF COPYRIGHT NOTICE***************************************** */ 00150030
/* ***************************************************************** */ 00160030
/*                                                                   */ 00230030
P(AOFSAP,24,80,SYSTEM,SYSTEM, , , )                                     00240030
TF(01,12,60,Y,R)                                                        00250047
TT(            S A P  High Availability  )                              00260046
/*                                                                   */ 00261030
TF(03,01,21,P,NORMAL)                                                   00270044
TT(Local Applications)                                                  00280044
/*                                                                   */ 00290030
TF(03,40,70,P,NORMAL)                                                   00291044
TT(Moving Applications)                                                 00292033
/*                                                                   */ 00300030
TF(04,01,06,T,NORMAL)                                                   00350048
TT(SC04)                                                                00360030
TF(04,15,20,T,NORMAL)                                                   00370048
TT(SC42)                                                                00380030
/*                                                                   */ 00390030
TF(04,40,45,T,NORMAL)                                                   00400048
TT(SC04)                                                                00410032
TF(04,54,59,T,NORMAL)                                                   00411048
TT(SC42)                                                                00412032
/*                                                                   */ 00413032
TF(05,01,30,T,NORMAL)                                                   00420048
TT(-----------------------------)                                       00430033
TF(05,39,69,T,NORMAL)                                                   00431048
TT(-----------------------------)                                       00432033
/*                                                                   */ 00480030
SF(SC04.RED_DB2MSTR,06,01,13,N, ,)                                      00490044
ST(RED_DB2MSTR)                                                         00500035
SF(SC42.RED_DB2MSTR,06,15,27,N, ,)                                      00510044
ST(RED_DB2MSTR)                                                         00520035
SF(SC04.RED_DB2DBM1,07,01,13,N, ,)                                      00530044
ST(RED_DB2DBM1)                                                         00540036
SF(SC42.RED_DB2DBM1,07,15,27,N, ,)                                      00550044
ST(RED_DB2DBM1)                                                         00560036
SF(SC04.RED_DB2IRLM,08,01,13,N, ,)                                      00570044
ST(RED_DB2IRLM)                                                         00580036
SF(SC42.RED_DB2IRLM,08,15,27,N, ,)                                      00590044
ST(RED_DB2IRLM)                                                         00600036
SF(SC04.RED_DB2DIST,09,01,13,N, ,)                                      00610044
ST(RED_DB2DIST)                                                         00620036
SF(SC42.RED_DB2DIST,09,15,27,N, ,)                                      00630044
ST(RED_DB2DIST)                                                         00640036
SF(SC04.RED_DB2SPAS,10,01,13,N, ,)                                      00650044
ST(RED_DB2SPAS)                                                         00660036
```

```
SF(SC42.RED_DB2SPAS,10,15,27,N, ,)                                      00670044
ST(RED_DB2SPAS)                                                         00680036
SF(SC04.RED_RFC,12,01,13,N, ,)                                          00690044
ST(RED_RFC)                                                             00700037
SF(SC42.RED_RFC,12,15,27,N, ,)                                          00710044
ST(RED_RFC)                                                             00720037
SF(SC04.REDICLI6,13,01,13,N, ,)                                         00730044
ST(REDICLI6)                                                            00740037
SF(SC42.REDICLI6,13,15,27,N, ,)                                         00750044
ST(REDICLI6)                                                            00760037
SF(SC04.REDICLI7,14,01,13,N, ,)                                         00770044
ST(REDICLI7)                                                            00780037
SF(SC42.REDICLI7,14,15,27,N, ,)                                         00790044
ST(REDICLI7)                                                            00800037
SF(SC04.REDICLI8,15,01,13,N, ,)                                         00810044
ST(REDICLI8)                                                            00820037
SF(SC42.REDICLI8,15,15,27,N, ,)                                         00830044
ST(REDICLI8)                                                            00840037
SF(SC04.REDICLI9,16,01,13,N, ,)                                         00850044
ST(REDICLI9)                                                            00860037
SF(SC42.REDICLI9,16,15,27,N, ,)                                         00870044
ST(REDICLI9)                                                            00880037
SF(SC42.APPSRV10,18,15,27,N, ,)                                         00910044
ST(APPSRV10)                                                            00920038
SF(SC04.APPSRV11,18,01,13,N, ,)                                         00930049
ST(APPSRV11)                                                            00940038
SF(SC04.SAP_OSCOL,19,01,13,N, ,)                                        01014049
ST(SAP_OSCOL)                                                           01015038
SF(SC42.SAP_OSCOL,19,15,27,N, ,)                                        01016049
ST(SAP_OSCOL)                                                           01017038
/*                                                                   */ 01020030
/*                                                                   */ 01021030
SF(SC04.MVSNFSSA,06,40,52,N, ,)                                         01030544
ST(MVSNFSSA  )                                                          01030630
SF(SC04.SAP_RTVIPA,08,40,52,N, ,)                                       01030944
ST(SAP_RTVIPA)                                                          01031030
SF(SC04.SAP_ROUTER,09,40,52,N, ,)                                       01031144
ST(SAP_ROUTER)                                                          01031244
SF(SC04.RED_VIPA,11,40,52,N, ,)                                         01031344
ST(RED_VIPA  )                                                          01031440
SF(SC04.RED_ES,12,40,52,N, ,)                                           01031544
ST(RED_ES    )                                                          01031640
SF(SC04.RED_MS,13,40,52,N, ,)                                           01031744
ST(RED_MS    )                                                          01031840
SF(SC04.RED_GW,14,40,52,N, ,)                                           01031944
ST(RED_GW    )                                                          01032040
SF(SC04.RED_CO,15,40,52,N, ,)                                           01032144
ST(RED_CO    )                                                          01032240
SF(SC04.RED_SE,16,40,52,N, ,)                                           01032344
ST(RED_SE    )                                                          01032440
SF(SC04.RED_ERS,17,40,52,N, ,)                                          01032544
ST(RED_ERS   )                                                          01032630
SF(SC04.APPSRV06,19,40,52,N, ,)                                         01032744
ST(APPSRV06  )                                                          01032830
SF(SC04.APPSRV07,20,40,52,N, ,)                                         01032944
ST(APPSRV07  )                                                          01033030
SF(SC04.APPSRV08,21,40,52,N, ,)                                         01033144
ST(APPSRV08  )                                                          01033230
/*                                                                   */ 01033330
/*                                                                   */ 01035930
```

```
SF(SC42.MVSNFSSA,06,54,66,N, ,)                                         01036044
ST(MVSNFSSA  )                                                          01036140
SF(SC42.SAP_RTVIPA,08,54,66,N, ,)                                       01036444
ST(SAP_RTVIPA)                                                          01036540
SF(SC42.SAP_ROUTER,09,54,66,N, ,)                                       01036644
ST(SAP_ROUTER)                                                          01036744
SF(SC42.RED_VIPA,11,54,66,N, ,)                                         01036844
ST(RED_VIPA  )                                                          01036940
SF(SC42.RED_ES,12,54,66,N, ,)                                           01037044
ST(RED_ES    )                                                          01037140
SF(SC42.RED_MS,13,54,66,N, ,)                                           01037244
ST(RED_MS    )                                                          01037340
SF(SC42.RED_GW,14,54,66,N, ,)                                           01037444
ST(RED_GW    )                                                          01037540
SF(SC42.RED_CO,15,54,66,N, ,)                                           01037644
ST(RED_CO    )                                                          01037740
SF(SC42.RED_SE,16,54,66,N, ,)                                           01037844
ST(RED_SE    )                                                          01037940
SF(SC42.RED_ERS,17,54,66,N, ,)                                          01038044
ST(RED_ERS   )                                                          01038140
SF(SC42.APPSRV06,19,54,66,N, ,)                                         01038244
ST(APPSRV06  )                                                          01038340
SF(SC42.APPSRV07,20,54,66,N, ,)                                         01038444
ST(APPSRV07  )                                                          01038540
SF(SC42.APPSRV08,21,54,66,N, ,)                                         01038644
ST(APPSRV08  )                                                          01038740
/*                                                                   */ 01038840
TF(24,01,49,T,NORMAL)                                                   01250037
TT(PF1=HELP 2=DETAIL 3=END        6=ROLL 7=UP 8=DN)                     01260030
TF(24,51,79,T,NORMAL)                                                   01270030
TT( 9=DEL  10=LF 11=RT 12=TOP)                                          01280030
PFK9('EVJEAB11 &SNODE,&ROOT.&COMPAPPL,&RV,&DATA')                       01290030
EP
```

AOFTSC04

```
/* **START OF COPYRIGHT NOTICE*************************************** */ 00010000
/*                                                                    */ 00020000
/* Proprietary Statement:                                             */ 00030000
/*                                                                    */ 00040000
/*       5655-137                                                     */ 00050000
/*       Licensed Materials - Property of IBM                         */ 00060000
/*       (C) COPYRIGHT IBM CORP. 1990, 2000   All Rights Reserved.    */ 00070000
/*                                                                    */ 00080000
/*       US Government Users Restricted Rights -                      */ 00090000
/*       Use, duplication or disclosure restricted by                 */ 00100000
/*       GSA ADP Schedule Contract with IBM Corp.                     */ 00110000
/*                                                                    */ 00120000
/*       STATUS= HKYS100                                              */ 00130000
/*                                                                    */ 00140000
/* *END OF COPYRIGHT NOTICE****************************************** */ 00150000
/* ****************************************************************** */ 00160000
/* Change-Activity:                                                   */ 00170000
/*                                                                    */ 00180000
/* Change Code Vers Date    Who       Description                     */ 00190000
/* ----------- ---- ------- --------- ------------------------------- */ 00200000
/*                                                                    */ 00210000
/* $L0=FEATURE,SA21,06JUL00,APC(MIK): Sample rework for V2R1          */ 00220000
/* ****************************************************************** */ 00230000
1 SC04                                                                   00430000
  2 SYSTEM                                                               00440000
    3 APPLIC                                                             00450000
      4 SUBSYS                                                           00460000
  2 WTOR                                                                 00470000
  2 SPOOL                                                                00480000
  2 GATEWAY                                                              00490000
  2 MVSCOMP                                                              00500000
  2 APG                                                                  00510000
    3 GROUPS                                                             00520000
/*                                                                    */ 00530000
/* ------------------------------------------------------------------ */ 00540000
/*                                                                    */ 00550000
/* The following subtree is required if the extended CICS product    */ 00560000
/* automation has been activated for the system.                      */ 00570000
/*                                                                    */ 00580000
  2 CICS                                                                 00590000
    3 CICSHLTH                                                           00600000
    3 CICSLMT                                                            00610000
    3 CICSAUTO                                                           00620000
    3 CICSMSG                                                            00630000
    3 CICSSTG                                                            00640000
      4 CICSSOS                                                          00650000
      4 CICSVIOL                                                         00660000
    3 CICSTIMR                                                           00670000
    3 CICSTRAN                                                           00680000
    3 VTAMACB                                                            00690000
/*                                                                    */ 00700000
/* ------------------------------------------------------------------ */ 00710000
/*                                                                    */ 00720000
/* The following subtree is required if the extended IMS product     */ 00730000
/* automation has been activated for the system.                      */ 00740000
/*                                                                    */ 00750000
  2 IMS                                                                  00760000
    3 IMSMSG                                                             00770000
    3 IMSARCH                                                            00780000
    3 IMSMSCL                                                            00790000
    3 IMSOLDS                                                           00800000
    3 IMSRECN                                                           00810000
    3 IMSTIMR                                                           00820000
    3 IMSTRAN                                                           00830000
    3 IMSSTRCT                                                          00840000
/*                                                                   */ 00850000
/* ----------------------------------------------------------------- */ 00860000
/*                                                                   */ 00870000
/* The following subtrees are required if the extended OPC product   */ 00880000
/* automation has been activated for the system.                     */ 00890000
/*                                                                   */ 00900000
  2 OPCERR                                                              00910000
  2 BATCH                                                               00920000
  2 TSOUSERS                                                            00930000
  2 SYSTEM                                                              00940000
    3 MESSAGES                                                          00950000
    3 IO                                                                00960000
      4 TAPES                                                           00970000
      4 ONLINE                                                          00980000
/*                                                                   */ 00990000
/* ----------------------------------------------------------------- */ 01000000
/*                                                                   */ 01010000
/* The following subtree is required if the extended DB2 product     */ 01020000
/* automation has been activated for the system.                     */ 01030000
/*                                                                   */ 01040000
  2 DB2                                                                 01050000
    3 DB2MSG                                                            01060000

/* ----------------------------------------------------------------- */ 01061001
/*                                                                   */ 01062001
/* The following subtree is required if the SAP HA is used           */ 01063008
/* All resource names needs to be customized for your environment.   */ 01064008
  2 SAP                                                                 01070001
    3 MVSNFSSA                                                          01080005
    3 SAP_ROUTER                                                        01090005
    3 SAP_RTVIPA                                                        01100005
    3 RED_ERS                                                           01110005
    3 RED_SE                                                            01130005
    3 RED_VIPA                                                          01140005
    3 RED_MS                                                            01150005
    3 RED_CO                                                            01160005
    3 RED_GW                                                            01170005
    3 RED_ES                                                            01180005
    3 APPSRV06                                                          01190005
    3 APPSRV07                                                          01200005
    3 APPSRV08                                                          01210005
    3 RED_DB2MSTR                                                       01220007
    3 RED_DB2DBM1                                                       01230007
    3 RED_DB2IRLM                                                       01240007
    3 RED_DB2DIST                                                       01250007
    3 RED_DB2SPAS                                                       01260007
    3 RED_RFC                                                           01270007
    3 REDICLI6                                                          01280007
    3 REDICLI7                                                          01281007
    3 REDICLI8                                                          01282007
    3 REDICLI9                                                          01283007
    3 APPSRV10                                                          01290007
    3 APPSRV11                                                          01291007
    3 SAP_ROUTER                                                        01300007
    3 SAP_RTVIPA                                                        01310007
    3 SAP_OSCOL                                                         01320007
```

Sample REXX procedure

SANCHK

This REXX procedure can be used to display and to clear EXCLUDEs and AVOIDs from Move Groups.

```
/* REXX SANCHK ------------------------------------------------------ */00001102
/*                                                                    */00001202
/* FUNCTION   : Display or CLEAR EXCLUDEs or AVOIDs from MOVE "groups */00001302
/*                                                                    */00001402
/*                                                                    */00001502
/*                          +---- DISPLAY ------+                     */00001602
/* SYNTAX     : sanchk -----+-------------------+-------------------- */00001702
/*                          +----  CLEAR  ------+                     */00001802
/*                                                                    */00001902
/* ------------------------------------------------------------------ */00002502
Trace 0                                                                 00020000
                                                                        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
  errdata.errcnt = strip(excl avoid)

  cnt = cnt + 1
  outline.cnt = '--------------------------------------------'
  cnt = cnt + 1
  outline.cnt = 'Group      = '||group
  cnt = cnt + 1
  outline.cnt = '  Excluded = '||excl
  cnt = cnt + 1
  outline.cnt = '  Avoided  = '||avoid

End i

If cnt = 0 Then Do
  If action = 'CLEAR' Then act = 'clear'
  Else act = 'display'
  cnt = cnt + 1
  outline.cnt = 'Nothing to '||act||' ...'
End
Else Do
  cnt = cnt + 1
  outline.cnt = '--------------------------------------------'
  cnt = cnt + 1
  outline.cnt = 'End of Sanity Check'
End

outline.0 = cnt
errgroup.0 = errcnt
errdata.0 = errcnt

Select
  When action = 'DISPLAY' Then Do
    "PIPE (STAGESEP | NAME DISPLAY)" ,
      "STEM outline. COLLECT" ,
      "| COLOR YELLOW" ,
      "| CONS ONLY"
  End
  When action = 'CLEAR' & errcnt = 0 Then Do
    "PIPE (STAGESEP | NAME DISPLAY)" ,
      "STEM outline. COLLECT" ,
      "| COLOR YELLOW" ,
      "| CONS ONLY"
  End
  When action = 'CLEAR' Then Do
    /* Issue processing message ... */
    Address NetVAsis ,
    "PIPE LIT /Processing CLEAR .../" ,
      "| COLOR RED" ,
      "| CONS ONLY"

    Do i = 1 to errgroup.0
      /* Issue processing message ... */
      Address NetVAsis ,
      "PIPE LIT \Processing CLEAR for "||errgroup.i||"\" ,
        "| COLOR RED" ,
        "| CONS ONLY"

      "PIPE (STAGESEP | NAME INGGROUP)" ,
        "NETV INGGROUP "||errgroup.i||",ACTION=INCLUDE,"|| ,
        "SYSTEMS=("||errdata.i||"),"|| ,
        "OUTMODE=LINE" ,
        "| CONS ONLY"
    End i

    /* Issue processing message ... */
    Address NetVAsis ,
    "PIPE LIT /Finished CLEAR processing/" ,
      "| COLOR RED" ,
      "| CONS ONLY"

  End
  Otherwise Nop
End

Exit
```

Appendix E. Detailed 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_v7.zip** at the time of this writing:

Table 41. Scripts for SA z/OS

Filename	Description
readme_v7	Supplementary and explanatory information for this version.
HABook_SA31Policy_xmit.bin	Sample SA z/OS V3.1 Policy Database in TSO XMIT format. Upload this file in binary format to a z/OS data set with the following attributes: • Space units: BLOCK • Primary quantity: 90 • Secondary quantity: 10 • Directory blocks: 0 • Record format: FB • Record length: 80 • Block size: 3120 Then receive this data set via XMIT and 'Receive from INDSN'.
SA23_SAPMSGUX	Sample Automation Table for SAP HA
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 local and remote application servers.
DEFAULT.PFL	Default profile for ASCS.
RED_ASCS00	Instance profile for ASCS.

Table 41. Scripts for SA z/OS (continued)

Filename	Description
start_rfcoscol	Sample shell script used to start rfcoscol.
startappsrv_v5	Sample shell script used to start a local or remote application server instance.
startsap_em00	Sample shell script used to start the components of ASCS.
stopappsrv_v5	Sample shell script used to stop a local or remote application server instance.
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.
640stopsapCodeForzOS.txt	The ’DB2 Performance data collector’ is automatically started by the SAP-provided 6.40 ’startsap’ script when running on z/OS. This file contains the sample code to automatically stop this collector when running the 6.40 SAP kernel on z/OS. The code must be added to the SAP-provided ’stopsap’ script.
startsap_jscs	Sample shell script used to start the components of J2EE SCS.
SAPe_to_ePolicyVSDraft10.xml	Sample end-to-end policy for the SAP EP0 system.

Script descriptions

Introduction

The scripts are for SAP 6.20 (or higher) kernels. With 6.20, the syntax of the SAP start and stop commands was changed compared to 4.6D. The 4.6D syntax

```
startsap_<hostname>_<instance number>_adm
```

has been replaced by a new startsap in the executable’s directory which takes two parameters. The same is true for the stopsap command. Therefore, the startappsrv_v5 and stopappsrv_v5 scripts are invoked with at least three [or four] 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.

Note: Initial 4.6D-based versions of the scripts used parameter <SysNr>. This parameter has been renamed to <instnr> because it is really the instance number of the SAP application server.

<instancedir>
is a directory required by SAP 6.20 for the startsap and stopsap commands if more than one instance is running on the host, and this is always the

case when ASCS and an application server are running, as is true in the HA solution. We run ASCS *and* an application server instance on the host. Therefore, this parameter is required. For example, if the SAP 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. It can also be set to SSH if the remote application. is controlled via SSH.

Note: The SA z/OS invocation (policy) must be adapted from a 4.6D based HA solution to provide the <instancedir> and then the <via> parameter. To use the 6.20 based script for a 4.6D HA solution, make the following two changes:

1. Provide a dummy value for the <instancedir> parameter or change the line of code that checks the number of input parameters to allow just 2 parameters,
2. Activate the 4.6D command syntax and comment out the 6.20 syntax.

To control a remote 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:

z/OS high availability scripts

```
@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
set RSDB_ICLILIBRARY=d:\usr\sap\RED\sys\exe\run\ibmiclic.dll

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.
set ICLI_TRUSTED_CONNECTIONS=1

d:\usr\sap\RED\sys\exe\run\R3trans -d
exit errorlevel
```

startappsrv_v5

This script is used to start a local or remote 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.

The line starting with rfcping= has to be edited to reflect the full path of the rfcping utility.

The remote execution must be set up to run without password prompt.

```
#!/bin/sh

if [[ $# -lt 3  || $# -gt 4 ]]; then
    echo "Usage:  $0  <Hostname> <InstNr> <InstanceDir> [SSH|REXEC]"
    exit
fi

ashost=$1
instancedir=$3
sysnr=$2
via=$4

# next line with fully qualified rfcping needs to be adapted to your env.
rfcping=/usr/sap/RED/rfc/rfcping
#4.6D syntax:
# command="cleanipc $sysnr remove; ./stopsap_${ashost}_${sysnr}; ./startsap_${ashost}_${sysnr}"
#6.20 syntax: if more than one instance is running, ${instancedir} as additional parameter is needed:
command="cleanipc $sysnr remove; stopsap r3 ${instancedir}; startsap r3 ${instancedir}"
commandfile="remote_startsap_${ashost}_${sysnr}.bat"
commandfile1="remote_checkdb_${ashost}_${sysnr}.bat"

# The following commands are for debugging purpose only
echo "Actual environment of the process ......"
echo "****************************************"
env
ulimit -a
echo "****************************************"

# Check then whether Database can be reached
dbconnect_test="R3trans -d"
if [ `hostname -s` = $ashost ]
then
    eval $dbconnect_test
    rc=$?
    if [[ $rc -ne 0 && $rc -ne 4 && $rc -ne 8 ]]; then
       echo "$_BPX_JOBNAME STARTUP FAILED (R3TRANS RC $rc)" > /dev/console
       echo ">>> $ashost $sysnr STARTUP FAILED (R3TRANS RC $rc)"
       exit 0
    fi
    echo "R3TRANS RC: $rc (DATABASE active)"
else
    # Check first whether ashost can be reached
    ping $ashost
    rc=$?
    if [ rc -gt 0 ]; then
        echo "$_BPX_JOBNAME STARTUP FAILED (PING)" > /dev/console
        echo ">>> $ashost $sysnr STARTUP FAILED (PING)"
        exit 0
    fi
```

Figure 76. startappsrv script (Part 1 of 3)

```
    case $via in
        SSH)
            echo "USING SSH METHOD"
# protocol version 1 (not recommended) and WINDOWS (one command file):
#           SSHOUT=`/bin/ssh -1 $ashost $commandfile1 | tr -d '\r' | tr '\' '/'`
# protocol version 2 (recommended):
            SSHOUT=`/bin/ssh -2 $ashost $dbconnect_test | tr -d '\r' | tr '\' '/'`
            if [[ ${#SSHOUT} -gt 0 ]]; then
               RC0=$(echo $SSHOUT|sed -n 's/.*R3trans finished (\(.*\))\./\1/p')
               # echo "R3trans return code is $RC0"
               if [[ -z $RC0 ]]; then
                  echo "$_BPX_JOBNAME STARTUP FAILED (SSH returned $SSHOUT)" > /dev/console
                  echo ">>> $ashost $sysnr STARTUP FAILED (SSH returned $SSHOUT)"
                  exit 0               fi
               if [[ $RC0 -ne 0 && $RC0 -ne 4 && $RC0 -ne 8 ]]; then
                  echo "$_BPX_JOBNAME STARTUP FAILED (R3TRANS RC $RC0)" > /dev/console
                  echo ">>> $ashost $sysnr STARTUP FAILED (R3TRANS RC $RC0)"
                  exit 0
               fi
            else
               echo "$_BPX_JOBNAME STARTUP FAILED (R3TRANS FAILED)" > /dev/console
               echo ">>> $ashost $sysnr STARTUP FAILED (R3TRANS FAILED)"
               exit 0
            fi
            echo "R3TRANS RC: $RC0 (DATABASE active)"
            ;;
        RBAT)
            echo "USING RBAT METHOD"
            /bin/rexec $ashost "$commandfile1" ;;
        REXEC)
            echo "USING REXEC METHOD"
            REXECOUT=`/bin/rexec $ashost $dbconnect_test`
            if [[ ${#REXECOUT} -gt 0 ]]; then
               RC1=$(echo $REXECOUT|sed -n 's/.*R3trans finished (\(.*\))\./\1/p')
               # echo "R3trans return code is $RC1"
               if [[ -z $RC1 ]]; then
                  echo "$_BPX_JOBNAME STARTUP FAILED (REXEC returned $REXECOUT)" > /dev/console
                  echo ">>> $ashost $sysnr STARTUP FAILED (REXEC returned $REXECOUT)"
                  exit 0
               fi
               if [[ $RC1 -ne 0 && $RC1 -ne 4 && $RC1 -ne 8 ]]; then
                  echo "$_BPX_JOBNAME STARTUP FAILED (R3TRANS RC $RC1)" > /dev/console
                  echo ">>> $ashost $sysnr STARTUP FAILED (R3TRANS RC $RC1)"
                  exit 0
               fi
            else
               echo "$_BPX_JOBNAME STARTUP FAILED (R3TRANS FAILED)" > /dev/console
               echo ">>> $ashost $sysnr STARTUP FAILED (R3TRANS FAILED)"
               exit 0
            fi
            echo "R3TRANS RC: $RC1 (DATABASE active)"
            ;;
        *)
            echo "NO (valid) METHOD SPECIFIED. USING DEFAULT REXEC METHOD"
            REXECOUT=`/bin/rexec $ashost $dbconnect_test`
            if [[ ${#REXECOUT} -gt 0 ]]; then
               RC1=$(echo $REXECOUT|sed -n 's/.*R3trans finished (\(.*\))\./\1/p')
               # echo "R3trans return code is $RC1"
               if [[ -z $RC1 ]]; then
                  echo "$_BPX_JOBNAME STARTUP FAILED (REXEC returned $REXECOUT)" > /dev/console
                  echo ">>> $ashost $sysnr STARTUP FAILED (REXEC returned $REXECOUT)"
                  exit 0
               fi
               if [[ $RC1 -ne 0 && $RC1 -ne 4 && $RC1 -ne 8 ]]; then
                  echo "$_BPX_JOBNAME STARTUP FAILED (R3TRANS RC $RC1)" > /dev/console
                  echo ">>> $ashost $sysnr STARTUP FAILED (R3TRANS RC $RC1)"
                  exit 0
               fi
```

Figure 76. startappsrv script (Part 2 of 3)

|

```
            else
               echo "$_BPX_JOBNAME STARTUP FAILED (R3TRANS FAILED)" > /dev/console
               echo ">>> $ashost $sysnr STARTUP FAILED (R3TRANS FAILED)"
               exit 0
            fi
            echo "R3TRANS RC: $RC1 (DATABASE active)"
            ;;
    esac
fi
# Check then whether AppServer is working
# On timeout (rc=3) don't restart, indicate startup failure
$rfcping ping_cnt=1 ashost=$ashost sysnr=$sysnr
rc=$?
if [ rc -eq 0 ]; then
    echo "$_BPX_JOBNAME ACTIVE" > /dev/console
    echo ">>> $ashost $sysnr ACTIVE"
    exit 0
fi
if [ rc -gt 2 ]; then
    echo "$_BPX_JOBNAME STARTUP FAILED (TIMEOUT)" > /dev/console
    echo ">>> $ashost $sysnr STARTUP FAILED (TIMEOUT)"
    exit 0
fi

# Start remote AppServer
if [ `hostname -s` = $ashost ]
then
    eval $command
else
    case $via in
        SSH)
# protocol version 1 (not recommended) and WINDOWS (one command file):
#           /bin/ssh -1 $ashost "$commandfile" &;;
# protocol version 2 (recommended):
            /bin/ssh -2 $ashost "$command" &;;
        RBAT)
            /bin/rexec $ashost "$commandfile" &;;
        REXEC)
            /bin/rexec $ashost "$command" &;;
        *)
            echo "NO (valid) METHOD SPECIFIED. USING DEFAULT REXEC METHOD"
            /bin/rexec $ashost "$command" &;;
    esac
fi
if [ $? -ne 0 ]; then
    echo "$_BPX_JOBNAME STARTUP FAILED (START)" > /dev/console
    echo ">>> $ashost $sysnr STARTUP FAILED (START)"
    exit 0
fi

# Verify that startup was successful, retry for 120 seconds
retries=0
while [ retries -lt 12 ]
do
    /bin/sleep 10
    $rfcping ping_cnt=1 ashost=$ashost sysnr=$sysnr
    if [ $? -eq 0 ]; then
        echo "$_BPX_JOBNAME ACTIVE" > /dev/console
        echo ">>> $ashost $sysnr ACTIVE"
        exit 0
    fi
    retries=`expr $retries + 1`
done

# Obviously startup has failed
echo "$_BPX_JOBNAME STARTUP FAILED" > /dev/console
echo ">>> $ashost $sysnr STARTUP FAILED"
```

Figure 76. startappsrv script (Part 3 of 3)

stopappsrv_v5

This script is used to stop a local or remote 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.

The remote execution must be set up to run without password prompt.

```
#!/bin/sh

if [[ $# -lt 3  || $# -gt 4 ]]; then
    echo "Usage:  $0  <Hostname> <InstNr> <InstanceDir> [SSH|REXEC]"
    exit
fi

ashost=$1
instancedir=$3
sysnr=$2
via=$4

#4.6D syntax: command="./stopsap_${ashost}_${sysnr}"
#6.20 syntax: if more than one instance is running, ${instancedir} as additional parameter is needed:
command="stopsap r3 ${instancedir}"
commandfile="remote_stopsap_${ashost}_${sysnr}.bat"

# Stop remote AppServer
if [ `hostname -s` = $ashost ]
then
    eval $command
    rc=$?
else
    # Check first whether ashost can be reached
    ping $ashost
    rc=$?
    if [ rc -gt 0 ]; then
        echo "$_BPX_JOBNAME STOP FAILED (PING)" > /dev/console
        echo ">>> $ashost $sysnr STOP FAILED (PING)"
        exit 0
    fi

    case $via in
        SSH)
            echo "USING SSH METHOD"
# protocol version 1 (not recommended) and WINDOWS (one command file):
#           /bin/ssh -1 $ashost "$commandfile" &
# protocol version 2 (recommended):
            /bin/ssh -2 $ashost "$command" &
            rc=$?;;
        RBAT)
            echo "USING RBAT METHOD"
            /bin/rexec $ashost "$commandfile" &
            rc=$?;;
        REXEC)
            echo "USING REXEC METHOD"
            /bin/rexec $ashost "$command" &
            rc=$?;;
        *)
            echo "NO (valid) METHOD SPECIFIED. USING DEFAULT REXEC METHOD"
            /bin/rexec $ashost "$command" &
            rc=$?;;
    esac
fi
if [[ $rc -ne 0 && $rc -ne 6 ]]; then
    echo "$_BPX_JOBNAME STOP FAILED (RC $rc)" > /dev/console
    echo ">>> $ashost $sysnr STOP FAILED (RC $rc)"
    exit 0
fi
```

Figure 77. stopappsrv script

checkappsrv_v4

This script is used to start the monitor for a local or remote application server instance. It takes the host name and the instance number of the application server as parameters.

The line starting with rfcping= has to be edited to reflect the full path of the rfcping utility. In addition, the *cd* command has to be adapted.

```
#!/bin/sh

if [ $# -lt 2 ]; then
        echo "Usage:  $0  <Hostname> <InstNr>"
        exit
fi

ashost=$1
sysnr=$2
rfcping=/usr/sap/RED/rfc/rfcping

# next 2 lines with fully qualified rfcping need to be adapted to your env.
cd /usr/sap/RED/rfc
/bin/ln -sf $rfcping ./rfcping_${ashost}_${sysnr}
./rfcping_${ashost}_${sysnr} ping_cnt=-1 ping_interval=10 ashost=$ashost sysnr=$sysnr > /dev/null
echo "$_BPX_JOBNAME ENDED" > /dev/console
```

Figure 78. checkappsrv script

startsap_em00

This shell script is used to start the components of ASCS. The component is identified with one of the following abbreviations: MS, ES, ERS, CO, SE, GW. Furthermore, the CHECK option performs a health check on the enqueue server.

The lines starting with DIR_INSTANCE=, DIR_EXECUTABLE=, and PROFILE= , and the values of the variables _ES, _MS, _CO, _SE, _GW and _ERS, must be adapted to your environment.

```
#!/bin/sh
DIR_INSTANCE=/usr/sap/RED/ASCS00
DIR_EXECUTABLE=/usr/sap/RED/SYS/exe/run
PROFILE=/usr/sap/RED/SYS/profile/RED_ASCS00

_ES=es.sapRED_ASCS00
_MS=ms.sapRED_ASCS00
_CO=co.sapRED_ASCS00
_SE=se.sapRED_ASCS00
_GW=gw.sapRED_ASCS00
_ERS=ers.sapRED_ASCS00

cd $DIR_INSTANCE/work

case "$1" in

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

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

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

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

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

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

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

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

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

Figure 79. startsap_em00 script

startsap_jscs

This shell script is similar to startsap_em00. It is used to start the components of the J2EE SCS. The component is identified with one of the abbreviations MS, ES, and ERS. Furthermore, the CHECK option performs a health check on the enqueue server.

The lines starting with DIR_INSTANCE=, DIR_EXECUTABLE=, and PROFILE= , and the values of the variables _ES, _MS, and _ERS, must be adapted to your environment.

```
#!/bin/sh
DIR_INSTANCE=/usr/sap/RED/SCS01
DIR_EXECUTABLE=/usr/sap/RED/SYS/exe/runU
# If an ABAP SCS (ASCS) is active in your SAP system which is using the ASCII character set,
# and if also a J2EE SCS (SCS) is active in your SAP system which can only use the UNICODE
# character set, then the SCS resource executables are located in the special
#
# .../SYS/exe/runU directory (U for UNICODE) instead of the normal  .../SYS/exe/run.
#
# This is normally the case for an 'Add-In' J2EE SCS, when the ABAP part is using ASCII
# character set and the J2EE part is using UNICODE.

PROFILE=/usr/sap/RED/SYS/profile/RED_SCS01

_ES=es.sapRED_SCS01
_MS=ms.sapRED_SCS01
_ERS=ers.sapRED_SCS01

cd $DIR_INSTANCE/work

case "$1" in

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

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

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

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

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

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

Figure 80. startsap_jscs script

Appendix F. Detailed description of the sample Tivoli System Automation for Multiplatforms high availability policy for SAP

This appendix explains why the SA MP HA policy for SAP (version 4.2) described in this document was defined as it is. It gives some background information and should help you to understand what is happening and why.

Refer to Chapter 11, “Customizing the Tivoli System Automation for Multiplatforms base component,” on page 199 for a general discussion on implementing this policy.

The ABAP SAP Central Services group (ASCS)

The ASCS group (for example XI_ABAP_EP0_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. 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 resources (GW, CO, or SE) fails and cannot be restarted, this does not trigger a group failover, and the resource stays down.

Note: With 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 (see “Setting up the policy (mksap)” on page 341).

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 (XI_ABAP_EP0_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 329. If the ASCS is run in ASCII mode, the ERS is also automatically created to run in ASCII mode.

The J2EE SAP Central Services group

The J2EE SCS group, for example XI_J2EE_EP0_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 J2EE 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 J2EE SCS and the NFS server.

The ENQREP group for the J2EE SCS

The J2EE ENQREP group (XI_J2EE_EP0_ENQREP) contains one mandatory floating resource, the ERS. The relationships of this group and its members to the J2EE SCS group are described in “Interaction between ES and ERS” on page 329.

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.

With the sample HA policy for SAP (version 4), the support for ABAP-only application servers has been enhanced to also support J2EE-only and ‘add-in’ application servers. Although an ‘add-in’ application server is physically one instance, we separated it into its two logical parts, the ABAP application server and the J2EE application server. In other words, within an SA MP domain, one ‘add-in’ application server instance is automated as two logical application server instances, an ABAP application server instance and a J2EE application server instance. However, there is a close relationship between these two logical application servers. The J2EE instance is always started after the ABAP instance. This 'StartAfter' relationship guarantees that starting the J2EE instance automatically triggers the prior start of the ABAP instance. On the other hand, stopping the J2EE application server does not stop any of the ‘add-in’ server processes. It only stops the monitoring Java program ‘GetWebPage’, which does a primitive health check of J2EE application servers as the rfcping program does for ABAP application servers.

Note: 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. 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 10, “Customizing Tivoli System Automation for z/OS,” on page 161. If you use the sample HA policy for SAP (version 4.2) 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 (XI_ABAP_SYS_ROUTER) contains two floating resources, the SAProuter (SAPROUTER) and 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 J2EE systems. The SAP Web Dispatcher group (XI_ABAP_SYS_WEBDISP) also contains two floating resources, the SAP Web Dispatcher (SAPWEBDISP) and 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.

The SAPOSCOL group

The SAPOSCOL program is the SAP utility that collects operating system information for an application server. It is started by default when the first application server is started on a machine via the startsap script. It runs only once on a machine and must be stopped manually. The sample policy does not cover this group by default. However, if the SAP executables are not local and if the NFS server that provides access to the SAP executables is automated in the same SA MP domain as the SAP application server, shutdown of the NFS server will hang while SAPOSCOL is running. To avoid this, SAPOSCOL can only be started and stopped by SA MP.

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 J2EE SCS instances with enqueue server replication enabled.

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.

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.

Creating the resources

Basically we have 4 different types of SAP resources to manage:

- resources of the ASCS (ES, ERS, MS, GW, SE, CO) and SCS (ES, ERS, MS)
- application servers (ABAP-only, J2EE-only, and 'add-in')
- SAPSID-independent resources (SAPROUTER, SAPWEBDISP, SAPOSCOL)
- service IP addresses of ES (ABAP and/or J2EE) and SAPROUTER/ SAPWEBDISP

The SAP processes

All the SAP programs run as operating system processes, so we can control them by SA MP resources of the class IBM.Application. Resources of class IBM.Application require some attributes and some others can be optionally set; defaults are used otherwise. The required attributes are:

- Name
- Start command that SA MP can execute whenever the resource is to be brought online
- Stop command that SA MP can execute whenever the resource is to be taken offline
- Monitor command that SA MP can execute to retrieve the current status
- User name under which the process will execute

Examples of optional attributes are:

- Resource type (floating in the domain or fixed on one node)
- Monitoring period
- List of allowed nodes
- Whether the resource is critical or not

Resources of the ABAP SCS and J2EE SCS

Consider the mksrc command for the ES, as an example:

```
mkrsrc IBM.Application
  Name="XI_ABAP_EP0_ENQ_ES"
  ResourceType=1
  NodeNameList="{lnxsapg,lnxsaph,lnxsapi}"
  ...
```

The name is XI_ABAP_EP0_ENQ_ES because ES is part of the ENQ group, which is part of SAPSID EP0, and the resource prefix in this case is XI_ABAP (see the naming conventions under “Tivoli System Automation for Multiplatforms” on page 120). It is a resource of type 1 (floating) because – for HA reasons – we want the ES to be able to float between different nodes in the domain. We allow the ES to run on nodes lnxsapg, lnxsaph and lnxsapi.

We manage all the resources of the ASCS (ES, ERS, MS, GW, SE, CO) and J2EE SCS (ES, ERS, MS) with one script (sapctrl_em). sapctrl_em uses parameters to start, stop and monitor (check) the different resources. One parameter specifies which resource of the respective SCS instance is to be managed. Another parameter is needed to make the script independent of the SAPSID. The invocation syntax of sapctrl_em is:

```
./sapctrl_em <resource-prefix> <sapsid> <instance-dir> <host name> <exeU flag>
    {MS|ES|ERS|CO|SE|GW} {START|STOP|CHECK}
```

For details on sapctrl_em, see “Managing the ABAP or J2EE SCS with sapctrl_em” on page 347. Currently, we have not implemented any ’health checking’ for ES and MS. Instead, we use PID monitoring. The relevant part of the SA MP mkrsrc command (for the ES) would be

```
 mkrsrc ...
  StartCommand="<install dir>/sapctrl_em XI_ABAP EP0 ASCS00 no_enq_host 0 ES START"
  StopCommand="<install dir>/sapctrl_em XI_ABAP EP0 ASCS00 no_enq_host 0 ES STOP"
  MonitorCommand="<install dir>/sapctrl_em XI_ABAP EP0 ASCS00 no_enq_host ES CHECK"
  MonitorCommandPeriod=120
  MonitorCommandTimeout=10  ...
```

We therefore want SA MP to call our monitoring script every 120 seconds and give it 10 seconds to return the status. The 2 minute check interval is OK, because it is really for checking only. The script which is called to start the resource notices immediately when the process stops, and issues the SA MP action routine refreshOpState, which triggers monitoring immediately. This monitor run will find the changed resource status and SA MP can initiate appropriate action.

We still need to specify the user name under which the ES should run. This is the <sapsid>adm userid:

```
mkrsrc ...
  UserName="ep0adm"
  ...
```

One more attribute is required. By default SA MP expects any start or stop command to return within the given timeout, which is 5 seconds by default. The

ES, however, is, from an HA perspective, ideally a never-ending-process. We need to tell SA MP not to wait at all for the start command to terminate and start monitoring immediately instead. This is done with the following attribute:

```
mkrsrc ...
  RunCommandsSync=0
```

The ES is not critical in terms of SA MP. Therefore the default of attribute ProtectionMode=0 for class IBM.Application is fine.

The preceding is true for all resources of the respective SCS, so you can substitute ES with MS and then have the mkrsrc command for the MS.

Application servers

Considering the application servers, we select as an example the one on lnxsapg with instance number 95:

```
mkrsrc IBM.Application
  Name="XI_ABAP_EP0_lnxsapg_D95_AS"
  ResourceType=0
  NodeNameList="{lnxsapg}"
  ...
```

The name is `XI_ABAP_EP0_lnxsapg_D95_AS` because AS has instance directory 95, runs on lnxsapg, is part of SAPSID EP0, and the resource prefix in this case is XI_ABAP (see the naming conventions under “Tivoli System Automation for Multiplatforms” on page 120). We have a resource of type 0 (fixed) because we generally do not move application servers in case of a failure. We assume that there are at least two application server instances defined for one SAP system and that they are operational. This allows users of a failing application server to switch to a still-working instance immediately. To have an idle ‘standby’ machine for application server failover makes little sense because it would take a long time to move the application server instance and would also be a waste of server resources.

We have one script (sapctrl_as) to manage all types of application servers: ABAP-only, J2EE-only, and 'add-in'. It uses one parameter to identify the action to be executed, which can be 'start', 'stop', or 'check'. Additional parameters are needed. One is to make the script independent of the SAPSID, application server host name, actual name of the instance directory, instance number of the application server, etc. The invocation syntax of sapctrl_as is:

```
./sapctrl_as <resource-prefix> <sapsid> <host> <instance-dir> <instance-nr>
    <http-port-nr> <inst-type> <action>
```

For details on sapctrl_as see “Managing the application server instances with sapctrl_as” on page 348. The relevant part of the SA MP mkrsrc command for an ABAP-only or 'add-in' AS with instance number 95 would be:

```
mkrsrc ...
 StartCommand="<install dir>/sapctrl_as XI_ABAP EP0 lnxsapg D95 95 ABAP 0 START"
 StopCommand="<install dir>/sapctrl_as XI_ABAP EP0 lnxsapg D95 95 ABAP 0 STOP"
 MonitorCommand="<install dir>/sapctrl_as XI_ABAP EP0 lnxsapg D95 95 ABAP 0 CHECK"
 MonitorCommandPeriod=300
 MonitorCommandTimeout=10  ...
```

For a J2EE-only AS with instance number 95, the parameters would be:

```
mkrsrc ...
   StartCommand="<install dir>/sapctrl_as J2EE EP0 lnxsapg D95 95 95 50950 2 START"
 ...
 StartCommandTimeout=200
 StopCommandTimeout=200
```

For a J2EE 'add-in' AS with instance number 95, the parameters would be:

```
mkrsrc ...
   StartCommand="<install dir>/sapctrl_as XI_ABAP EP0 lnxsapg D95 95 95 50950 1 START"
 ...
 StartCommandTimeout=200
 StopCommandTimeout=200
```

We want SA MP to call our monitoring script every 300 seconds and give it 10 seconds to return the status. For J2EE 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.

For a J2EE 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.

An ‘add-in’ application server is physically one instance that runs both the ABAP and J2EE stacks. Although it is physically one instance, we separate it into its two logical parts, the ABAP application server and the J2EE application server. There is a close relationship between these two logical application servers. The J2EE instance is always started after the ABAP instance. This StartAfter relationship between the two logical instances guarantees that starting the J2EE instance automatically triggers the prior start of the ABAP instance. On the other hand, stopping the J2EE application server does not stop any of the ‘add-in’ server processes. It only stops the monitoring Java program ‘GetWebPage’. The HA policy for SAP (version 4) creates this 'StartAfter' relationship automatically if the corresponding ABAP application server resource has previously been created. Therefore, it is important to first create the ‘add-in’ ABAP application server resource(s) and then the corresponding ‘add-in’ J2EE application server resource(s).

We still need to specify the user name under which the AS should run. This is the <sapsid>adm userid:

```
mkrsrc ...
  UserName="ep0adm"
  ...
```

One more attribute is required. By default, SA MP expects any start or stop command to return within the given timeout, which is 5 seconds by default. The AS (and rfcping and GetWebPage), however, is ideally a never-ending-process. We need to tell SA MP not to wait for the start command to terminate and instead to start monitoring immediately. This is done with the following attribute:

```
mkrsrc ...
  RunCommandsSync=0"
```

The AS is not critical in terms of Tivoli System Automation. Therefore the default of attribute ProtectionMode=0 for class IBM.Application is acceptable.

The above is true for all application servers on all nodes, so you can substitute lnxsapg with lnxsaph and D95 with D96 and you then have the mkrsrc command for the application server with the instance number 96.

SAPSID-independent resources

Finally we look at the SAPSID-independent processes. These can be SAPROUTER, SAPWEBDISP, and SAPOSCOL. The following describes the creation of SAPROUTER, but creation of SAPWEBDISP is similar.

```
mkrsrc IBM.Application
  Name="XI_ABAP_SYS_ROUTER_SAPROUTER" (or "XI_ABAP_SYS_WEBDISP_SAPWEBDISP")
  ResourceType=1
  NodeNameList="{lnxsapg,lnxsaph,lnxsapi}"
  ...
```

The name is XI_ABAP_SYS_ROUTER_SAPROUTER because SAPROUTER is part of the ROUTER group, which is not related to any SAPSID (we therefore use SYS) but rather is part of SAP (see the naming conventions under “Tivoli System Automation for Multiplatforms” on page 120). We have a resource of type 1 (floating) because we have exactly one router that should run anywhere in the domain. We allow the SAPROUTER to run on lnxsapg, lnxsaph, and lnxsapi.

We have one script (sapctrl_sys) to manage all the SAPSID-independent processes. sapctrl_sys uses parameters to start, stop, and monitor (check) the different resources. Other parameters specify which resource is to be managed. The invocation syntax of sapctrl_sys is: .

```
./sapctrl_sys <resource-prefix> <sapsid> <instance-dir> <resource-identifier}
    <SAP routing table file> | SAP Web Dispatcher profile> <action>
```

For details on sapctrl_sys, see “Managing SAPSID-independent resources with sapctrl_sys” on page 353. The relevant part of the SA MP mkrsrc command (for the SAPROUTER) would be:

```
mkrsrc ...
  StartCommand="<install dir>/sapctrl_sys XI_ABAP EP0 ASCS00 RT /usr/sap/EP0/SYS/profile/saprouttab START"
  StopCommand="<install dir>/sapctrl_sys XI_ABAP EP0 ASCS00 RT /usr/sap/EP0/SYS/profile/saprouttab STOP"
  MonitorCommand="<install dir>/sapctrl_sys XI_ABAP EP0 ASCS00 RT /usr/sap/EP0/SYS/profile/saprouttab CHECK"
  MonitorCommandPeriod=120
  MonitorCommandTimeout=10  ...
```

We want Tivoli System Automation to call our monitoring script every 120 seconds and give it 10 seconds to return the status. Monitoring every 2 minutes seems to be appropriate, but other values could be chosen as well. There are no special considerations to take into account.

We still need to specify the user name under which the SAPROUTER should run. This is the <sapsid>adm userid:

```
mkrsrc ...
  UserName="ep0adm"
  ...
```

One more attribute is required. By default, SA MP expects any start or stop command to return within the given timeout, which is 5 seconds by default. The SAPROUTER, however, is, from an HA perspective, ideally a never-ending-process. We need to tell SA MP not to wait at all for the start command to terminate and start monitoring immediately instead. This is done with the following attribute:

```
mkrsrc ...
  RunCommandsSync=0
```

The SAPROUTER is not critical in terms of SA MP. Therefore, the default of attribute ProtectionMode=0 for class IBM.Application is acceptable.

SAPOSCOL: SAPOSCOL is not covered by version 4 of the sample HA policy for SAP. Therefore, you must create the resource group and the resource itself manually on each application server node if you need to automate SAPOSCOL. The following is a step-by-step description explaining how a SAPOSCOL resource group and resource can be created and what other changes are necessary.

1. Log on as **<sapid>adm**
2. In the startsap shell script in the SAP executable directory, comment out all ‘start_saposcol’ lines, and change all ‘echo ″start_saposcol;″’ lines to ‘echo ″NO start_saposcol;″’. Be sure to change startsap on the NFS server file system if the executable directory is shared. Make sure that this modified startsap script is used on each node.
3. As **root**, define the SAPOSCOL resource group and resources for each node of the SA MP domain where an SAP application server is running. The following are the sample SA MP commands to create the resource group and resource on node lnxsapg, where the saposcol executable is located in the SAP executable directory of SAP system EP0. Repeat these steps for each node of the TSA domain where an SAP application server is running. Adapt the commands to your actual SAP installation.
   a. Stop a running SAPOSCOL on node lnxsapg by issuing the command:

      `/usr/sap/EP0/SYS/exe/run/saposcol -k`

   b. Create the resource group for node lnxsapg:

      `mkrg -l none SAP_SYS_SAPOSCOL_LNXSAPG`

   c. Create the SAPOSCOL resource for node lnxsapg:

      ```
      mkrsrc IBM.Application Name="SAP_SYS_SAPOSCOL_EXE_LNXSAPG" ResourceType=0
        NodeNameList="{lnxsapg}" StartCommand="/usr/sap/EP0/SYS/exe/run/saposcol -l"
        StopCommand="/usr/sap/EP0/SYS/exe/run/saposcol -k"
        MonitorCommand="/usr/sbin/rsct/bin/pidmon /usr/sap/EP0/SYS/exe/run/saposcol"
        MonitorCommandPeriod=120 MonitorCommandTimeout=60 UserName="root"
        RunCommandsSync=0
      ```

   d. Add the SAPOSCOL resource to the group for node lnxsapg:

```
addrgmbr -m F -g SAP_SYS_SAPOSCOL_LNXSAPG
     IBM.Application:SAP_SYS_SAPOSCOL_EXE_LNXSAPG:lnxsapg
```

   e. Create the StartAfter relationship to the NFS server for node lnxsapg:

```
mkrel -S IBM.Application:SAP_SYS_SAPOSCOL_EXE_LNXSAPG:lnxsapg -p StartAfter
  -G IBM.Application:SA-nfsserver-server SAPOSCOL_on_lnxsapg_StartAfter_NFS
```

4. Start SAPOSCOL on all nodes via:

```
chrg –o online -s "Name like 'SAP_SYS_SAPOSCOL%'"
```

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 4 service IPs: one or two for ABAP and J2EE SCS and one or two for SAPROUTER/SAPWEBDISP. To achieve the collocation, we put the service IPs into the corresponding groups, where we used member location=collocated. Consider the mkrsrc commands for the enqueue IP:

```
mkrsrc IBM.ServiceIP
  Name="XI_ABAP_EP0_ENQ_IP"
  ResourceType=1
  NodeNameList="{lnxsapg,lnxsaph,lnxsapi}"
  ...
```

The name is `XI_ABAP_EP0_ENQ_IP` because the IP address is part of the ENQ group, which is part of SAPSID EP0, and the resource prefix in this case is XI_ABAP (see the naming conventions under “Tivoli System Automation for Multiplatforms” on page 120). ResourceType and NodeNameList are clear because we want the IP to move with the ES.

We need to specify the IP address and net mask to be used:

```
mkrsrc ...
  IPAddress="9.152.81.230"
  NetMask="255.255.248.0"
  ...
```

In our sample test environment, we did not use dynamic routing for SAP Central Services, etc. 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.

Note: 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.

IP addresses are critical in terms of SA MP. Therefore, the default of attribute `ProtectionMode=1` for the class IBM.Service is acceptable.

Note: You need the softdog kernel module when using critical resources.

Creating the resource groups

The following provides a more detailed look at the groups and the included resources. First of all, a few fundamental statements:

- An RSCT resource becomes an SA MP managed resource by adding it to a resource group.
- Each RSCT resource can be a member of only one resource group.
- Group members can be mandatory or optional.
- It is possible to have nested groups.

The SAP policy currently has no nested groups. It has a flat structure in which each resource is in exactly one group. A possible example of a nested group within the SAP policy would be the introduction of a group containing all application server groups. We decided against that, however, because application server resources should not affect each other in any way.

Now some words about mandatory or non-mandatory members. Mandatory member means that if such a member goes offline and cannot be brought online again on the same node, the complete group is brought offline, and, if possible, started on another node. For all groups that contain only one member, it is irrelevant whether this member is mandatory or not. The behavior is the same, so we can make them mandatory (which is the default).

Within our SAP policy, we have up to four groups that contain more than one member. These are the SAPROUTER, SAPWEBDISP, ASCS, and SCS groups. First, we consider the SAPROUTER group. In this group we have the SAPROUTER program and the associated service IP address. No matter which one is broken, the SAPROUTER would not be operational. So here we see that both members are mandatory. The same is true for SAPWEBDISP.

Now look at the ABAP and J2EE SCS groups. For ASCS, we have six resources: ES, MS, GW, SE, CO and IP. For SCS, we have three: ES, MS, and IP. The ES and MS resources are essential and mandatory of course. The same is true for the IP address. It is no use having ES and MS running when they cannot be reached from the network. Within the ASCS group, SE and CO deal with syslog information. If some of these messages are lost it might be a problem, but we have to weigh this point against the availability of the ES and MS. If we make the SE and/or CO mandatory and one of them fails, it would cause the complete ASCS group (including ES and MS) to be stopped and restarted on a different node. Because we consider the availability of ES and MS to be more important than losing syslog information, we chose to make SE and CO non-mandatory. Now what is left in the ASCS group is the GW. Here we have two possibilities:

- If the SAP system is set up such that users connect directly to a certain application server, GW is not important and can be non-mandatory.
- If the system is set up to use group logon, however, GW is indeed important and should be set to mandatory. In our sample policy, we chose to have it non-mandatory, though.

In summary, therefore:

- ES, MS, IP are mandatory
- GW, SE, CO are non-mandatory

Here is an example for creating the ASCS group and adding the members to it:

```
mkrg -l collocated XI_ABAP_EP0_ENQ (member location collocated is default)
addrgmbr -m T -g XI_ABAP_EP0_ENQ IBM.Application:XI_ABAP_EP0_ENQ_ES
addrgmbr -m T -g XI_ABAP_EP0_ENQ IBM.Application:XI_ABAP_EP0_ENQ_MS
addrgmbr -m F -g XI_ABAP_EP0_ENQ IBM.Application:XI_ABAP_EP0_ENQ_GW
addrgmbr -m F -g XI_ABAP_EP0_ENQ IBM.Application:XI_ABAP_EP0_ENQ_SE
addrgmbr -m F -g XI_ABAP_EP0_ENQ IBM.Application:XI_ABAP_EP0_ENQ_CO
addrgmbr -m T -g XI_ABAP_EP0_ENQ IBM.ServiceIP:XI_ABAP_EP0_ENQ_IP
```

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 42. 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
ABAP_instances.conf	Executable configuration file used by the above scripts to create ABAP resources for • ABAP SAP Central Services • ABAP application server(s) and SAP system independent resources • SAProuter • SAP Web Dispatcher
J2EE_instances.conf	Executable configuration file used by the above scripts to create J2EE resources for • J2EE SAP Central Services • J2EE application server(s)

Important: All scripts must reside in the same directory.

You must adapt ABAP_instances.conf to create SA MP resources for ABAP-only SAP systems. For J2EE-only SAP systems, you must adapt J2EE_instances.conf. For SAP systems that run both stacks (ABAP and J2EE) in parallel as so-called ‘add-in’ systems, such as SAP XI, you must adapt ABAP_instances.conf to create the ABAP resources and the SAP-system-independent ones. Additionally, you have to adapt J2EE_instances.conf to create the J2EE resources.

Specifying the ABAP configuration via ABAP_instances.conf

`ABAP_instances.conf` has the following content as delivered:

```
#!/bin/sh
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 2003-2005
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
#.------------------------------------------------------------.
#|                                                            |
#| TSA for Multiplatforms: HA policy for SAP     Version  4.2 |
#|                                                            |
#| Configuration Information                                  |
#| =========================                                  |
#|                                                            |
#| This script is used by mksap, rmsap and lssap scripts to   |
#| obtain configuration information.                          |
#| You must adapt the indicated lines to your environment.    |
#|                                                            |
#| VS: 2.6 Now contains entries for ENQSRV_IP_INTERFACE and   |
#|         SAPROUTER_IP_INTERFACE to generate an equivalency  |
#|         of the network interfaces which run corresponding  |
#|         (virtual) IP addresses.                            |
#| VS: 4.0 - Added SAP web dispatcher support                 |
#|         - Added optional host name for ASCS IP address     |
#|         - Removed ANSI codes for colored terminal output.  |
#| VS: 4.2 AIX and SAP kernel 7.00 support added.             |
#'------------------------------------------------------------'

###### START OF CUSTOMIZABLE AREA ################################################################

INSTALL_DIR="/home/lopadm/ha/salinux"                  # installation directory of SAP policy scripts
CLUSTER="LoP_XI_TSA"                                   # TSA domain name for below mentioned nodes; not used in HA scripts
NODES="siccps06 siccps07"                              # list of nodes included in the TSA domain
PREF="XI_ABAP"                                         # prefix of all TSA SAP ABAP resources and SAProuter/SAP web dispatcher
# for ABAP instance we recommend using <common prefix>_ABAP
SAPSID="LOP"                                           # SAP system ID
SAP_ADMIN_USER="lopadm"                                # SAP administration user ID
ENQSRV_IP="9.153.165.88"                               # ASCS instance IP address
ENQSRV_HOSTNAME="siccps29"                             # (virt.) hostname for ASCS IP address; if not defined, comment our of leave empty
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="02"                                             # instance number of the ASCS instance; for future use
ENQDIR="ASCS02"                                        # instance directory of the ASCS instance
ASNOS="00 00"                                          # list of instance numbers of the SAP App. servers
INSTDIRS="DVEBMGS00 D00"                               # list of instance directories of the SAP App. servers

# in our test setup for XI we do not have an SAP router...
#SAPROUTER_IP="9.153.165.xx"                            # SAProuter IP address
#SAPROUTER_IP_NETMASK="255.255.255.0"                   # SAProuter IP address' netmask
#SAPROUTER_IP_INTERFACE="eth0"                          # interface on which SAProuter IP address is activated on each node as alias
#ROUTTAB="/usr/sap/EP0/SYS/profile/saprouttab"          # fully qualified SAProuter routing table

# in our test setup for XI we do not have an SAP web dispatcher...
#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 on which SAP web dispatcher IP address is activated on each node as alias
#SAPWEBDISP_DIR="/usr/sap/EP0/sapwebdisp"               # directory where SAP web dispatcher is installed and where its profile is located

###### END OF CUSTOMIZABLE AREA #################################################################
```

Figure 81. ABAP_instances.conf file

Following the comment block is the customizable area. Here, you must adapt the policy to your installation. The comment for each parameter gives a short description. 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 (00) and first instance directory (DVEBMGS00) runs on node 1 (siccps06)
- the application server with the second instance number (00) and second instance directory (D00) runs on node 2 (siccps07)

and so on.

Specifying the J2EE configuration via J2EE_instances.conf

`J2EE_instances.conf` has the following content as delivered:

```
#!/bin/sh
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 2003-2005
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
#.-----------------------------------------------------------.
#|                                                           |
#| TSA for Multiplatforms: HA policy for SAP     Version  4.2 |
#|                                                           |
#| Configuration Information                                 |
#| =========================                                 |
#|                                                           |
#| This script is used by mksap, rmsap and lssap scripts to  |
#| obtain configuration information.                         |
#| You must adapt the indicated lines to your environment.   |
#|                                                           |
#| VS: 2.6 Now contains entries for ENQSRV_IP_INTERFACE and  |
#|         SAPROUTER_IP_INTERFACE to generate an equivalency |
#|         of the network interfaces which run corresponding |
#|         (virtual) IP addresses.                           |
#| VS: 4.0 - Added J2EE support                              |
#|         - Added optional host name for SCS IP address     |
#|         - Removed ANSI codes for colored terminal output. |
#| VS: 4.2 AIX and SAP kernel 7.00 support added.            |
#'-----------------------------------------------------------'

###### START OF CUSTOMIZABLE AREA ################################################################

INSTALL_DIR="/home/lopadm/ha/salinux"                 # installation directory of SAP policy scripts
CLUSTER="LoP_XI_TSA"                                  # TSA domain name for below mentioned nodes; not used in HA scripts
NODES="siccps06 siccps07"                             # list of nodes included in the TSA domain
PREF="XI_J2EE"                                        # prefix of all TSA SAP J2EE resources
# for J2EE instance we recommend to use <common prefix>_J2EE
ABAP_PREF="XI_ABAP"                                   # prefix of all TSA SAP ABAP resources; it is only needed, if the J2EE
                                                      # application server is part of an 'Add-In' application server (like for XI);
                                                      # it must have the same value as PREF in ABAP_instances.conf
SAPSID="LOP"                                          # SAP system ID
SAP_ADMIN_USER="lopadm"                               # SAP administration user ID
ENQSRV_IP="9.153.165.99"                              # SCS instance IP address
ENQSRV_HOSTNAME="siccps40"                            # (virt.) hostname for SCS IP address; if not defined, comment our of leave empty
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; for future use
ENQDIR="SCS01"                                        # instance directory of the SCS instance
ASNOS="00 00"                                         # list of instance numbers of the SAP J2EE App. servers
INSTDIRS="DVEBMGS00 D00"                              # list of instance directories of the J2EE SAP App. servers

# setting J2EE_HTTP_PORTS indicates, that profile is for J2EE
J2EE_HTTP_PORTS="50000 50000"                         # list of J2EE instance HTTP ports (default: 50000+instance number*100 +0)
###### END OF CUSTOMIZABLE AREA ##################################################################
```

Figure 82. J2EE_instances.conf file

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 (00) and first instance directory (DVEBMGS00) runs on node 1 (siccps06)
- the application server with the second instance number (00) and second instance directory (D00) runs on node 2 (siccps07)

and so on.

The entries that differ from the ABAP profile ("Specifying the ABAP configuration via ABAP_instances.conf" on page 338) are:

Table 43. Differences in the J2EE configuration compared to ABAP

PREF	We need a different prefix for J2EE resources.
ABAP_PREF	We need to know the prefix for ABAP resources. This entry is only necessary for the case in which the J2EE application server is part of an 'add-in' application server. We then need to know the ABAP resource prefix in order to be able to generate a StartAfter relationship between the J2EE and ABAP instances.
ENQSRV_xxx, ENQNO_xxx, ENQDIR_xxx	We have a separate SCS instance for J2EE.
J2EE_HTTP_PORTS	Each add-in J2EE instance needs an `http` port.

Setting up the policy (mksap)

If `mksap` is called without arguments, it performs the following steps, which are logged in file mksap.log, to set up the policy (the default configuration file is saphalinux.conf):

1. Call the configuration file (default: saphalinux.conf) to get the configuration information.
2. Create the SAPSID-independent resources (if not commented out). That is:
   - Create the ROUTER group.
   - Create SAPROUTER as a resource of class IBM.Application and add it to the ROUTER group.
   - Create IP as a resource of class IBM.ServiceIP and add it to the ROUTER group.
   - Create the relationship between SAPROUTER and IP.
   - Create the equivalency between the IP and network interface.
3. Create the ASCS for the specified SAPSID. That is:
   - Create the ENQ group. That is:
     - For each of ES, MS, CO, SE, GW
       - Create a resource of class IBM.Application with identical attributes (different name and arguments of sapctrl_em of course). This resource will be floating across all nodes specified in NODES.
       - Add it to the ENQ group: ES and MS as mandatory, the rest as non-mandatory members.
     - Create IP as resource of class IBM.ServiceIP.
     - Add IP to ENQ group and mandatory member.
   - Create the ENQREP group. That is:
     - Create ERS as resource of class IBM.Application.
     - Add it to ENQREP group as mandatory member.
   - Create the relationships between ES, MS, CO, SE, GW, and IP.
   - Create the relationships between ES and ERS and vice versa.
   - Create the equivalency between the IP and network interface.
4. For each node specified in NODES
   - Create the ABAP application server instance for the specified SAPSID. That is:

- Create application server group with the name

  `<node name>_D<instance number>`

  .
- Create AS as a resource of class IBM.Application. This will be fixed on the given node.
- Add AS to application server group.
- Create the relationships between AS and ES+MS.

The mksap script creates, or appends to, the mksap.log file to log all generated and executed SA MP commands and a list of all generated resources.

You might want to have different setups:

- not every node has an application server
- more than one application server on a single node
- SCS floating on a subset of nodes
- SAPROUTER not desired

In this case, you must call mksap with the correct argument set and the correct settings in one (or more) configuration files. 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]]
mksap creates (parts of) the sample SA MP policy for a highly available SAP environment.
  -h,  Print this help.
  -s,  Use static equivalencies for sample SA MP 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 SA MP policy entries for an SAP application server on NodeName and add them to an existing SA MP policy.
  -t,  Define the type of the sample SA MP policy to be created; valid types: CentralServices|SAPCommon|AppServer;  default: None, create all.
  -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 SA MP policy entries for all nodes specified; default: create entries for all nodes defined in the configuration file.
```

For example, to create an SA MP policy for a highly available SAP XI system that runs on a two-node domain, with an ‘add-in’ application server on each node, we ran mksap with different argument settings (ABAP_instances.conf and J2EE_instances.conf as shown above). As user `root`:

1. `./mksap -c ABAP_instances.conf -t CentralServices -i ABAP`

   This created the SA MP resources, etc. for the ABAP SCS.
2. `./mksap -c ABAP_instances.conf -t AppServer -i ABAP -n siccps06,siccps07`

   This created the SA MP resources, etc. for ABAP application servers on nodes siccps06 and siccps07.

   Note: Option -n <node list> is not necessary if an application server resource should be created for each node listed in the configuration file entry ‘NODES’.
3. `./mksap -c ABAP_instances.conf -t SAPCommon`

   This created the SA MP resources, etc. for the SAProuter (SAP Web Dispatcher), because the corresponding entries in the configuration file were active (not commented out).
4. `./mksap -c J2EE_instances.conf -t CentralServices -i ADDIN`

   This created the SA MP resources, etc. for the J2EE SCS.
5. `./mksap -c J2EE_instances.conf -t AppServer -i ADDIN`

   This created the SA MP resources, etc. for a J2EE application server on each node defined in the configuration file entry ‘NODES’, in our case on nodes siccps06 and siccps07. It also generated the StartAfter relationship(s) between the J2EE application server and its corresponding ABAP application server.

Note: This relation can only be created if the ABAP application server was created previously.

To create an SA MP policy for a highly available SAP Enterprise Portal 6.0 (EP 6.0) system that runs on a two-node domain, with a J2EE-only application server on each node, we ran:

1. `./mksap -c ABAP_instances.conf -t SAPCommon`

   This created the SAP Web Dispatcher resources because the corresponding entries in the configuration file were active (not commented out).
2. `./mksap -c J2EE_instances.conf -t CentralServices -i J2EE`

   This created the J2EE SCS resources
3. `./mksap -c J2EE_instances.conf -t AppServer  -i J2EE -n siccps43,siccps44`

   This created the J2EE-only application server resources on nodes siccps43 and siccps44.

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 <PREF>_<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 <PREF>_<SAPSID>
   - 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>

Note: All resources must be offline before you can remove them.

If you want to remove resources with different prefixes, you have to use different configuration files.

The rmsap script supports the following command line options:

```
./rmsap -h
Usage: rmsap [-h][-c ConfigFile]]
rmsap removes all currently defined resources of the sample SA MP policy for a highly available SAP environment.
  -h,  Print this help.
  -c,  Use configuration file ConfigFile (fully qualified or in current dir); default: saphasalinux.conf in current dir.
```

Monitoring the status of the policy (lssap)

If lssap is called without any arguments, it performs the following actions to gather status information of the groups and resources of the policy (the default configuration file name is saphalinux.conf):

1. Call the configuration file (default: saphalinux.conf) to get the configuration information.
2. Start an infinite loop with the following:
   - Use lsrsrc-api to get status information of all groups that start with the prefix specified in <PREF>.
   - Use lsrg to get the status and membership information of all resources that start with the prefix specified in <PREF>.
   - Use lsrsrc-api to get status information of all resources of classes IBM.Application and IBM.ServiceIP that start with the prefix specified in <PREF>.
   - Prepare output from the gathered information.
     - For each group found
       - Print group name, observed and nominal state.
       - For each member within group.
         - Print resource name and observed state.
         - If online, print node.
   - Send output to terminal.
   - Sleep for 10 seconds (to avoid unnecessary resource loading).

You can stop lssap by pressing CTRL-C.

The lssap script also supports the following command line options:

```
./lssap -h
Usage: lssap [-h][-c ConfigFile] [-p ResourcePrefix]]
lssap displays the status of all currently defined resources of the sample SA MP policy for a highly available SAP environment.
  -h,  Print this help.
  -c,  Use configuration file ConfigFile (fully qualified or in current dir); default: saphasalinux.conf in current dir.
  -p,  Use ResourcePrefix as the prefix for all resources to display; overrides the PREF entry of configuration file, if present.
```

For example, if you run

```
./lssap –p XI
```

you get a list of all currently defined resources starting with the characters 'XI'. Using '–p %' as an argument shows you all currently defined resources.

Automation scripts

We deliver the following scripts that are used for starting, monitoring and stopping the resources of the sample policy:

Table 44. 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 (J2EE SCS)
sapctrl_as	Manages the application server resources, which can be ABAP-only, J2EE-only, or -'add-in' application servers
sapctrl_sys	Manages the SAPSID-independent resources (currently SAPROUTER and/or SAPWEBDISP)

Monitoring or stopping an operating system process with sapctrl_pid

We had to write our own process identifier (pid) monitor (sapctrl_pid) to get the status of a process, because the pid monitor (`pidmon`) supplied with SA MP does not support

- stop command sequences
- Java processes
- checking the status of a secondary process if the status of the primary process is 'offline'

When looking at a process from the outside, you can only tell whether it exists or not. You do not know how healthy an existing process is. For example, it may be in a loop and therefore unusable. So the only states that sapctrl_pid can deliver are ’online’ and ’offline’ (or ’unknown’ in case of an error during execution of sapctrl_pid). That is sufficient for our purposes, however.

The invocation syntax for sapctrl_pid is:

```
./sapctrl_pid <action> <symbolic name> <process> [<action-dependent parameters>]
```

Parameters

<action>
Action to be performed

<symbolic name>
Symbolic name that appears in messages in the syslog, for example.

<process>
Process affected (as shown in the `pidmon` output)

<action-dependent parameters>
Additional parameters depending on the action.

Monitoring a process

The invocation syntax for monitoring a process is:

```
./sapctrl_pid CHECK <symbolic name> <process>
```

sapctrl_pid performs the following actions to acquire the status:

- If the process is not a Java process, runs pidmon for the process
- If pidmon returns 1, it is 'online', otherwise 'offline'
- If the process is a Java process, performs a 'ps -ef' command to check for a process like 'java GetWebPage'.
- If the Java process exists, it returns 'online', otherwise 'offline'

Some of the SA MP processes have a ’regular’ stop command, while others should be stopped by sending a SIGINT signal. Unfortunately, sometimes the processes do not stop as intended, so it is required to send a SIGKILL after a certain time. SA MP does not offer such a stop command escalation, so sapctrl_pid takes care of this.

Stopping a process

The invocation syntax for stopping a process is:

```
./sapctrl_pid STOP <symbolic name> <process> command [waittime command] ...
```

Action-dependent parameters:

<command> [waittime command]
One or more stop commands separated by a wait time in seconds.

sapctrl_pid performs the following actions to stop the given process:

- Extract the first stop command
- If it is the special command to stop the J2EE application server of an 'add-in' instance, kill only the primary health-check process 'java GetWebPage' of that application server
- Otherwise, run the stop command (under AIX, a killall command is translated to kill)
- While there are commands left in the list
  - Set status to online
  - While wait time has not elapsed and status still online
    - Sleep for one second
    - Call sapctrl_pid with option CHECK to update the status
  - If the status is now offline then verify that the process in fact no longer exists
    - Sleep again for one second
    - If the stop was *not* for an application server, call sapctrl_pid a second time for verification
    - If the stop was for an application server (the process is one of the primary health check processes) then check additionally (during the rest of the `waittime` interval) whether a corresponding secondary stopsap command is still running. In this case, sapctrl_pid returns ’online’ although the primary health check process is ’offline’. This behavior is necessary because, during an application server stop, the health check is stopped almost immediately, while the application server processes continue running, and stopping them takes some time.
  - Get the next stop command in the list

sapctrl_pid is called by the other automation scripts, not directly from SA MP. Note that you can adjust the amount of information written to the syslog by specifying a different value for the variable DEBUG inside the script.

Managing the ABAP or J2EE SCS with sapctrl_em

sapctrl_em is used to start, stop, and monitor the resources of the ABAP and J2EE SCS instances.

The invocation syntax for 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

<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>
If 1, the resource executables are located in the special `.../SYS/exe/runU` (U for Unicode) directory instead of `...SYS/exe/run`. You must set this for the 'add-in' J2EE SCS when the ABAP part is running with the ASCII character set and the J2EE part is running Unicode.

<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`)

Starting a resource

Note: After starting a resource, sapctrl_em subsequently enters a wait state in order to detect when the resource has stopped. It then triggers monitoring by SA MP to allow it to immediately register the 'offline' status. When you issue sapctrl_em manually, you can use the command-line terminator "&" to ensure that control is returned immediately to the invoking shell.

The invocation syntax for starting a resource is:

```
./sapctrl_em ... START
```

sapctrl_em performs the following actions:

1. Set some variables according to the standard SAP naming conventions. For instance, the profile data set, and the work and executable directories are built from the SAPSID. Also, a symbolic name the executable to start, and its arguments are defined.

   Notes:

   a. If you did not follow the SAP standard installation you must adapt the script.
   b. If you made changes to mksap or wrote your own script to set up the resources, you may encounter difficulties at this point.

2. Check if the work directory is accessible and, if so, change to the work directory. Otherwise, exit with error messages.
3. Create a symbolic link to the executable in the work directory and start it with the proper arguments
4. After 10 seconds, trigger monitoring by calling action routine ’refreshOpState’ using runact-api. Here, we need the resource name as defined in SA MP. We wait 10 seconds to be sure that process has been started in the background. Triggering monitoring then makes sure that SA MP shows the status ’online’ immediately, not only when the monitoring interval has elapsed.
5. Wait until the process stops and then trigger monitoring again via ’refreshOptState’ to allow SA MP to react immediately to the stopping of the resource.

Stopping a resource

The invocation syntax for stopping a resource is:

```
./sapctrl_em ... STOP
```

sapctrl_em performs the following actions to stop a resource:

- Set up the stop sequence for the resource. Within the SCS, all resources are stopped in the same way. We issue a killall -2 and, if after 30 seconds the process is still alive, we issue a killall -9. We noticed that killall -2 does not work for the enqueue server under Linux on System z. Therefore, when using the script as delivered, stopping the enqueue server under Linux on System z will always require at least 30 seconds before the killall -9 is issued.
- Call sapctrl_pid with the STOP argument and pass the stop sequence.

Monitoring a resource

The invocation syntax for monitoring a resource is:

```
./sapctrl_em ... CHECK
```

sapctrl_em performs the following actions to monitor a resource:

- Call sapctrl_pid with the CHECK argument
- Return the status as reported by sapctrl_pid

Note: You can adjust the amount of information written to the syslog by specifying a different value for the variable DEBUG inside the script.

Managing the application server instances with sapctrl_as

sapctrl_as is used to start, stop, and monitor the application server resources.

The invocation syntax for sapctrl_as is:

```
./sapctrl_as <resource-prefix> <sapsid> <host> <instance-dir> <instance-nr>
    <http-port-nr> <inst-type> <action> [&]
```

Parameters

<resource-prefix>
Prefix of the application server resource name

<sapsid>
SAPSID to which the server belongs

<host>
Host on which the server runs

<instance-dir>
Instance directory of the server

<instance-nr>
Instance number of the server

<http-port-nr>
Port number via which the J2EE instance can be accesssed. For ABAP-only application servers it is not used, but it must be present (use "ABAP", for example)

<inst-type>
Application server type (0 means ABAP, 1 means 'add-in' and 2 means J2EE).

<action>
Action to be performed for the resource (`START|STOP|CHECK`).

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 (e.g. 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 J2EE 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. It tries to connect to a Web page, in our case the `index.html` file of a J2EE application server. If this page is reachable, it means that the J2EE application server is operational. If not, it indicates that the J2EE 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 J2EE 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>
Prefix of the SCS resource names

<URL>
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
```

The sapctrl_as script can be used for ABAP-only, J2EE-only and ‘add-in’ application servers. An ‘add-in’ application server is physically a single instance that runs both the ABAP and J2EE stacks simultaneously. If you start an ‘add-in’ application server, both stacks are started. If you stop it, both stacks are stopped by default. Although it is physically one instance, we separated it into its two logical parts, the ABAP application server and the J2EE application server. In other words, within an SA MP domain, one ‘add-in’ application server instance is automated as two logical application server instances, an ABAP application server instance and a J2EE application server instance. However, there is a close relationship between these two logical application servers. The SA MP sample policy generates a StartAfter relationship between the J2EE and ABAP instances. This StartAfter relationship guarantees that starting the J2EE instance automatically triggers the prior start of the ABAP instance. If the sapctrl_as script is used to start the J2EE stack of an ‘add-in’ application server instance, the ‘add-in’ (ABAP and J2EE) is started and the script waits until the health checker Java program GetWebPage is active. On the other hand, stopping the J2EE application server does not stop any of the ‘add-in’ server processes. It only stops the Java health checker program ‘GetWebPage’. Additionally, the logical ABAP application server cannot be stopped before the logical J2EE application server has been stopped.

Starting an application server

Note: After starting an application server, sapctrl_as subsequently enters a wait state in order to detect when the server has stopped. It then triggers monitoring by SA MP to allow it to immediately register the 'offline' status. When you issue sapctrl_as manually, you can use the command-line terminator "&" to ensure that control is returned immediately to the invoking shell.

The invocation syntax for starting an application server via sapctrl_as is:

```
./sapctrl_as ... START
```

sapctrl_as performs the following actions:

- Set some variables according to the standard SAP naming conventions. For instance, the profile data set and the work and executable directories are built from the SAPSID.

  Notes:

  1. If you did not follow the SAP standard installation you must adapt the script.
  2. If you made changes to mksap or wrote your own script to set up the resources, you may encounter difficulties at this point.
- Check that the work directory is accessible and, if so, change to it. Otherwise, exit with an error message. Check that the SAP kernel version is supported.
- Define the command (runact-api) to be run to trigger SA MP monitoring of the application server. This allows SA MP to detect a status change in the application server almost immediately.

- Create stop and start command sequences depending on the SAP kernel release and application server type.
- Create a symbolic link for the health checker program, depending on the application server type.
- Call a single rfcping/GetWebPage to check whether the application server is already running. Because we monitor the rfcping/GetWebPage process and not the application server itself, it may happen that only the rfcping/GetWebPage dies because there was a temporary problem with the connection to the database, for example. In such a case, the application server is still working properly and we detect this directly. When checking the J2EE part of an 'add-in' application server with GetWebPage, we wait much longer in a loop, because starting the J2EE part may have triggered the start of the ABAP part, and we therefore have to wait much longer in order to allow the complete 'add-in' application server to start..
- If the application server is not running
  - For an ABAP application server, first check that the database can be accessed, by executing `R3trans -d`. If it is not accessible, do not start the application server at this time.
  - Start ABAP- and J2EE-only application servers using the command sequence

    ```
    cleanipc; stopsap; startsap
    ```

    with the proper arguments. This is done because we cannot be sure that the application server is really down. rfcping/GetWebPage only says that the 'dummy' request did not complete successfully. But the application server might be hung up as well. We therefore clean up and stop it as a precaution.

    Starting only the J2EE part of an 'add-in' application server is done with a

    ```
    startsap j2ee
    ```

    command. cleanipc and stopsap cannot be issued, because that would stop the already running ABAP part. If the application server has already been started, the start request fails. Otherwise, a start is attempted.
  - If the start return code is OK
    - Try 12 (ABAP-only), 15 ('add-in'), or 24 (J2EE-only) times
      - Call single rfcping/GetWebPage to check whether the application server is now active
      - If active, leave loop
      - If last try
        - Return status 'failed'
        - Trigger monitoring by calling action routine 'refreshOpState' using runact-api. Here, we need the resource name as defined in SA MP. Triggering monitoring then ensures that SA MP shows status 'failed' immediately and not when the monitoring interval has elapsed.
    - Sleep for 10 seconds
  - If the start return code is not OK
    - Copy the startsap log file to a startsap error log file in order to protect it from being overwritten by the next startup attempt.
    - Trigger monitoring by calling action routine 'refreshOpState' using runact-api. Here we need the resource name as defined in SA MP. Triggering monitoring then makes sure that SA MP shows status failed immediately and not when the monitoring interval has elapsed.
    - Exit with status 'failed'

- Spawn infinite rfcping/GetWebPage process
- After 10 seconds, trigger monitoring by calling action routine 'refreshOpState' using runact-api. Here we need the resource name as defined in SA MP. We wait 10 seconds to be sure that the process has been started in the background. Triggering monitoring then makes sure that SA MP shows status ’online’ immediately, not only when the monitoring interval has elapsed.
- Wait until the process stops and then trigger monitoring again via ’refreshOpState’ to allow SA MP to react to the stopping of the resource.

Stopping an application server

Stop application server: The invocation syntax for stopping an application server is:

```
./sapctrl_as ... STOP
```

sapctrl_as performs the following actions to stop the resource:

- Set some variables according to the standard SAP naming conventions. For instance, the profile data set and the work and executable directories are built from the SAPSID.

  Notes:

  1. If you did not follow the SAP standard installation you must adapt the script.
  2. If you made changes to mksap or wrote your own script to set up the resources, you may encounter difficulties at this point.
- Check that the work directory is accessible and, if so, change to it. Otherwise, exit with an error message. Check that the SAP kernel version is supported.
- Define the command (runact-api) to be run to trigger SA MP monitoring of the application server. This allows SA MP to detect a status change in the application server almost immediately.
- Create start and stop command sequences depending on the SAP kernel release and application server type.
- Create a symbolic link for the health checker program, depending on the application server type.
- Call sapctrl_pid with the STOP argument and pass the stop sequence. As soon as the application server is stopped, the spawned rfcping/GetWebPage process will also terminate.
- After one second, trigger monitoring by calling action routine 'refreshOpState' using runact-api.

Monitoring an application server

The invocation syntax for monitoring an application server (in fact its rfcping or GetWebPage process) is:

```
./sapctrl_as ...  CHECK
```

sapctrl_as performs the following actions to monitor the resource:

- Set some variables according to the standard SAP naming conventions. For instance, the profile data set and the work and executable directories are built from the SAPSID.

  Notes:

  1. If you did not follow the SAP standard installation you must adapt the script.

2. If you made changes to mksap or wrote your own script to set up the resources, you may encounter difficulties at this point.

- Check that the work directory is accessible and, if so, change to it. Otherwise, exit with an error message. Check that the SAP kernel version is supported.
- Define the command (runact-api) to be run to trigger SA MP monitoring of the application server. This allows SA MP to detect a status change in the application server almost immediately.
- Create stop command sequences depending on the SAP kernel release and application server type.
- Call sapctrl_pid with the CHECK argument. We initially monitor only the pid of the rfcping/GetWebPage process, not the application server itself.
- If the health checker program is no longer running, we also check whether a stopsap for the application server is running.
- If stopsap is still running, return an 'offline pending' status to SA MP, otherwise the status as reported by sapctrl_pid.

Note: You can adjust the amount of information written to the syslog by specifying a different value for the variable DEBUG inside the script.

Managing SAPSID-independent resources with sapctrl_sys

sapctrl_sys is used to start the SAPSID-independent resources. Currently we have two, SAPROUTER and SAPWEBDISP.

The invocation syntax for sapctrl_sys is:

```
./sapctrl_sys <resource-prefix> <sapsid> <instance-dir> <resource-identifier>
     <SAP routing table file | SAP Web Dispatcher profile> <action>
```

Parameters

<resource-prefix>
Prefix of the SAPSID-independent resource name

<sapsid>
Needed to locate the directory in which the resource executable can be found.

<instance-dir>
Instance directory used to determine the work directory

<resource-identifier>
'RT' (SAProuter) or 'WD' (SAP Web Dispatcher)

<file identifier>
File containing the routing table for SAProuter or the profile for the SAP Web Dispatcher.

<action>
Action to be performed for the resource (`START|STOP|CHECK`)

Starting a SAPSID-independent resource

Note: After starting a resource, sapctrl_sys subsequently enters a wait state in order to detect when the resource has stopped. It then triggers monitoring by SA MP to allow it to immediately register the 'offline' status. When you issue sapctrl_sys manually, you can use the command-line terminator "&" to ensure that control is returned immediately to the invoking shell.

The invocation syntax for starting an SAPSID-independent resource is:

```
./sapctrl_sys ... START
```

sapctrl_sys performs the following actions:

- Set some variables according to the standard SAP naming conventions. For instance the profile data set and the work and executable directories are built from the SAPSID.

  Notes:

  1. If you did not follow the SAP standard installation you must adapt the script.
  2. If you made changes to mksap or wrote your own script to set up the resources, you may encounter difficulties at this point.
- Create a symbolic name and the executable to start (including its arguments)
- Change to the work directory
- Create a symbolic link to the executable in the work directory and start it with the proper arguments
- After 10 seconds, trigger monitoring by calling action routine ’refreshOpState’ using runact-api. Here, we need the resource name as defined in SA MP. We wait 10 seconds to be sure that the process has been started in the background. Triggering monitoring then makes sure that SA MP shows status ’online’ immediately and not only when the monitoring interval has elapsed.
- Wait until the process stops and then trigger monitoring again via ’refreshOpState’ to allow SA MP to react to the stopping of the resource.

Stopping a SAPSID-independent resource

The invocation syntax for stopping a SAPSID-independent resource is:

```
./sapctrl_sys ... STOP
```

sapctrl_sys performs the following actions to stop a SAPSID-independent resource:

- Set some variables according to the standard SAP naming conventions. For instance the profile data set and the work and executable directories are built from the SAPSID.

  Notes:

  1. If you did not follow the SAP standard installation you must adapt the script.
  2. If you made changes to mksap or wrote your own script to set up the resources, you may encounter difficulties at this point.
- Create a symbolic name and the executable to stop
- Set up the stop sequence for the resource. For example, for SAPROUTER we invoke the executable with the -s option and if after 20 seconds the process is still alive we issue a killall -9.
- Call sapctrl_pid with the STOP argument and pass the stop sequence.

Monitoring a SAPSID-independent resource

The invocation syntax for monitoring a SAPSID-independent resource is:

```
./sapctrl_sys ... CHECK
```

Here, the last argument says that we want to monitor the given resource. sapctrl_sys performs the following actions to monitor a SAPSID-independent resource:

- Set some variables according to the standard SAP naming conventions. For instance the profile data set and the work and executable directories are built from the SAPSID.

  Notes:

  1. If you did not follow the SAP standard installation you must adapt the script.
  2. If you made changes to mksap or wrote your own script to set up the resources, you may encounter difficulties at this point.
- Create a symbolic name and the executable to monitor
- Call sapctrl_pid with the CHECK argument.
- Return the status as reported by sapctrl_pid.

Note: You can adjust the amount of information written to the syslog by specifying a different value for the variable DEBUG inside the script.

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

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

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
RFCOSCOL	RFC OS Collector
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
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 J2EE 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 Attach Name. The name the ICLI server uses to attach to the DB2 subsystem.

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 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). 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 J2EE 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 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 an 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.

Bibliography

IBM documents

In the IBM Collection Kits listed in Table 45 you will find most of the documents that provide information related to the topics covered in this publication.

Table 46 through Table 50 on page 371 list the IBM documents you will most likely have to consult in addition to this one. In most cases, the titles and form numbers given apply to the minimum release of the software required to run the SAP software described in this document. Always use the latest edition of a manual that applies to the software release running on your system.

Table 51 on page 371 lists the IBM Redbooks that contain further information of interest.

Table 45. List of IBM Collection Kits

IBM Online Documents	Collection Kit Number
z/OS V1Rx Collection	SK3T-4269
z/OS Software Products Collection	SK3T-4270
IBM eServer zSeries Redbooks Collection	SK3T-7876
DB2 for OS/390 Licensed Online Books	SK2T-9075
DB2 for OS/390 Online Library	SK2T-9092
Transaction Processing and Data Collection	SK2T-0730

Table 46. Selected IBM DB2 publications

IBM DB2 Documents	Order Number (V7)	Order Number (V8)
Administration Guide	SC26-9931	SC18–7413
Application Programming and SQL Guide	SC26-9933	SC18–7415
Command Reference	SC26-9934	SC18–7412
Data Sharing: Planning and Administration	SC26-9935	SC18-7417
Installation Guide	GC26-9936	GC18–7418
Messages and Codes	GC26-9940	GC18–7422
Reference for Remote DRDA Requesters and Servers	SC26-9942	SC18–7424
Reference Summary	SX26-3847	SX26–3853
Release Planning Guide	SC26-9943	SC18–7425
SQL Reference	SC26-9944	SC18–7426
Utility Guide and Reference	SC26-9945	SC18–7427
What's New?	GC26-9946	SC18–7428

Table 47. Selected IBM z/OS publications

IBM Documents	Order Number
z/OS DFSMS Implementing System-Managed Storage	SC26-3123
z/OS DFSMS Managing Catalogs	SC26-7409

Table 47. Selected IBM z/OS publications (continued)

IBM Documents	Order Number
z/OS DFSMSdss Storage Administration Guide	SC35-0423
z/OS DFSMSdss Storage Administration Reference	SC35-0424
z/OS Communications Server: CSM Guide	SC31-8808
z/OS Communications Server IP Configuration Reference	SC31-8776
z/OS Communications Server: IP Migration Guide	GC31-8773
z/OS Communications Server: IP User's Guide and Commands	SC31-8780
z/OS Distributed File Service SMB Administration Guide and Reference	SC24-5918
z/OS Hardware Configuration Definition (HCD) User's Guide	SC33-7988
z/OS Language Environment Debugging Guide and Run-Time Messages	GA22-7560
z/OS Language Environment Programming Guide	SA22-7561
z/OS Language Environment Programming Reference	SA22-7562
z/OS MVS Initialization and Tuning Reference	SA22-7592
z/OS MVS JCL Reference	SA22-7597
z/OS MVS Planning: APPC Management	SA22-7599
z/OS MVS Planning: Global Resource Serialization	SA22-7600
z/OS MVS Planning Workload Management	SA22-7602
z/OS MVS Programming: Resource Recovery	SA22-7616
z/OS MVS Programming: Sysplex Services Guide	SA22-7617
z/OS MVS Programming: Workload Management Services	SA22–7619
z/OS MVS Setting Up a Sysplex	SA22-7625
z/OS MVS System Commands	SA22-7627
z/OS Network File System Customization and Operation	SC26-7417
z/OS Network File System User's Guide	SC26-7419
z/OS Resource Measurement Facility (RMF) Performance Management Guide	SC33-7992
z/OS Communications Server: IP Configuration Guide	SC31-8775
z/OS SecureWay Security Server RACF Security Administrator's Guide	SA22-7683
z/OS SecureWay Security Server External Security Interface (RACROUTE) Macro Reference	SA22-7692
z/OS UNIX System Services Messages and Codes	SA22-7807
z/OS UNIX System Services Command Reference	SA22-7802
z/OS UNIX System Services Planning	GA22-7800
z/OS UNIX System Services User's Guide	SA22-7801

Tivoli System Automation for z/OS

See also `http://www.ibm.com/servers/eserver/zseries/software/sa`

Table 48. Selected IBM Tivoli System Automation for z/OS publications

IBM Documents	Order Number
Tivoli System Automation for z/OS (Release 3.1)	
System Automation for z/OS: Customizing and Programming	SC33-8260-03

Table 48. Selected IBM Tivoli System Automation for z/OS publications (continued)

IBM Documents	Order Number
System Automation for z/OS: Planning and Installation	SC33-8261-02
System Automation for z/OS: Programmer's Reference	SC33-8266-01
System Automation for z/OS: User's Guide	SC33-8263-02
Tivoli System Automation for z/OS (Release 2.3)	
System Automation for z/OS: Customizing and Programming	SC33-7035-09
System Automation for z/OS: Planning and Installation	SC33-7038-09
System Automation for z/OS: Programmer's Reference	SC33-7043-11
System Automation for z/OS: User's Guide	SC33-7040-08

Tivoli System Automation for Multiplatforms

See also

`http://www.ibm.com/software/tivoli/products/sys-auto-linux`

Table 49. Selected IBM Tivoli System Automation for Multiplatforms publications

IBM Documents	Order Number
Tivoli System Automation for Multiplatforms (Release 2.1)	
Tivoli System Automation for Multiplatforms: Base Component User's Guide	SC33-8210-04
Tivoli System Automation for Multiplatforms: Base Component Reference	SC33-8212-01
Tivoli System Automation for Multiplatforms: End-to-End Automation Management User's Guide and Reference	SC33-8211-01
Tivoli System Automation for Multiplatforms (Release 1.2)	
Tivoli System Automation for Multiplatforms: Guide and Reference	SC33-8210-02
Tivoli System Automation for Multiplatforms: Application Enablement of NFS File Server	

Table 50. Other IBM reference documents

IBM Documents	Order Number
Porting Applications to the OpenEdition MVS Platform	GG24-4473
SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th Edition	SC33-7965-03
SAP R/3 on DB2 for OS/390: Planning Guide 2nd Edition; SAP R/3 Release 4.6D	SC33-7966-04
SAP on DB2 UDB for OS/390 and z/OS: Planning Guide; SAP Web Application Server 6.10	SC33-7959-00
SAP on DB2 UDB for z/OS: Planning Guide 2nd Edition; SAP Web Application Server 6.20	SC33-7959-02
S/390 ESCON Channel PCI Adapter: User's Guide and Service Information.	SC23-4232
Using REXX and z/OS UNIX System Services	SA22-7806
Linux for zSeries Device Drivers, Features, and Commands (Linux kernel 2.6)	SC33-8289
Linux for zSeries Device Drivers, Features, and Commands (Linux kernel 2.4)	SC33-8282
OSA-Express Customer's Guide and Reference	SA22-7403

Table 51. IBM Redbooks and Redpapers covering related topics

IBM Redbooks/Redpapers (published by the IBM International Technical Support Organization, ITSO)	Order Number
SAP R/3 on DB2 for OS/390: OS/390 Application Server	SG24-5840

Table 51. IBM Redbooks and Redpapers covering related topics (continued)

IBM Redbooks/Redpapers (published by the IBM International Technical Support Organization, ITSO)	**Order Number**
SAP R/3 on DB2 for OS/390: Disaster Recovery	SG24-5343
SAP on DB2 for z/OS and OS/390: DB2 System Cloning	SG24-6287
Open Source Software for z/OS and OS/390 UNIX	SG24-5944
Implementing SAP R/3 in an OS/390 Environment Using AIX Application Servers	SG24-4945
SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations [1]	SG24-5690
SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution Using System Automation [1]	SG24-6836
SAP on DB2 for z/OS and OS/390: High Availability and Performance Monitoring with Data Sharing [1]	SG24-6950
SAP on DB2 Universal Database for OS/390 and z/OS: Multiple Components in One Database (MCOD)	SG24-6914
DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There	SG24-7088
SAP on DB2 UDB for OS/390 and z/OS - Implementing Application Servers for Linux on zSeries	SG24-6847
DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know , ... and More	SG24-6079
DB2 UDB for z/OS Version 8 Technical Preview	SG24-6871
Distributed Functions of DB2 for z/OS and OS/390	SG24–6952
zSeries HiperSockets	SG24–6816
mySAP Business Suite Managed by IBM Tivoli System Automation for Linux (Redpaper) [1]	REDP-3717
Linux on IBM zSeries and S/390: VSWITCH and VLAN Features of z/VM 4.4 (Redpaper)	REDP-3719
[1] Information from these publications was updated and used as the basis for the current book.	

SAP documents

SAP documents are available in the SAP Service Marketplace under `http://service.sap.com/instguides`

Table 52. SAP publications

SAP High Availability
BC SAP High Availability (SAP online documentation is available in the SAP Library or at `http://service.sap.com/ha`)
SAP Web Application Server 6.20
SAP on DB2 UDB for OS/390 and z/OS: Database Administration Guide: SAP Web Application Server
SAP Web Application Server Heterogeneous System Copy
SAP Web Application Server Homogeneous System Copy
SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS
SAP Web Application Server Installation on Windows: IBM DB2 UDB for OS/390 and z/OS
SAP Software on UNIX: OS Dependencies
SAP NetWeaver '04 (SAP Web Application Server 6.40)

Table 52. SAP publications (continued)

The installation guides each comprise two documents: • *Part I: Planning and Preparation* • *Part II: Installation and Post-Installation* **Note:** Publications are listed for ABAP-based application servers. Corresponding publications are also available for Java-based application servers.
Component Installation Guide: SAP Web Application Server ABAP 6.40 on AIX: IBM DB2 UDB for z/OS
Component Installation Guide: SAP Web Application Server ABAP 6.40 on Linux: IBM DB2 UDB for z/OS
Component Installation Guide: SAP Web Application Server ABAP 6.40 on Windows: IBM DB2 UDB for z/OS
Planning Guide: z/OS Configuration for SAP on DB2 UDB for z/OS
SAP Database Administration Guide for SAP on IBM DB2 UDB for z/OS
Homogeneous and Heterogeneous System Copy for SAP Systems Based on SAP Web Application Server ABAP 6.40 SR1
SAP NetWeaver 2004s **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.
Component Installation Guide: SAP NetWeaver 2004s ABAP on AIX: IBM DB2 UDB on z/OS
Component Installation Guide: SAP NetWeaver 2004s ABAP on Linux: IBM DB2 UDB on z/OS
Component Installation Guide: SAP NetWeaver 2004s ABAP on Windows: IBM DB2 UDB on z/OS
Homogeneous and Heterogeneous System Copy for SAP Systems Based on SAP NetWeaver 2004s
SAP Planning Guide for SAP NetWeaver for IBM DB2 UDB for z/OS (SAP NetWeaver 2004s SR1)
SAP Database Administration Guide for SAP NetWeaver on IBM DB2 UDB for z/OS (SAP NetWeaver 2004s SR1)
Other The following document is available in the SAP Service Marketplace under `http://service.sap.com/solutionmanagerbp`
SAP on IBM DB2 UDB for OS/390 and z/OS: Best Practice for Installing or Migrating to DB2 V8

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 53. Relevant SAP Notes

SAP Note	Title
81737	APAR List
83000	SAP DB2 Database Recovery Options
98051	Database Reconnect: Architecture and function
182207	DB2/390: Improving Virtual Storage Utilization
363189	DB2/390: Volume Copies Consistency
426863	DB2/390: DB Performance Monitor/IFI Data Collector
509529	DB2/390: Changing the DB2 host proactively

Table 53. Relevant SAP Notes (continued)

SAP Note	Title
524816	Standalone enqueue server
538081	High-availability SAPLICENSE
559996	High-availability and automation solution for DB2 on z/OS
684835	Availability of Rolling Kernel Upgrades for 4.6D_EXT Kernel
728743	zSeries: Release of DB2 V8 for SAP Components
809477	startsap and stopsap <= SAP WebAS 640
821904	Separating SCS instances for ABAP and J2EE
915482	DB2-z/OS: Automating DB failover

APARs

This section lists selected APARs that are referenced in this publication. See also SAP Note 81737.

Table 54. Relevant APARs

APAR	Description
PQ79387	DB2 V7 (see SAP Note 81737)
OW53313	NFS (see SAP Note 81737)
OW48503, OW51676	SA z/OS
OW53950	First ICLI with Linux on System z support
VM63282, VM63397	z/VM (see SAP Note 81737)
II11352	z/OS Release Matrix
PK01667	Option to propagate child locks prematurely

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 UDB 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

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

Readers' Comments — We'd Like to Hear from You

SAP on System z
High Availability for SAP on IBM System z
Using Autonomic Computing Technologies

Publication No. SC33-8206-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

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.

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
SC33-8206-01

Cut or Fold
Along Line

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 Entwicklung 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

Cut or Fold
Along Line

Printed in USA

SC33-8206-01