SAP High Availability (BC-CCM-HAV)

HELP.BCCCMHAV

Release 4.6C

Copyright

Icons

Icon	Meaning
⚠	Caution
💬	Example
💡	Note
🧭	Recommendation
SYN	Syntax

Contents

SAP High Availability (BC-CCM-HAV)

Purpose

As more and more customers deploy the R/3 System for their global operations to support mission-critical business functions such as sales and order-entry and continuous manufacturing, the need for maximized system availability becomes more and more crucial. Many companies now require 24 x 7 reliability for their R/3 Systems, that is, 24 hours a day, 7 days a week.

This documentation has been written to help you assess your current configuration and procedures and their implications on systems availability and to offer recommendations in formulating a high systems availability strategy for your R/3 System.

For up-to-date information on high availability in SAPNet, you can use the alias “ha.” Enter the following in the address line of your web browser:

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

Implementation Considerations

High availability is a technically complex area, and implementation considerations vary according to the nature of your system setup. This documentation is primarily intended to illustrate the available possibilities. Therefore, for detailed technical guidance when implementing a specific product or feature, contact the appropriate source, such as your SAP consultant, your hardware supplier, the SAP Competence Center, and so on.

To get started quickly with high availability, see High Availability Procedures at Your Site [Page 247], especially the General Checklist [Page 250]. The checklist directs you to the relevant parts of the documentation according to the characteristics of your system and other factors.

Features

An operational system (that is, a system in productive use) can be defined as "a system that can be used for its intended purpose". The term "operational" contains elements of performance as well as availability (that is, downtime), focusing on availability in the R/3 System, including the R/3 services, database management system (DBMS) services, network and operating system services, and hardware services. Availability must be discussed together with performance.

For more information, see the following:

- Availability and Performance [Page 8]
- Planned and Unplanned Downtime [Page 8]
- R/3 System Services [Page 11]

Constraints

Third-party products are used extensively in high availability solutions. Therefore, you need to make sure of compatibility with the R/3 System before implementing high availability products and features. For detailed technical guidance, contact the appropriate source, such as your SAP consultant, your DBMS supplier, the SAP Competence Center, and so on.

> This documentation covers R/3 Releases 3.0 and later. For more information on earlier releases, see previous versions of the documentation.

Availability and Performance

The two relevant aspects of an operational system are availability and performance:

- Availability

  A service is said to be available if it is capable of performing the task it is designed to perform. This is a "yes-no" concept, because the service is either available or unavailable.

  When considering unplanned downtime, this "yes-no" concept is related to a failure model called "crash failure". This is an approximation to real system failures, which are usually more complex.

- Performance

  This is measured by the ability to meet certain pre-defined criteria such as throughput of the system (for example, in number of users supported) and average response time for each user. Performance is said to be acceptable when a certain level has been achieved over a given period.

Performance clearly depends on availability. However, availability does not always guarantee performance. For example, an R/3 System where the CPU utilization of the database host is very high has low performance, but it could be considered as an available system. In practice, the distinction between the two aspects often blurs in extreme cases (that is, extremely poor performance means the system is considered to be unavailable or "down"). For instance, a network connection might be terminated (that is, be unavailable) after a time-out has occurred due to unacceptably slow performance. Even though this documentation focuses on the availability aspects of the system, the close relationship to the performance aspects should not be ignored.

To increase the availability of a system, it is essential to minimize downtime. Downtime can be categorized as planned and unplanned, and at a more detailed level, it can be separated into planned and unplanned downtimes for the R/3 services, DBMS services, network and operating system services and hardware services.

Planned and Unplanned Downtime

Planned Downtime

Planned downtime is the "time for scheduled maintenance during which a system cannot be used for normal productive operations". This time is used for a variety of purposes so that a system can function optimally and reliably.

Some of the possible causes of planned downtime are as follows:

- Hardware maintenance.

Refer to Hardware and System Software Key Issues [Page 129].

- Upgrades to new releases of R/3, DBMS, or operating system

  Refer to:

  – R/3 Upgrade [Page 34]
  – Upgrade with Oracle [Page 67]
  – Upgrade with Informix [Page 81]
  – Upgrade with SAP DB [Page 89]
  – Upgrade with DB2 UDB [Page 95]
  – Upgrade with DB2 for OS/390 [Page 105].

- DBMS reorganization

  Refer to:

  – Space Management With Oracle [Page 48]
  – Space Management with Informix [Page 71]
  – Space Management with SAP DB [Page 86]
  – Space Management with DB2 UDB [Page 91]
  – Space Management with DB2 for OS/390 [Page 100]

- DBMS backup

  Refer to:

  – Backup with Oracle [Page 60]
  – Backup with Informix [Page 69]
  – Backup with SAP DB [Page 83]
  – Archive and Backup with DB2 UDB [Page 91]
  – Backup with DB2 for OS/390 [Page 97]

- Archiving of R/3 business objects (logical backup)

  Refer to R/3 Level Data Archiving [Page 39].

Unplanned Downtime

Unplanned downtime is the "time during which a system cannot be used for normal productive operations due to unforeseen failure in hardware or software components".

Unplanned downtime can be extremely costly to an organization. The source of unplanned downtime can be in any of the layers that make up the complete software and hardware environment:

- Environment (for example, power)
- CPU
- Disk
- Memory

- Network (including all components such as hubs, routers, etc.)
- Network interface card (NIC)
- Operating system
- DBMS
- Application software (R/3) or user error (for example, deletion of database table)

R/3 System Key Issues

Purpose

This section describes in detail issues affecting downtime in key areas of the R/3 System. For each area, a discussion of the potential causes is followed by specific recommendations on how to reduce downtime.

To help you better manage prevention and recovery to minimize downtime, this section discusses the following important factors:

- Monitoring and tuning the system to avoid downtime
- Recovering from downtime

Implementation Considerations

The following table shows when you might find the information useful to improve the high availability of your R/3 System:

If you want to	Then refer to
Install an R/3 System	Mapping of R/3 System Services [Page 14]
Reduce failures in the R/3 System	R/3 System Failures [Page 27]
Upgrade an R/3 System	R/3 Upgrade [Page 34]
Improve R/3 archiving	R/3 Level Data Archiving [Page 39]
Improve the R/3 spool service	R/3 Spool Service [Page 40]
Convert and reorganize R/3 tables	R/3 Incremental Table Conversion [Page 42]
Update kernel but not applications	R/3 Downward-Compatible Kernel [Page 44]
Transport into production R/3 System	R/3 Transports [Page 45]

For detailed technical guidance when implementing a specific feature, contact the appropriate source, such as your SAP consultant, the Going Live and EarlyWatch [Page 169] services, and so on.

Integration

High availability for the R/3 System should be part of a system-wide strategy. Therefore, you should also consider other components of the system, such as the database management system (DBMS), the network, the hardware and operating system services, and so on.

R/3 System Classifications

Definition

We distinguish between the following kinds of R/3 System from the high availability viewpoint:

- Standard R/3 System

  The standard R/3 System already contains a range of possibilities to improve the availability of your R/3 System without purchasing additional software or hardware.

- High availability R/3 System

  A high availability R/3 System consists of additional hardware and software, chosen in consultation with your SAP partner. For example, you can use a cluster solution to protect the database or the enqueue service.

- Business continuity R/3 System

  A business continuity R/3 System involves a standby R/3 System, usually at a remote site. This enables you to recover from disasters such as the destruction of your primary R/3 System.

Use

The classification is useful when discussing high availability aspects of the R/3 System:

- Standard R/3 System

  You use a standard R/3 System when you have no special additional requirements for high availability.

- High availability R/3 System

  You use a high availability R/3 System when you need to ensure fault-tolerant operation of your R/3 System. A high availability R/3 System is robust without being completely disaster-proof.

- Business continuity R/3 System

  You use a business continuity R/3 System for critical operations when it is essential that the system is always available. In the event of a disaster, such as a fire at the primary site, you can resume operations immediately or quickly from the standby site.

R/3 System Services

Use

This documentation focuses on the notion of services. We view the R/3 System as a set of software and hardware components that are hierarchically assembled into the fully functioning system. The inter-relationship between the components of the R/3 System is of central importance for high availability. The individual components deliver services to one another. This means that, if one component fails, the effect is felt by other parts of the system that rely on the delivery of services from the failed component to complete their tasks, especially if a single point of failure is involved. If you do not provide a backup in the case of failure, unplanned system downtime results from failure.

R/3 System Services

How you protect R/3 System services depends on your overall approach to high availability, for example, whether you are using a standard R/3 System or a high availability R/3 System. For more information on the distinction, see R/3 System Classifications [Page 11]. In a standard configuration, certain services are unprotected (for example, enqueue, message, and database) whereas in a high availability configuration you can protect vulnerable services.

In a high availability R/3 System, you can replicate vulnerable services, such as the enqueue, message, and database services by using, for example, cluster solutions or switchover solutions. This protects such services against failure. For more information, see:

- Cluster Technology [Page 130]
- Microsoft Cluster Server on Windows NT [Page 220]
- Switchover Software [Page 216]
- Protection of the Enqueue Service [Page 229]

Prerequisites

There are the following types of logical R/3 service:

- Presentation
- Application
- Database

This section concentrates on the mapping of the application services, since the presentation and database services are usually fixed. We describe how you can distribute the R/3 application services – that is, the dialog, update, enqueue, background, message, gateway, and spool services – on the physical host machines that are used as R/3 application and DBMS hosts. This mapping process is part of the wider subject of R/3 System configuration (customization for business objectives, maximizing the use of available computing resources and so on). This section concentrates on mapping, since the wider configuration issues are covered elsewhere.

For more information, see R/3 Architecture and Logical Services [Page 13].

Features

The following aspects interrelate in the mapping of R/3 application services:

- High Availability

  This aspect looks at the resilience of a given mapping design in terms of system failures. For example, if there are at least two application hosts running dialog services and one of the application hosts fails, then the users are able to connect themselves to the second application host running dialog services. Therefore, having dialog services on different application hosts improves availability.

- Performance

  Two different mappings, where the same set of application services is mapped to a given hardware configuration, can have very different performance behavior. Refer to Inter-Service Communications [Page 17].

To map your R/3 Services, you also need to consider other specific circumstances of your installation, for example, the backup drives connected to an R/3 host machine.

Activities

The R/3 System offers great flexibility in its configuration. Proper mapping of the R/3 System services is essential to meet high availability requirements. This section discusses how you can assign logical R/3 services, especially application services, to physical host machines for high availability.

You can distribute the R/3 application services on the physical host machines that are used as R/3 application and DBMS hosts. This mapping process is part of the wider subject of R/3 System configuration (that is, customization for business objectives, maximizing the use of available computing resources, and so on). This section concentrates on mapping, since the wider configuration issues are covered elsewhere.

Example

A simple example of the idea of services is the central role played by hardware services in the R/3 System. If a disk drive containing data vital to system functioning fails, other aspects of the system are also threatened. In service terms, the hardware has failed to deliver the required service, which has effects on higher-level services (for example, R/3 services), possibly leading to system downtime.

R/3 Architecture and Logical Services

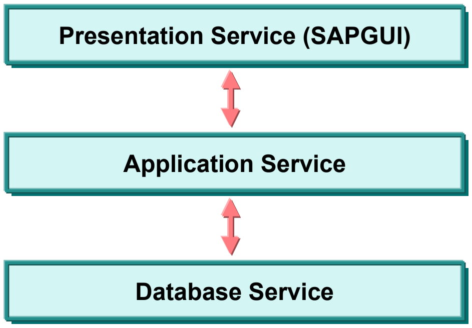
Definition

The R/3 System has a specific architecture with a specific set of logical services.

Structure

In the three-tier, client/server architecture of the R/3 System, there are three service layers:

R/3 System Service Layers



These are logical divisions of the services provided by an R/3 System, and the mapping process distributes these services across one or more physical host machines. The R/3 System services can be further logically divided, as described below:

R/3 System Logical Services

Service	Description	Layer
SAPGUI	The user interface module that implements input and output functions of the R/3 System	Presentation
Dialog	Processes the online user input from the SAPGUI and executes the application logic	Application
Update	Executes changes to the database asynchronously	Application
Enqueue	Supports R/3 logical lock management of business objects	Application
Background	Processes programs that require no interaction with users	Application
Message	Coordinates requests across R/3 System services within a single R/3 System	Application
Gateway	Handles the communication services between R/3 Systems and external systems such as R/2 and others	Application
Spool	Manages spool requests and formatting	Application
Database	Database services provided by the underlying DBMS	Database

The following discussion focuses mainly on the mapping of the R/3 application services.

The gateway service consists of several components. This documentation refers to the core process (known as `gwrd`) when discussing recovery, for example. Connections to particular external systems (for example, R/2) running under additional processes are not included in such discussions.

Mapping of R/3 System Services

Purpose

Mapping R/3 System services to host machines improves the availability of your R/3 System.

Prerequisites

Mapping Guidelines

The following guidelines, dictated by the R/3 architecture [Page 13], apply to the mapping of services:

- Most of the R/3 application services can be replicated (for example, there can be several host machines running several dialog work processes). In this sense, such replicated R/3 services form "virtual clusters."

- Some R/3 application services cannot be replicated and must be offered by only one host machine within the system (for example, enqueue service). Such services can only be provided by **one** instance of the service. Therefore, they are more susceptible to system failures because of the lack of redundancy. However, in a high availability R/3 System [Page 11], you can replicate such services.
- Multiple instances offering the same application service might or might not reside on the same host (for example, one host might have one dialog process and a second host might have five dialog processes).

For simplicity, we assume that there is only one so-called R/3 application instance on each application host machine (that is, no more than one R/3 dispatcher process for each application host machine).

Architecture Characteristics of R/3 System Services

The key characteristics of the services within a standard R/3 System are as follows:

Architecture Characteristics of R/3 System Services

		Number of services per R/3 System	Number of services per application host
R/3 Services	SAPGUI	>= 1	Not relevant
	Dialog	>= 2	>= 2
	Update	>= 1	>= 0
	Enqueue	1	0 or 1
	Background	>= 1	>= 0
	Message	1	0 or 1
	Gateway	>= 1	1
	Spool	>= 0	>= 0
	Database	1	0 or 1

Each application host running a dispatcher process has one gateway service.

In a high availability R/3 System, you can replicate vulnerable services, such as the enqueue, message, and database services by using, for example, cluster solutions or switchover solutions. This protects such services against failure. For more information, see:

- Cluster Technology [Page 130]
- Microsoft Cluster Server on Windows NT [Page 220]

- Switchover Software [Page 216]
- Protection of the Enqueue Service [Page 229]

Mappings Supported by R3SETUP

The R/3 System installation program, R3SETUP, supports the installation of the database service (DB), the central instance (CI) and dialog services (D). When you install R/3 on each host machine, you can choose one of the following installation options with R3SETUP:

DB + CI	Central instance with DBMS
CI	Central instance without DBMS
DB	Standalone database host
DI	Dialog instance host

From the service mapping standpoint, the key question is whether to install your R/3 System on the same host machine (that is, "DB + CI") or to have the DBMS and CI on different host machines. Additionally, you need to decide whether to install a separate dialog instance on one or more application hosts.

The options described above are the default installation options. Other mapping schemes are possible using custom procedures based on the specific application environment.

What is a central instance?

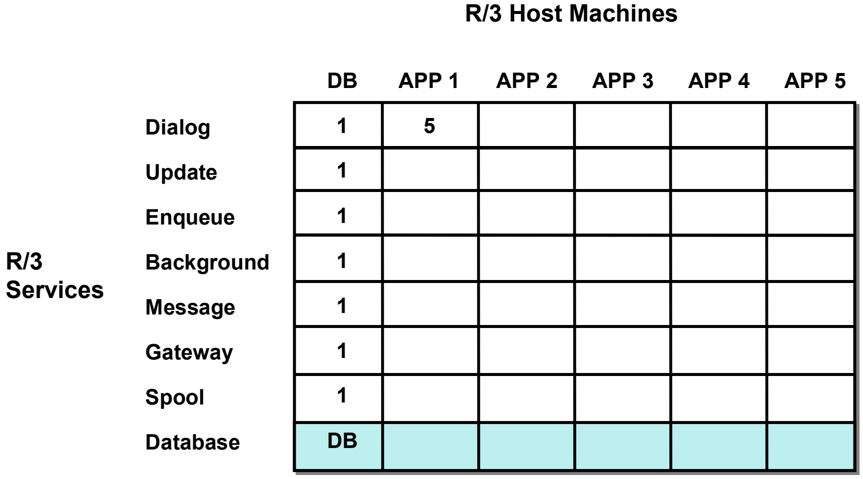
The central instance can be described in shorthand as "CI=DVEBMGS" (that is, **C**entral **I**nstance = **D**ialog, **V** for Update – stemming from the German "**V**erbucher" – **E**nqueue, **B**ackground, **M**essage, **G**ateway, **S**pool). The machine where the central instance runs is called the central host and **always** has at least the enqueue and message services. The CI usually has multiple dialog, update and background work processes. So references to the CI implicitly mean that multiple processes are involved.

Process Flow

1. You review the recommendations for mapping R/3 Services [Page 25].
2. You map your R/3 System services to host machines following the information in "Prerequisites" above and in Inter-Service Communications [Page 17].
3. You use the R/3 service mapping table – see example below – to record your chosen mapping.

This example maps a small installation consisting of two host machines:

R/3 Service Mapping Table: Small Installation Example



R/3 Host Machines

		DB	APP 1	APP 2	APP 3	APP 4	APP 5
R/3 Services	Dialog	1	5				
	Update	1					
	Enqueue	1					
	Background	1					
	Message	1					
	Gateway	1					
	Spool	1					
	Database	DB					

The above diagram shows an R/3 System with the following aspects:

- The DB host runs both the database service and the central instance.
- A separate application host runs five dialog services.

For simplicity the hardware configuration used consists of a maximum of six host machines, that is, one DB host machine and five R/3 application hosts. The only reason for using five application hosts is to keep the table and the discussion simple. This also helps you in mapping the dialog service groups.

R/3 mapping (as shown in the tables used in this section) depends heavily on your actual hardware configuration. Therefore, the configurations given are examples only. Also, for proper mapping, it is necessary to distinguish between a single processor and a multi-processor SMP application host. The example above assumes that the hosts are single processor hosts. For more details on the differing behavior of single processor and multi-processor SMP machines in R/3 Systems, contact your hardware vendor.

Result

You have mapped the services of your R/3 System to host machines. Some typical mapping approaches are shown as follows:

- Mapping of a Central System [Page 19]
- Mapping of a Central System with Additional Dialog Hosts [Page 20]
- Mapping of a Standalone DBMS [Page 22]
- Mapping of a Custom Setup [Page 23]

Inter-Service Communications

Definition

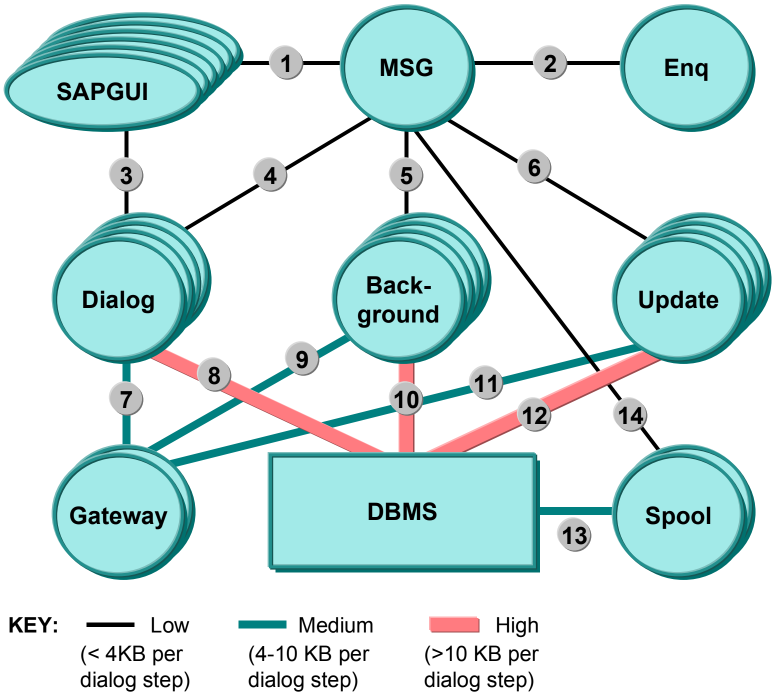
The communications between the R/3 System services [Page 11] influence the way you map the services to host machines.

Inter-Service Communications

Structure

The diagram below shows the R/3 System services and the communication traffic between them (assuming, for simplicity, that the services are on different host machines). The convention used here assumes a one-to-one relationship between any one service and another. The cumulative traffic is not shown on this diagram. For example, the traffic between the message service and the dialog service assumes a one-to-one relationship between these services, whereas in reality there are often multiple dialog hosts offering dialog services and communicating with one message service. The number of hosts offering specific services is different for each installation, and the cumulative traffic between multiple services varies accordingly.

Communications between R/3 System Services


SAPGUI
1
MSG
2
Enq
3
4
5
6
Dialog
Back-ground
Update
7
8
9
10
11
12
14
Gateway
DBMS
13
Spool
KEY:
Low (< 4KB per dialog step)
Medium (4-10 KB per dialog step)
High (>10 KB per dialog step)

The above graphic relates to a **standard** R/3 System. In a high availability R/3 System, you can add software and hardware components to replicate the vulnerable services. These services are the message, enqueue, and DBMS, which are not replicated in a standard R/3 System.

For more information about the difference between a standard and a high availability R/3 System, see R/3 System Classifications [Page 11].

Integration

From this diagram, you can see that:

- Paths 1 and 3

  For logon processing, the SAPGUI first connects to the message service (path 1) and is then allocated a dialog host (path 3).

- Paths 4 and 2, or paths 5 and 2

  When an update is made, the dialog or background service sends an enqueue request message along paths 4 and 2 or paths 5 and 2 respectively.

- Paths 4 and 6, or paths 5 and 6

  When an asynchronous update is made, the dialog or background service sends an update request message along paths 4 and 6 or paths 5 and 6 respectively.

- Paths 7, 9, or 11

  An RFC request is made to the gateway service along one of these paths.

- Paths 8, 10, 12, or 13

  A request for database services is made to the DBMS using remote SQL along one of these paths.

- Path 14

  When a print request has been generated by the dialog, background or update services, a “wake-up” message is also sent by these services to the spool service.

Mapping of a Central System

Definition

When you map your R/3 System as a central system, all R/3 application services and the DBMS services are mapped onto a single host (that is, one physical hardware machine).

The features of a central system are as follows:

- Ease of administration

  Since there is only one host machine for the R/3 application services and the DBMS service, ease of administration and configuration are two of the chief benefits of a central system.

- High availability

  A central system depends crucially on the availability of one host machine. SAP recommends that you use a standby host (with dual-ported shared disks) to protect the R/3 System from failure of the central host machine. This allows the system to continue operating by taking over all R/3 System services in the event of primary host machine failure. You can then use Switchover Software [Page 216] to automatically transfer system operation to the standby host.

  The use of a single standby host to improve R/3 System availability also brings administrative benefits since only one extra machine needs to be duplicated and administered.

- Performance

  Generally, a central system can be recommended as long as the host has been adequately sized for all the R/3 System services. A central system has less communications overhead across the network than mappings that contain application services mapped onto multiple application hosts.

Note that a central system is scaleable through the use of Symmetric Multiprocessor (SMP) host machines.

Use

Depending on whether there is sufficient spare capacity and bearing in mind possible future expansion of the business, it might make sense to use a central system. Typically, small R/3 installations run on a central system, although there are exceptions to this rule.

Small installation

Consider a standby host machine for a central system. Refer to Switchover Software [Page 216].

Example

An example of an R/3 service mapping table for a central system looks as follows:

R/3 Service Mapping Table for Central System

		R/3 Host Machines					
		DB	APP 1	APP 2	APP 3	APP 4	APP 5
R/3 Services	Dialog	7					
	Update	2					
	Enqueue	1					
	Background	3					
	Message	1					
	Gateway	1					
	Spool	1					
	Database	DB					

Mapping of a Central System with Additional Dialog Hosts

Definition

You can install additional dialog hosts when you map your R/3 System by using the R3SETUP option, “Dialog instance host.” For more information on R3SETUP mapping options, see Mapping of R/3 System Services [Page 14].

The particular features of a central system with additional dialog hosts are as follows:

- High availability of DB host machine

  The R/3 System depends on the availability of the DB host running DB and CI. So it is important here to consider the use of a standby host. Refer to Switchover Software [Page 216].

- High availability of dialog host machines

  If there are enough dialog hosts to maintain adequate performance after the failure of one dialog host machine, it should not be necessary to introduce an additional standby dialog host – even for an R/3 customer with strong high availability needs in the unplanned downtime area. This is because the dialog service already forms a virtual cluster, which is resistant to failure of single instances. For more information, see section “Logon Load Balancing” in R/3 System Failures [Page 27].

- Only one standby host needed

  The main host in this configuration (that is, the DB host) requires only a single standby host to improve R/3 System availability. As already shown in the case of a central system, the use of a single standby host to improve R/3 System availability also brings administrative benefits since only one extra machine needs to be duplicated and administered. The standby host could be either one of the existing additional dialog hosts or an entirely new host machine.

- Performance

  Mapping the dialog services onto separate hosts from the DB host frees the central host from the burden of performing dialog services. This is of key importance for the scalability of the R/3 System as more users can be supported simply by introducing additional dialog hosts.

Use

Typically, small to medium sized R/3 installations run on a central system with additional dialog hosts.

Small to medium installations

Consider a standby host machine for a central system with additional dialog hosts. Refer to Switchover Software [Page 216].

Example

An example of an R/3 service mapping table using this approach looks as follows:

R/3 Service Mapping Table for Central System with Additional Dialog Hosts

R/3 Host Machines

R/3 Services		DB	APP 1	APP 2	APP 3	APP 4	APP 5
	Dialog	2	7	7	7	7	7
	Update	4					
	Enqueue	1					
	Background	3					
	Message	1					
	Gateway	1					
	Spool	1					
	Database	DB					

To increase the number of users supported by an R/3 System, additional dialog hosts can be used, as long as you have enough CPU cycles on the DB host to support the additional dialog hosts.

Mapping of a Standalone DBMS

Definition

The main reason for introducing a standalone DBMS host, (rather than having the DBMS and CI on the same host) when mapping the R/3 System is to save CPU cycles on the DB host by mapping CI and dialog services to other host machines.

The particular features of this mapping are as follows:

- High availability of DB host machine

  The R/3 System depends on the availability of the DB host running the DBMS. So it is important to consider the use of a standby DB host. Refer to Switchover Software [Page 216]. If the primary DB host machine fails, the standby DB host takes over the DBMS. The standby host can also be used in normal operation for R/3 applications services.

- High availability of CI host machine

  The R/3 System also depends on the availability of the CI host (APP 1 in the example below). So it is important to consider the use of a standby CI host.

Use

Typical medium to large sized R/3 installations run a standalone DBMS host, although there are exceptions to this rule. The recommendations for improving availability with a standalone DBMS vary according to the size of your installation:

Medium to large installations

You should consider using the DBMS and CI hosts as mutual standby hosts in medium-sized installations (that is, the DBMS host is the standby host for the CI host and vice versa) if you wish to improve R/3 System availability. Refer to Switchover

Software [Page 216]. In larger installations, you could consider having separate standby hosts for the DBMS and CI.

Medium to large installations

You should consider using a single standby host for the DBMS host and the CI host if you wish to improve R/3 System availability in medium to large installations. Refer to Switchover Software [Page 216].

Large installations

If saving CPU cycles on the DB host is of concern in large installations, then SAP recommends that you evaluate the use of two separate standby host machines for the DB host and the CI host. Refer to Switchover Software [Page 216].

Example

An example of an R/3 service mapping table using this approach looks as follows:

R/3 Service Mapping Table for a Standalone DBMS Host

R/3 Host Machines

		DB	APP 1	APP 2	APP 3	APP 4	APP 5
R/3 Services	Dialog		2	7	7	7	7
	Update		4				
	Enqueue		1				
	Background		3				
	Message		1				
	Gateway		1				
	Spool		1				
	Database	DB					

Mapping of a Custom Setup

Definition

A custom setup for the mapping of R/3 System services to host machines is one that you design yourself.

Custom mappings are **not** supported by the R/3 System installation tool R3SETUP.

Mapping of a Custom Setup

Use

If you use more than one application host, the flexibility for mapping the R/3 application services increases greatly. This approach is mostly used in medium to large sized installations. Make sure that your mapping conforms to the information in Mapping of R/3 System Services [Page 14].

There are pros and cons for distributing application services from the high availability and performance perspectives:

- High availability of application services

  Distributing dialog, update, background, gateway and spool services across multiple hosts increases the availability of these services and protects the services from hardware failures of any one host. Refer to R/3 System Failures [Page 27].

  For example, with multiple dialog hosts, R/3 users can select at connect time which dialog service groups to attach to; further, within a selected service group, users connect to dialog services based on the availability of dialog services and the load of dialog processes within a given group. Refer to section “Logon Load Balancing” in R/3 System Failures [Page 27].

- Performance

  An application host running only specific application services can be better tuned to support the specific processing characteristics of that service.

  However, you must plan carefully to make sure that the communications bandwidth between the different services does not become a bottleneck, especially where update processes have been distributed.

Example

An example of an R/3 service mapping using a custom mapping approach looks as follows:

R/3 Service Mapping Table for Custom Mapping with Additional Updates

		R/3 Host Machines					
		DB	**APP 1**	**APP 2**	**APP 3**	**APP 4**	**APP 5**
R/3 Services	**Dialog**		2	5	5	5	5
	Update		2	2	2	2	2
	Enqueue		1				
	Background		4				
	Message		1				
	Gateway		1				
	Spool		1				
	Database	DB					

If you have an update intensive application, you could consider mapping update processes onto one or more dedicated application hosts that support additional update services. This frees CPU cycles on the DB host.

Mapping R/3 System Services using SAP Recommendations

Use

When mapping your R/3 System [Page 14], the reduction of single points of failure is very important in improving availability but how you achieve this depends on the type of service you are trying to protect. Therefore, keep in mind the following generalizations when reading the recommendations below:

- Consolidate critical services (that is, DBMS, enqueue, and message)

  Since each R/3 System has just one DBMS, enqueue, and message service, locate these critical services on the most reliable machines. Protect their host machines by standby hosts.

- Distribute non-critical services to form virtual clusters

  For services that can be offered by multiple host machines, such as any of the R/3 application services except the enqueue and the message services, mapping the services onto multiple hosts can provide redundancy by forming failure-resistant virtual clusters.

Prerequisites

The recommendations below are grouped into those primarily for high availability and those primarily for performance. You need to balance high availability with performance depending on the requirements of your system setup.

Procedure

1. Review the following **high availability** recommendations and choose which ones you want to implement.

   Small installations

   All services on one machine with a standby machine

   Locate the database and all application services on a single powerful and reliable machine. You can then configure an identical standby machine to improve system availability. This simplifies administration since only one machine needs to be maintained.

   Medium to large installations

   Use most reliable host(s) for critical services

   The database, enqueue and message services should be mapped to run on the most reliable host(s), and the host(s) supporting these critical services should be protected by standby hosts.

   Medium to Large installations

   Install dialog services on multiple hosts

Set up at least two dialog hosts to improve the availability of dialog services in the event that one host fails.

Medium to Large installations

Logical dialog service groups for load balancing

Establish logical groups of dialog services and assign SAPGUI clients to appropriate groups based on usage patterns. For example, a group of dialog services can be defined for the sales and distribution applications and other groups can be set up for human resources and financial applications. While the assignment of users to groups of dialogue services is performed statically by the administrator, users connecting to a given group are dynamically allocated by the system to different host machines based on predetermined parameters. The message service dynamically allocates users across physical machines using administrator-specified load balancing algorithms based on response time and maximum number of users. Refer to “Logon Load Balancing” in R/3 System Failures [Page 27].

2. Review the following **performance** recommendations and choose which ones you want to implement:

Separate communications-intensive services if possible

The database, message, gateway and enqueue services should not all be located on the same host machine, because each of these services are communications-intensive and their resource usage patterns compete with one another.

Check possible separation of message/enqueue from database services

This frees up CPU cycles on the DB host but, on the other hand, tends to reduce high availability. This conflicts with a recommendation made above (“Use most reliable host for critical services”), where it was suggested to put database, enqueue and message services on a single highly reliable machine (usually the DB host).

Keep message and enqueue service on same host

Message and enqueue services should be located on the same host machine for performance reasons, because the enqueue service communicates only with the message service.

Database and update services on same host

SAP recommends in general that you set up the update services on the database host. This makes sure that there is a short communications path between two communications-intensive services, the database and the update services. Since the CPU load generated by an update work process is comparable to the induced DBMS CPU load, mapping DBMS and update services onto different hosts does not free many CPU cycles on the DB host.

Only for systems with high CPU load on the DBMS does it make sense to separate the DBMS and the update services.

FDDI connection if database and update separate

There are installations where the competition for CPU resources between the database and update services is so great that it becomes necessary to distribute the update processes to a separate host dedicated to update services. An FDDI based network connection can be used to provide adequate communication bandwidth between the database service and the update service(s) located on another instance.

WANs can be used for low volume communication links

The light communication traffic between SAPGUI and the message (only at connect establishment time) and dialog services makes the connection between SAPGUI and these services suitable for access across WANs. Refer to Network System Key Issues [Page 108].

R/3 System Failures

This section classifies service failures in the R/3 System in terms of high availability.

The term R/3 service refers to “a logical component that performs a specific kind of task for the R/3 system.”

The first part of this section looks at what constitutes failure in the R/3 System. The second part looks at each of the R/3 services in terms of high availability.

What is R/3 System Failure?

This section discusses what constitutes failure of the R/3 System in general terms. For a more specific discussion of how individual services fail, see the section “R/3 Service Failures” below.

Standard Failures

The following factors leading to failure are common to all services:

- Hardware

  Hardware includes central processing unit (CPU), memory, network interface card (NIC), and so on. The different kinds of service might reside on physically different hardware so the failure of a single machine can affect one or more R/3 service(s). This is a common cause of failure.

- Operating system services

  R/3 services depend in turn on operating system services. If operating system services fail, then so does the R/3 service. An example of an operating system service is the socket layer services, the failure of which affects the R/3 message service.

- Software

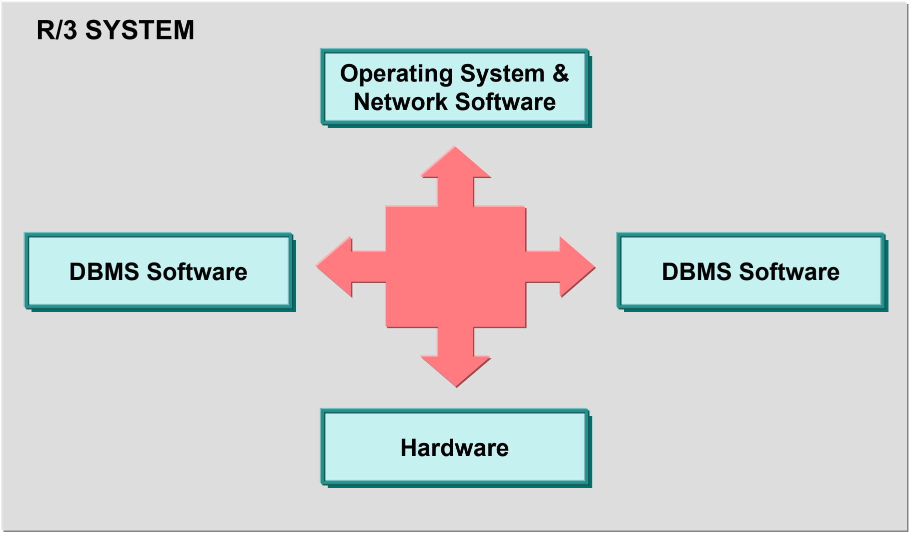
R/3 System Failures

As with any software, programming errors can lead to failure of an R/3 service.

Simple Failure Classification

The four categories in the R/3 System for classifying failures are R/3 services, database management system (DBMS) services, operating system and network services, and hardware services.

R/3 System: Failure Categories



When thinking about fault-tolerance of an R/3 System, it is useful to hierarchically break the system down into different services (the granularity should not be too small) according to the categories shown above. You can use the following table, showing the hierarchy of services, to discuss failures in your R/3 System (note the distinction between CPU+ and hard disk in the DBMS layer, due to the central importance of the disks for the database data):

Failure of R/3 System Components: Hardware Services

HW Service	Failure of Component			
	HW / OS / Network		DBMS	R/3
Power	Power Supply			
Presentation Layer	Host machine (e.g. PC)			
Application Layer	Host machine			
DBMS Layer	Host machine			
	CPU +	Hard disk		
Server Network	LAN Card (DB Host)	LAN Card (Appl. Host)		
	LAN Media & Active Components			
Access Network	LAN Card (Appl. Host)	LAN Card (PC)		
	LAN Media & Active Components	Router		

Failure of R/3 System Components: Software Services

SW Service	Failure of Component				
Presentation Layer	Operating System		SAPGUI		
Application Layer	Operating System	Remote SQL	Dispatcher	Dialog WP	Update WP
			Enqueue WP	Batch WP	Message
			Gateway	Spool WP	
DBMS Layer	Operating System	DBMS Software			

Single Points of Failure

The DBMS, enqueue, and message services in a **standard** R/3 System cannot be made redundant by configuring multiple instances of them on different host machines. The remaining services (such as dialog, update, background, gateway and spool) can all be configured redundantly (in other words, on multiple host machines) to provide improved availability.

In a high availability R/3 System, you can replicate vulnerable services, such as the enqueue, message, and database services by using, for example, cluster solutions or switchover solutions. This protects such services against failure. For more information, see:

- Cluster Technology [Page 130]
- Microsoft Cluster Server on Windows NT [Page 220]
- Switchover Software [Page 216]
- Protection of the Enqueue Service [Page 229]

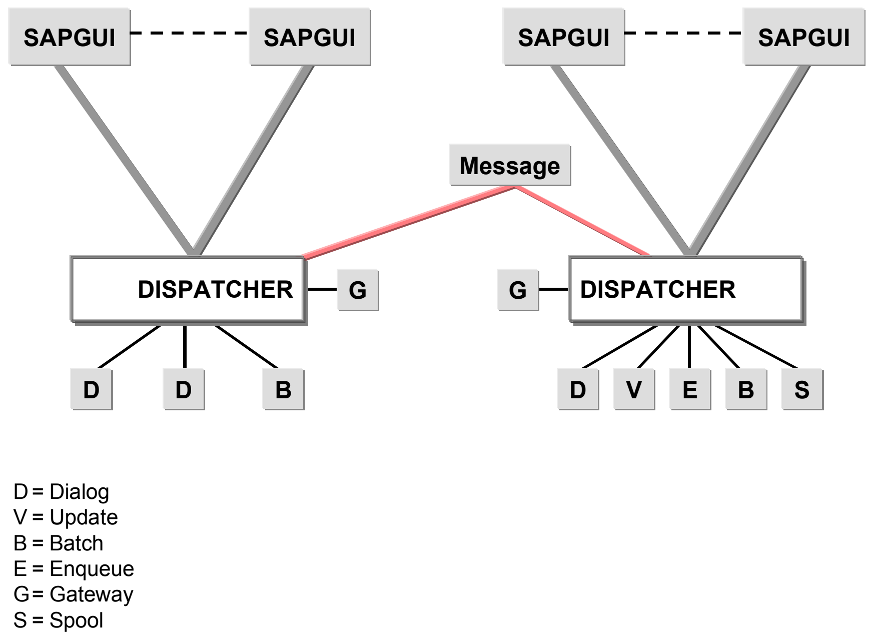
In an R/3 installation, Network File System (NFS) (for UNIX-based application hosts) and shares (for Microsoft NT-based applications hosts) are single points of failure. Some installations use an Internet Domain Name Service (DNS). DNS is also a single point of failure.

Automatic Recovery of R/3 Processes

Most R/3 services have the ability to reconnect dynamically if the corresponding process is restarted. The reconnect feature offers automatic recovery from many problems that affect only individual processes.

Most R/3 processes are restarted automatically by the dispatcher after process failure: this includes all the work processes (dialog, update, enqueue, background, spool) and the gateway process. Only the processes of the SAPGUI, message and dispatcher services are not restarted automatically. The diagram below illustrates how the connections between R/3 processes are recovered after failure (an explanation follows the diagram):

Recovery of Connections Between R/3 Services



- Connections denoted with thick line

These can only be recovered manually by restarting the SAPGUI. The user context is preserved on the application host.

- Connections denoted with medium line

  These are recovered by the automatic reconnect of the dispatcher to the message service. If the message service process fails, the process has to be restarted manually. For an automatic restart of this process, consider using Switchover Software [Page 216].

- Connections denoted with thin line

  These are automatically recovered if the dispatcher restarts one of the processes (dialog, update, enqueue, background, spool, gateway) after a failure.

Logon Load Balancing

R/3 logon load balancing allows application host machines in an R/3 System environment to be used more efficiently. Users can be dynamically distributed at logon time over several application hosts.

You can make a particular group of application hosts available for certain workgroups or certain tasks. An appropriate host is then automatically selected when a user logs on. A user is automatically logged on to the host with the best performance and the fewest users.

Logon groups are installed and maintained centrally in the R/3 System. You can define a maximum response time for each application host, and a maximum number of users that can log on to that host. This means that you can set up logon groups with improved response time behavior for important workgroups with time-critical transactions.

Logon load balancing protects the user from R/3 System failure involving dialog services by preventing logon to a host on which the dialog service has failed.

See also:

Logon Load Balancing [Ext.]

R/3 Service Failures

This section describes how to detect failure, what are the effects of failure and how to recover from failure of R/3 services. Note that the R/3 service processes can fail due to any one of the common causes outlined above in “Standard Failures” at the start of this section.

R/3 Service Failures: SAPGUI, Dispatcher, Dialog, Update, and Enqueue

R/3 System Failures

Failure	Detection	Effects	Recovery
SAPGUI	● By the user and dispatcher	● The sessions of one user will be aborted	● Manual restart ● SAPGUI Reconnect can be used
Dispatcher	● No automatic detection (except service manager on NT)	● Application service fails, all users are logged out ● If dispatcher is on enqueue host, then entire R/3 System affected	● Manual restart of application service ● If dispatcher is on enqueue host, all other application services must be restarted manually (see note below)
Dialog	● By the dispatcher ● A Syslog entry is created	● User session of one user can be aborted	● Automatic restart by the dispatcher
Update	● By the dispatcher ● A Syslog entry is created	● Possibly one aborted update	● Automatic restart by the dispatcher ● If update aborted, express mail sent to user
Enqueue	● By the dispatcher (see note below) ● A Syslog entry is created	● Temporarily not possible to obtain enqueues from application hosts other than the enqueue host	● Automatic restart by enqueue host dispatcher ● If enqueue host is restarted manually, transactions on other application hosts are automatically restarted

Starting with R/3 Release 3.0E, SAP provides extended functionality that enables user transactions to be aborted, without having to restart the entire application service. This extended functionality improves R/3 System performance following failure, since application data stored in memory is not deleted.

Starting with R/3 Release 4.0B, this functionality has been further extended. SAP now provides fully transparent support for transaction resets in the event of enqueue service failure. See "Enqueue Service Failure" below.

Application hosts (that is, where there is no enqueue service) are informed about the failure of the enqueue service by means of message service/dispatcher communication.

R/3 Service Failures: Background, Message, Gateway, Spool, and Database

Failure	Detection	Effects	Recovery
Background	● By the dispatcher ● Syslog and joblog entries are created	● A single batch job might fail	● Automatic restart of WP by dispatcher
Message	● By the dispatchers on all application hosts ● A Syslog entry is created	● Until message service is restarted, the entire R/3 System is affected (for example, requests cannot be executed for dialog, update & enqueue hosts)	● Manual restart
Gateway	● By the dispatcher ● A Syslog entry is created	● Temporarily no RFC and other external communications possible	● Automatic restart of WP by dispatcher
Spool	● By the dispatcher ● Syslog entries are created	● A single print job fails ● Failed print job restarted	● Automatic restart of WP by dispatcher
Database	● No automatic detection ● Syslog entries show database errors	● Entire R/3 System is affected ● No SQL requests to database are possible	● Manual restart (and recovery if necessary) of DBMS

Consider using Switchover Software [Page 216]

You can use switchover software for the DBMS host and the application host on which the enqueue and message services are running. Automatic failure detection and automatic restart (on a standby host) are thereby ensured.

For more information about using switchover to protect the enqueue service, see Protection of the Enqueue Service [Page 229].

Enqueue Service Failure

In a standard R/3 System, the enqueue table is held at a single location. If the enqueue service fails – for example, if the host supporting it crashes – then the information in the table is lost. This means that, when the enqueue service is restarted, all transactions that obtained locks from the enqueue table must be restarted.

Starting with R/3 Release 4.5A, application services constantly track the enqueue table. When a new, initialized version of the enqueue table appears following failure, all relevant transactions are automatically restarted. This feature is relevant to all standard, distributed R/3 Systems but especially to high availability R/3 Systems using switchover software. It has the following advantages:

- After enqueue service failure:

- You no longer need to restart the hosts supporting application services, so saving time and effort.
- All local R/3 data caches are preserved and the performance of application servers remains at optimal levels for all users.

- Switchover administration and switchover scripts become much simpler. For more general information about switchover, see Switchover Software [Page 216].

For more information, see Protection of the Enqueue Service [Page 229].

R/3 Upgrade

Use

This section describes R/3 software upgrades. An upgrade updates an existing R/3 System to a new release. The upgrade contains program code and data changes or introduces new areas of functionality. New R/3 Releases are issued periodically and are delivered as independent units that you must apply sequentially.

Integration

R/3 administrators must normally deal not only with new releases of R/3 software but also with upgrades to the operating system (OS) and database management system (DBMS). However, OS and DBMS upgrades are **not** discussed here.

An upgrade has to take into account the data in the customer system and other dependencies on external resources. A full upgrade consists of the following sections:

- Operating system upgrade
- Database upgrade
- SAP System upgrade

The interaction of these areas makes upgrading an R/3 System a complex task.

Prerequisites

- SAP aims to reduce the downtime during an upgrade, which depends on the:
  - Hardware you are using – this is the most important factor
  - Release **from** which you are upgrading and the release **to** which you are upgrading
  - Size of your database
  - Applications used

  You can now perform preparations – using the `PREPARE` program – and post-installation activities while the R/3 System is online, so reducing downtime.

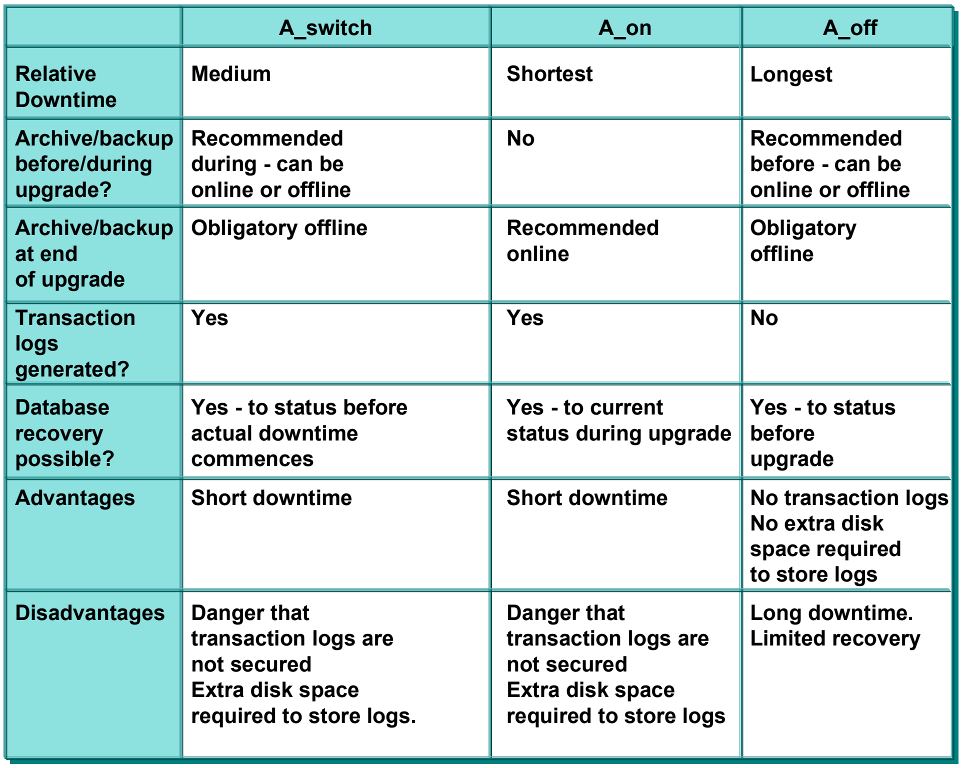
- It is sometimes necessary to upgrade in steps rather than directly. For example, to upgrade from 3.0E to 4.6B, you must first upgrade to 3.1I and then to 4.6B.

Features

All upgrades now use the repository switch upgrade method, which greatly reduces downtime compared to previous methods.

For more information on upgrade strategies, you **must** read the current SAP upgrade documentation **before** performing an upgrade.

Comparison of 4.x Upgrade Strategies



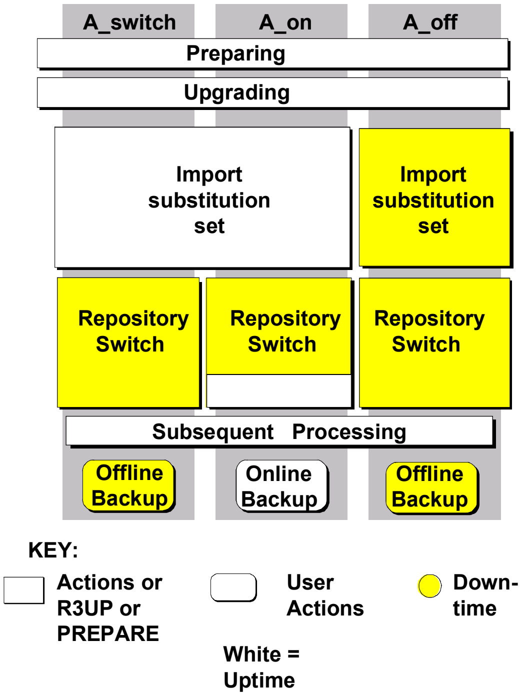
	A_switch	A_on	A_off
Relative Downtime	Medium	Shortest	Longest
Archive/backup before/during upgrade?	Recommended during - can be online or offline	No	Recommended before - can be online or offline
Archive/backup at end of upgrade	Obligatory offline	Recommended online	Obligatory offline
Transaction logs generated?	Yes	Yes	No
Database recovery possible?	Yes - to status before actual downtime commences	Yes - to current status during upgrade	Yes - to status before upgrade
Advantages	Short downtime	Short downtime	No transaction logs No extra disk space required to store logs
Disadvantages	Danger that transaction logs are not secured Extra disk space required to store logs.	Danger that transaction logs are not secured Extra disk space required to store logs	Long downtime. Limited recovery

On DB2 for OS/390, database actions occurring during the upgrade are saved by database mechanisms. Therefore, logging is always on regardless of the upgrade strategy. This lets you recover the database to the current status during the entire upgrade.

Therefore, for DB2 for OS/390, the upgrade strategy influences downtime but has no effect on database logging.

Starting with the upgrade to 4.6B, the application instance can run on OS/390, so that the overhead of network transmission is reduced.

Processing Sequence for 4.x Upgrade Strategies



Activities

SAP makes the following recommendations for upgrades:

Rehearse the upgrade in an appropriate test system

Whichever upgrade you are performing, the best recommendation of all is to rehearse it using a test system, that is, either a copy of your production system or a separately created and maintained development system. The test system should preferably have the same release and extent of modifications as the production system. This also gives you the advantage outlined in the next recommendation.

Automatic incorporation of customizations from test system

Customers with test systems that can be used to stage upgrades prior to upgrading the production system have the benefit that all customizations performed on the test system can be automatically imported into the production system. The interrupt-free integration of test system adjustments is a new feature available with Release 3.0.

Plan the upgrade carefully

Whichever upgrade you are performing, take the time to plan it carefully. This can be a laborious process but it helps to anticipate problems and avoid unplanned downtime.

Use the `PREPARE` program before the upgrade

From R/3 Release 4.0, you can prepare for the upgrade using the `PREPARE` program while the R/3 System is online, so reducing downtime. If necessary, you can use the `PREPARE` modules several times.

By using **all** `PREPARE` modules, including the optional ones, you can get precise information to help you with the modification adjustment using the transactions `SPDD` and `SPAU`. Also, `PREPARE` provides information to help you reduce the number of conversion errors due to lack of disk space during an upgrade

Accept automatic modifications for SAP objects in your live system

When you upgrade your live or quality assurance system, you can accept modifications that have been made in your development system (assuming the development system is equivalent to your live or quality assurance system). This avoids the manual effort of performing this process with transactions `SPDD` and `SPAU`.

Use strategy *A_on* for shortest possible downtime

A_on offers the best way to minimize downtime if you are upgrading to Release 4.x.

Create an archive (Informix) or backup (Oracle) after upgrade

Always create a complete archive (Informix) or offline backup (Oracle) immediately after a successful upgrade if transaction logging has been turned off during the upgrade (this applies to upgrade strategies *A_off* or *A_switch*).

Otherwise – that is, if transaction logging has not been turned off, with strategy *A_on* – an online backup is acceptable after an upgrade.

Set import destination time

You have the option of setting an import destination time at the beginning of the upgrade if you are using strategies *A_on* or *A_switch*. This enables you to optimize the extra system load due to the upgrade by “stretching” the import time for the new repository.

Increase number of parallel background processes for import

On multi-processor machines with sufficient main storage, you can reduce downtime by specifying up to four import processes to run in parallel.

Monitor the upgrade continuously

From Release 4.0, you can remotely (therefore, continuously) monitor the upgrade using the Upgrade Assistant. This allows you to avoid unnecessary downtime because you can react immediately if the upgrade program stops for any reason.

Consider using striped disks

Using striped disks can reduce the bottleneck problem caused by multiple importers competing for I/O resources. Note that this is a general approach to solving I/O bottleneck problems and is not specific to the R/3 System.

For more information about striped disks in the context of redundant arrays of independent disks (RAID), see Disk Technology [Page 132].

Consider using incremental table conversion

You can reduce downtime during the upgrade by performing table conversions incrementally, that is, during production operation and before you start the upgrade. For more information, see R/3 Incremental Table Conversion [Page 42].

Choose host names correctly if you use switchover software

If you use switchover software, see SAP Note 96317, which describes problems with host names.

Customer-based upgrades

You can further reduce downtime for upgrades to 4.6B systems by using this SAP Service. SAP offers customers individual upgrades on site that include SAP Standard, Add-On products, customer development, and modifications. There is no activation necessary and import time is reduced to a minimum.

For more information on this SAP Service send an email to upgradecc@sap.com.

See also:

Upgrade with Oracle [Page 67]

Upgrade with Informix [Page 81]

Upgrade with SAP DB [Page 89]

Upgrade with DB2 UDB [Page 95]

Upgrade with DB2 for OS/390 [Page 105]

Upgrade documentation issued by SAP

R/3 Level Data Archiving

Use

R/3 business objects are archived regularly. This level of archiving is distinct from backing up the underlying DBMS, therefore, separate high availability considerations apply. Since archiving consumes substantial system resources, you should plan it carefully with high availability in mind so as to minimize downtime and, from the performance perspective, system load.

Features

The archive process differs according to whether or not Archive Development Kit (ADK) is used. ADK can be either used with release 3.0A (or later) or release 4.0A (or later): A restart option is available with ADK. The archiving process is now performed in the following phases:

- All objects are read from the database and written to the archive.
- All objects are read from the archive to verify the successful completion of the first phase. After each object has been read, the corresponding object is deleted from the database.

The newer process (ADK archiving) can recover from operational failures, either in the database or in the archive files. Since no data can be lost during archiving, online archiving (without downtime) is now possible.

ADK can be used for specific applications in your R/3 System (if you are running a suitable release) and need not necessarily be used for all applications in a system.

Activities

Some of the recommendations below are **only** suitable for ADK (they are marked “ADK”).

Is database reorganization really necessary after an archiving session?

Archiving is often followed by database reorganization to reclaim the disk space freed by archiving. Since reorganization implies system downtime, you should carefully consider whether this is really necessary. For example, if a business application generates 100,000 business documents a week and they are archived on a weekly basis, nothing is gained by reorganizing after each archiving session because the released disk space is reused every week.

No backup is required before an archiving session – ADK

A backup (possibly implying downtime) is usually not necessary before archiving because an archiving session can be restarted without data loss if it has been unexpectedly interrupted.

Dialog sessions can continue during archiving – ADK

In general, archiving sessions can be run in parallel to dialog sessions. It is normally not necessary to incur downtime by stopping dialog sessions to archive data.

The archiving program cannot be used to delete data from the database – ADK

ADK is implemented using two independent programs, one that performs the archiving and one that performs the deleting. If you amend the archiving program, you should never include “delete” steps (these should always be performed in the separate delete program to avoid loss of data).

See also:

Application Data Archiving [Ext.]

R/3 Spool Service

Use

This section describes the possible reasons for failure in the R/3 spool service and describes how you can increase its availability.

Features

Each of the following components is involved in the R/3 spool service and each can fail:

- Spool work process (this prepares the output data and processes the print job)
- Database service and file system of operating system
- Number range service
- Transfer program (SAPLPD).
- Destination host
- Host spool system
- Output device (for example, printer)

The spool service does not work with transactions, as happens in the database service. Therefore, if a printout is incomplete due to a failure in the output device, the printout is not rolled back in the same way that database transactions can be rolled back. Instead, such a spool request is in most cases marked as “completed” in R/3.

The spool service [Ext.] needs the database service [Page 45] to function. Control information for the spool requests and, if so configured, the output print data itself, are stored in database tables. Output print data is stored in the Temporary Sequential Objects (TemSe) database. To improve system performance and reduce the size of the database, you can move the TemSe database to the file system. For more information, see Administering the TemSe Database [Ext.].

The spool work process is responsible for preparing the output data, processing the output request, and passing the output request data to the host spool system. Spool work processes run on the spool server. This host is known as the "spool server". A single spool server can support multiple spool work processes. Refer to Multiple Spool Workprocesses [Ext.].

Activities

Monitor the spool system [Ext.] to identify problems before they affect the availability of your spool service.

- When a spool server fails

  An output device is tied to a single spool server. Do the following:

  – Re-assign printers from a failed spool server to a new spool server. Refer to Assigning Output Devices to Another Server [Ext.].

  – To anticipate the problem, set up multiple and alternate spool servers to increase the availability of your spool service. You can also implement load balancing for the spool service. Refer to Spool Servers [Ext.].

- When the database service fails

  If the database service fails, no further spool requests can be generated and the following applies:

  – Existing spool requests that have not yet been processed are not lost but they cannot be processed.

  – Spool requests for which output requests have already been passed to the host spooler or SAPLPD (the transfer program) before the failure of the database service are processed by the host spool system. However, it is not possible to write a corresponding entry indicating “success” in the log table for the spool request.

    For more information about SAPLD, see Setting Up SAPLPD for Printers and Fax Devices [Ext.].

  – The spool service is unavailable for the period when the database is down.

  You must now check the consistency of the spool database [Ext.].

- When SAPLPD, the destination host, or the host spool system fail

  In this case the spool request cannot be successfully processed. Such requests are sometimes set to “failed” in the corresponding spool table entries and you can manually restart them, once you have fixed the original problem (that is, the failure in the program SAPLPD, destination host, host spool system).

  Refer to Problem Analysis: Print-Out Missing or Incorrect [Ext.].

- When the spool work process fails

  In this case the dispatcher restarts the failed spool work process. Then the relevant “administration transaction” on the database is rolled back, so that the uncompleted spool requests are not deleted. The new spool work process then takes over the work of the terminated one.

- When the output device (for example, printer) fails

  In this case the spool request is usually not set to “failed” in the log table. Instead it is normally set to “completed”. For example, this situation might arise because the host spool system incorrectly reports successful processing or because the host spool system does not report anything and processing is assumed to be complete, once the output request has been passed to the host spool system.

So that you can manually restart such spool requests, do the following:

– Do **not** set the flag “delete after print”.

– Set the “retention period” to a high value (for example, 8 days), so that these spool requests are not deleted.

Refer to Displaying and Changing Spool Request Information [Ext.].

You can avoid the manual restart if the host spool system itself automatically repeats the output request, once the existing device has been repaired or an equivalent device has been installed and defined to the host spool system. In this case, the spool request in the log table is not defined as failed since the host spool system has not reported a failure.

- When a failed output device cannot be repaired

  If an equivalent device is not available, then it is not possible to satisfactorily print the output requests that were prepared for the failed device type on a different device type. You need to regenerate the spool requests in this situation, or alter them, so that they can be printed on a different device type. For more information about printing on a different device type, see Displaying and Changing Spool Request Information [Ext.] [Ext.]. You can redirect spool requests from list-type output (that is, from SAP reports), assuming the spool type and print control exist on the different device type, but not output from SAPscript.

  You can define a “device pool” containing equivalent devices. If a device defined in a device pool fails, you must remove the failed device from the device pool using the spool administration (transaction SPAD) so that the failed device does not receive any further requests. The device pool continues operating after the failure of a device but with reduced throughput. Any requests that were assigned to the failed device before it was removed from the pool are not automatically redirected to functioning devices in the pool.

  Refer to Assigning an Output Device to a Pool [Ext.].

R/3 Incremental Table Conversion

Use

R/3 upgrades [Page 34] invariably lead to changes in the structure of database tables. Sometimes, this means that a complete restructure is necessary, with the conversion of each row in the table. In previous R/3 Releases, this conversion occurred during upgrade downtime, so increasing that downtime. Incremental table conversion with transaction ICNV now lets you perform conversions **before** the upgrade, that is, during production operation.

The benefits of incremental table conversion are:

- Reduced downtime during upgrade
- Simpler conversion back to SAP standard for modified tables
- Conversion of large tables during production operation

Features

- During the "prepare" phase of the upgrade, the system checks whether transaction ICNV can run with your database.

- If so, the system identifies all tables that might benefit from incremental conversion. For example, this includes tables containing large amounts of data, which would considerably extend the downtime of the upgrade.
- This all takes places automatically without any extra work.

Activities

If you have tables that might benefit from incremental conversion, then the system asks you to start transaction `ICNV`, leading to the following:

1. The system asks you:
   a. Which modified tables you want to incrementally convert back to the standard SAP table definition
   b. Which non-modified tables you want to incrementally convert
2. You start the incremental conversion.
3. You watch its progress.
4. The system estimates the time taken for the conversion, so helping you to plan the start of the upgrade.

The conversion is completed during upgrade downtime in phase PCON. Since most of the conversion has already been done, the downtime is significantly reduced.

Be sure to use the extensive online help with the ICNV transaction.

SAP recommends that you:

- Do **not** archive tables that are being incrementally converted. Instead, archive before the conversion.
- Do **not** attempt to modify tables that are being incrementally converted. These tables are locked until the end of the upgrade, so updates (including transaction `SE11`) are not possible.
- Observe the resource usage of the database so that you can spot bottlenecks early on. You might have problems because:
  - Extra space is required, as each converted table has to be replicated before conversion
  - Extra transactions are produced, leading to increased logging activity
- Make sure that enough batch work processes are available, preferably one batch work process for each table to be converted. If you are converting a large number of tables, transaction `ICNV` distributes the tables to the available batch work processes.
- Only start the upgrade when at least 95% of tables have been converted. This means that you have the greatest possible advantage in reducing

downtime. You can easily observe the progress of the conversion using transaction `ICNV`.

R/3 Downward-Compatible Kernel

Use

The R/3 System consists in general terms of a kernel and applications. The kernel is the central program that is essential to the R/3 System applications. From R/3 Release 3.0C, you can upgrade the kernel to a higher release while keeping the applications at the original release. Therefore, the kernel is said to be "downward-compatible".

The advantage is that you gain the latest kernel, incorporating the most up-to-date program patches and other improvements, while avoiding the extra effort of a full application upgrade. For example, this is very useful if you plan to do a full upgrade at a later date but still want to benefit from the latest kernel immediately.

Kernels are only downward-compatible within the same release sequence. For example, you cannot install a 4.X kernel when you are still running R/3 Release 3.X.

Integration

The downward compatibility of R/3 kernels is supplemented by the downward compatibility of the database management system (DBMS) and operating system (OS). This means that you can normally update the DBMS and OS at the same time as you update the kernel, without affecting your applications.

For example, suppose that you are running R/3 Release 3.0F and you want to update the kernel to 3.1I. At the same time, you could also update the DBMS or the OS. The higher DBMS or OS release does not need to have been explicitly approved for R/3 Release 3.0F.

Prerequisites

Before you upgrade to a new R/3 kernel with a new DBMS or OS release, make sure that there are no relevant restrictions listed in the SAP Notes. For example, a new release of the DBMS might only be downwardly compatible to R/3 Release 3.0F, but not right down to Release 3.0C. Such restrictions are always listed in the SAP Notes.

Activities

You must always check the relevant SAP Notes before you install a higher release R/3 kernel, DBMS, or OS.

See also:

SAP Note 93086

Other SAP Notes in the component XX-SER-SWREL (Release Planning)

R/3 Transports

Use

Transports into productive R/3 Systems sometimes cause problems because they require a significant amount of time, especially if the transport is large. Therefore, optimizing transports can contribute significantly to increasing the availability of your R/3 System.

Activities

You can take the following steps to solve this problem:

- Make sure that enough processing power is available for background processing. Remember that a transport is a background job, which competes with other jobs for processing power.

  To achieve this goal, do the following:

  – Plan your workload so that, at the time of the transport, there are few or no background jobs running. Any background jobs that are running should have minimal processing requirements.

    For more information, see SAP Notes 26966 and 50104.

  – Adjust the R/3 System operation mode for the period of the transports to favor background processing, using the Computing Center Management System (CCMS).

    For more information, see Operation Modes [Ext.].

- Use the new type of import available starting with R/3 Release 3.1I, which performs the command `TP IMPSYNC`. This lets you perform transports without having to bring down the R/3 System. By using `TP IMPSYNC` you can avoid problems with data buffering, so guaranteeing the consistency of your data.

  For more information, see SAP Note 102069.

- Use "merged" transports. A merged transport has been consolidated to remove duplicate entries, so leading to more efficient processing.

  For more information, see SAP Note 139513.

See also:

Transport Management System (BC-CTS-TMS) [Ext.]

Transport Organizer (BC-CTS-ORG) [Ext.]

Transport Tools (BC-CTS-TLS) [Ext.]

Database Key Issues

Purpose

This section describes in detail how you can minimize downtime caused by the database management system (DBMS) of the R/3 System with:

- Oracle [Page 46]
- Informix [Page 68]

- SAP DB [Page 82] (previously called "ADABAS")
- DB2 Universal Database [Page 89] (abbreviated to DB2 UDB, previously called "DB2 common server" or "DB2/CS")
- DB2 for OS/390 [Page 96]
- DB2/400 [Page 106]
- MS SQL Server [Page 108]

There are several database-related tasks and situations that can cause downtime of the database or prevent single transactions from proceeding.

Implementation Considerations

The information in this section is useful whether you are installing the DBMS for the first time, or considering ways to improve the availability of an existing installation. For detailed technical guidance when implementing a specific product or feature, contact the appropriate source, such as your SAP consultant, your DBMS supplier, the SAP Competence Center, and so on.

Integration

High availability for the DBMS should be part of a system-wide strategy for improving the availability of your R/3 System. Therefore, you should also consider other components of the system, such as the R/3 System itself, the network, the hardware and operating system services, and so on.

Features

The information is grouped into the following areas for each DBMS:

- Space Management (including configuration and reorganization)
- Backup or archive
- Recovery
- Upgrade

See also:

SAPDBA: Oracle [Page 155]

SAPDBA: Informix [Page 159]

Database Manager (DBMGUI): SAP DB [Page 163]

DB2CC Tools for DB2 UDB [Page 166]

Documentation in SAPNet

High Availability for the Oracle Database

Purpose

This section looks at database administration for the Oracle database and makes specific recommendations on improving availability.

For up-to-date information on Oracle with the R/3 System, you can use the alias "dbaora." Enter the following in the address line of your web browser:

```
http://sapnet.sap.com/dbaora
```

Process Flow

1. You work out your approach to backing up your database. You must do this before you start production with the database. When using the database for production, you must back up the database as regularly as possible.

   Refer to Backup with Oracle [Page 60].

2. You manage the space in your database. You do this both before you start production with the database and during production as the database grows.

   Refer to Space Management with Oracle [Page 48].

3. You upgrade your database when required.

   Refer to Upgrade with Oracle [Page 67].

4. You recover your database if a failure occurs with data loss.

   Refer to Recovery with Oracle [Page 66].

5. You consider using various tools and services offered as standard by SAP to increase the availability of your database:

   - SAPDBA [Page 155]
   - Computing Center Management System (CCMS) [Page 166]
   - GoingLive and EarlyWatch [Page 169]

   When the R/3 System is up, you should use the integrated CCMS facility to monitor the database. When the R/3 System is not up (for example, in the event of operational problems) or for extra functionality not offered by the CCMS, you should use the specially designed tool, SAPDBA for monitoring and administering the database.

6. You consider using advanced products and services to increase the availability of your database:

   - DB Reconnect [Page 174]
   - Oracle Standby Databases [Page 187]
   - Oracle Parallel Server [Page 208]
   - Switchover Software [Page 216]

7. You review your procedures to increase the availability of your database. Refer to High Availability Procedures at Your Site [Page 247].

Certain activities (for example, reorganization of database objects, recovery, tuning and configuration) require the database (or portions of the database) to be offline, resulting in R/3 downtime. Therefore, it is much better to anticipate problems by planning for them. By monitoring and pro-actively managing your database, you reduce the need to bring the database offline.

Most tuning activities require short downtimes only (that is, a restart of the database) for parameter changes to take effect. Therefore, these issues are not covered here.

Result

Your Oracle database is more available for production use.

See also:

Database Administration (Oracle) with SAPDBA [Ext.] [Ext.]

Documentation in SAPNet

Space Management with Oracle

Purpose

This section looks at space management (including reorganization) of database objects (that is, tables and tablespaces). If you neglect space management, this can lead to downtime due to normal database growth when database objects fill up. If this happens, applications cannot write to the database and you have to quickly make more space available. You might need to bring down the R/3 System to tune and configure the database. Therefore, it is much better to anticipate the problem by monitoring and pro-actively managing the disk space in your database.

SAP recommends you to manage space on your Oracle database using the Computing Center Management System [Page 166] (CCMS) in the R/3 System and SAPDBA: Oracle [Page 155]. You need to monitor regularly and occasionally take timely action to avoid the problem leading to downtime.

Situations when you need to reorganize include the following:

- Tablespace overflow (or data file freespace shortage)
- Fragmentation
- Chained rows
- Maximum number of files reached

The most likely events to require reorganization are index fragmentation, tablespace overflow, and chained rows.

Avoid reorganization if possible

SAP strongly stresses that you should avoid reorganizations wherever possible. You can achieve this by correct configuration and sizing of the database together with proper monitoring.

The problems of tablespace overflow and fragmentation are more likely to occur in the following tablespaces in an R/3 System (add "D" for data tablespace or "I" for index tablespace to the end of each tablespace name):

Tablespace	Comment
PSAPBTAB	Transaction data tables. Objects in this tablespace might expand very rapidly.
PSAPSTAB	Master data tables. Objects in these tablespaces might expand very rapidly.
PSAPCLU	Clustered tables, such as financial tables. Objects in these tablespaces might expand very rapidly.
PSAPPOOL	Pool tables, containing customization tables.
PSAPPROT	Spool (that is, print) requests, protocols

Pay special attention to the following tablespaces in certain circumstances:

- PSAPROLL

  Watch this tablespace if you are running a large export or reorganization, background jobs, or if your applications have high transaction rates and few commit points. The tablespace contains the rollback segments and these might be too small to handle large transactions for a particular installation.

- PSAPTEMP

  If you are running a large import or reorganization, you should closely watch this tablespace since it is used to store temporary objects, for example, objects to sort data for a “create index” operation.

  For more information about how to monitor these tablespaces see Computing Center Management System (CCMS) [Page 166]. For more information about when to perform the reorganization, see "reorganize your database" below in the process flow.

Process Flow

1. You manage tablespaces [Page 50], to avoid tablespace overflow.
2. You check for fragmentation [Page 54] and manage tables and indexes [Page 52], to avoid fragmentation.
3. You manage database blocks [Page 55], to avoid chained rows.
4. You manage files [Page 56], to avoid overflow.
5. You manage database files [Page 57] to avoid poor distribution and poor disk input/output (IO).
6. If necessary, you reorganize database objects [Page 58].

Result

By managing the space in your Oracle database, you can avoid unplanned downtime due to database objects filling up.

See also:

SAPNet documentation

Managing Tablespaces (Oracle)

Use

You manage tablespaces in your Oracle database to avoid tablespace overflow, which is when a tablespace runs out of freespace in the allocated file or files. This happens when an object requires a new extent but there is either no freespace or insufficient freespace in the tablespace.

Prerequisites

A tablespace overflow can occur in the following situations:

- Operations that greatly extend a table

  Be sure to plan certain operations (for example, client copy or batch input) carefully, because they might extend tables excessively.

- Poor monitoring

  During normal operation, database objects (that is, tables and indexes) grow steadily. Be sure to monitor the database, anticipate growth, and make sure there is always enough disk space available. With SAPDBA [Page 155] and the Computing Center Management System (CCMS) [Page 166], you can easily monitor your tablespaces.

Procedure

1. Regularly monitor freespace and rapidly growing objects in the database, using SAPDBA or the CCMS.

   If freespace in a tablespace decreases continuously, extend the tablespace in time to accommodate further growth.

   Also monitor the database for rapidly growing objects, that is, objects that allocate more and more extents, using the CCMS or SAPDBA.

2. Monitor disk space at operating system level

   In addition to using the CCMS and SAPDBA, you also need to monitor available disk space at the operating system level. Plan for additional disks in time to be able to accommodate tablespace growth.

3. If necessary, make more space available in a tablespace using one of the following approaches:

   – Add a data file to extend the tablespace while the database is online to avoid downtime

     If no disk space is available and you cannot use the other methods described below, this might cause a situation where the application is partially unusable because insert and update operations cannot continue. If you use SAPDBA for this task, the new file is created according to SAP's naming conventions.

Oracle also offers a feature to enable automatic extension of the data files of a tablespace – see the `autoextend` option of the `create tablespace` and `alter tablespace` commands. This avoids a tablespace overflow, but only as long as enough disk space is available. SAP does **not** recommend using this feature at present. Therefore, the approaches below do not take account of this feature.

– Reorganize the tablespace or single objects

It might be possible to free up space in a tablespace by reorganizing the whole tablespace or single objects in that tablespace. This is only true if particular objects or objects in general in a tablespace contain a lot of unused space (for example, after a table has many inserts followed by many deletes). The best solution is to reorganize the affected table(s) using SAPDBA (using *Reduce object size* yes). However, this causes downtime.

– Deallocate free space

If space allocated to a table or index has never been used (that is, data has never been written to the data blocks), you can free such space for use by other objects in the tablespace. You can do this online without incurring downtime using the command `alter table|index <name> deallocate unused`. When a tablespace overflows and there is no disk space available and a reorganization is not possible (for example, because downtime cannot be tolerated), you might be able to continue processing with this approach.

However, remember that the objects for which you have deallocated free space might themselves soon require new extents as they grow with inserts and updates. Therefore, a more permanent solution is to reorganize affected objects when downtime can be tolerated and also, if necessary, extend the tablespace.

This approach cannot free space that was once used, for example, when an insert has been followed by a delete.

SAP recommends the following concerning the above approaches to making more space available in a tablespace:

- If possible, add a data file instead of reorganizing objects or deallocating free space. If all the above approaches are available to resolve tablespace overflow – that is, if disk space is available to add a new data file and if a reorganization or free space deallocation would free up space in the tablespace – always choose to add a data file and avoid the reorganization or free space deallocation.
- Note that reorganization (either of individual objects, or of the entire tablespace) without recreating the data files might not be possible if the tablespace is full or very nearly full. The general rule is to reorganize before 90% of available space has been allocated.

Since the reorganization tends to create objects with fewer larger extents and evenly distributed freespace (according to `PCTFREE`), the end result might be that more space in total is needed than before the reorganization. As a result, it might be impossible to fit all the objects back into the tablespace. To avoid this it is recommended to reorganize a tablespace well before it is full. SAPDBA automatically checks for this condition when preparing for a reorganization. For more information, see Reorganizing Objects (Oracle) [Page 58].

Result

By managing the tablespaces in your Oracle database, you can avoid unplanned downtime due to database objects filling up.

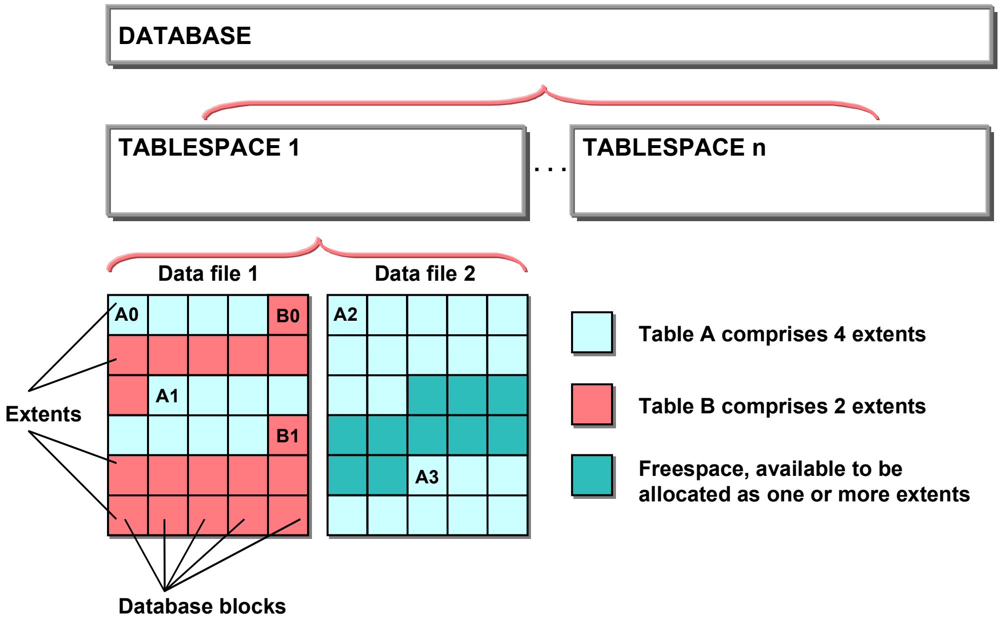
Managing Tables and Indexes (Oracle)

Use

To avoid fragmentation, you must manage tables and indexes in your Oracle database. An object (table or index) is fragmented if it has a high number of extents allocated or if it has a lot of unused space in the allocated blocks. A high number of extents is sometimes called “external fragmentation”, while unused space is sometimes called “internal fragmentation”. Fragmented objects can threaten continuous availability of the system by reducing performance, so leading to downtime for reorganization.

For data storage, the Oracle database management system (DBMS) allocates extents to objects (tables or indexes), as shown in the diagram below.

Oracle Storage Allocation


DATABASE
TABLESPACE 1
. . .
TABLESPACE n
Data file 1
Data file 2
A0
B0
A1
B1
A2
A3
Extents
Database blocks
Table A comprises 4 extents
Table B comprises 2 extents
Freespace, available to be allocated as one or more extents

In Oracle 8 there is a limit for the maximum number of extents that one object can have. This limit is determined by a parameter called `MAXEXTENTS` in the storage definition of an object. `MAXEXTENTS` is usually set to a value of 300 or 505. The limit can be altered without downtime, so allowing processing to continue against the affected object.

Up until Oracle 7.2 there was a hard limit for the number of extents that depended on the database block size, but this limit has now been removed.

Prerequisites

Fragmentation has the following effects:

- Table fragmentation

  This does not usually cause performance degradation and does not require a reorganization, because R/3 has been optimized to use indexes. Even tables with very many extents (for example, 100) might not suffer performance degradation. However, this depends on how the table is accessed in practice. For example, a read of one record using a unique key and an index is not affected by the number of table extents. However, a read of a range of records might be affected, even if an index is used.

- Index fragmentation

  A fragmented index can potentially degrade database performance to the extent that reorganization is necessary, so incurring downtime. Heavily used indexes might already cause problems with only 20 extents, whereas others do not. It is not possible to give a general rule. Causes for index fragmentation might be improper index storage parameters and frequent insert, update and delete operations.

  You need to reorganize an index if the index is heavily used and fragmented. Fragmentation can be a high number of extents or a lot of unused space in the allocated blocks (this happens if many insert operations in a table are followed by many delete operations).

Procedure

1. Regularly monitor the objects in your database using SAPDBA, as follows:

   – Monitor database objects (especially indexes but also tables) for the number of allocated extents. SAPDBA provides a function to list all objects with more than n extents, where n can be specified.

   – Monitor the database for indexes with excessive unused space. You can do this with SAPDBA or the database tools supplied by Oracle, as described below. See Checking for Fragmentation (Oracle) [Page 54] for a detailed description or refer to the SAPDBA documentation.

   You can also use the Computing Center Management System [Page 166] (CCMS) to monitor the number of extents.

2. Make sure that the extent size parameter `NEXT` is configured properly.

   You can also use SAPDBA to adjust the storage parameter `next`, which defines the size of the next extent to be created for objects with `sapdba -next`. Increasing the value of `next` results in the allocation of fewer but larger extents.

3. To raise the limit for the maximum number of extents that an object can have, use the command `alter table <table_name> storage (maxextents xx)` where `xx` is the new limit for the maximum number of extents.

   The limit can either be a number – for example, `2000` – or the keyword `unlimited`. Too high a value can lead to undesirable fragmentation. Currently you cannot use `unlimited` for rollback segments, and SAP recommends that you do **not** use it with other object types (such as tables, indexes, and so on).

4 If the object (table or index) has a high number of extents, use SAPDBA to reorganize the object.

For a table, you can use SAPDBA to adjust the `NEXT` parameter and perform the reorganization. For the reorganization of an index, extra disk space (in `PSAPTEMP`) is needed for index building (roughly 200% of the largest index to be reorganized, not the size of the index tablespace). Index reorganization is faster than table reorganization because the index is dropped and recreated whereas tables are exported and imported.

Reorganize at individual object level rather than at tablespace level

It is better to reorganize single tables or indexes rather than entire tablespaces. If you are closely monitoring these objects with the CCMS or SAPDBA, you can soon spot where problems are arising. For more information about reorganizations, see Reorganizing Objects (Oracle) [Page 58].

Result

You avoid fragmentation and so minimize downtime for your Oracle database and the R/3 System.

Checking for Fragmentation (Oracle)

Use

This section describes how to check for fragmentation in an Oracle database. For more information, see Managing Tables and Indexes (Oracle) [Page 52].

Procedure

You can check for fragmentation using SAPDBA or the database features supplied by Oracle:

- Check for unused space in an index using SAPDBA.

  You can do this interactively or using command line options, as follows:

  – Interactively, enter the following in SAPDBA:

    - **d** (for reorganization)
    - **a** (to check extents and fragmentation)
    - **e** (to validate index)

  – Using command line options to get a report including index statistics, for example:

    ```
    SAPDBA -analyze PSAPBTABI -option EI
    ```

    For a detailed descriptions of these options refer to the SAPDBA documentation.

    Towards the end, the report contains index statistics (charts of 20 empty indexes, using validate structure). The values of the two columns `used_by_btree` and `used` are important. The percentage used of used_by_btree is equivalent to the pct_used described below. It should not be below 50%.

- Check for unused space in an index using Oracle database features.

  The steps are as follows:

  a) Determine the index you want to analyze.

  b) Log in to SQL*Plus as user SAPR3.

  c) Run the command **analyze index <NAME> validate structure**

  d) Display statistics with the command **select * from index_stats**

  The column pct_used returns the average used space in index blocks. If this is below 50% the index is considered to have excessive unused space.

Managing Database Blocks (Oracle)

Use

You manage database blocks in your Oracle database to avoid chained rows. A chained row is spread over multiple database blocks.

Prerequisites

Chained rows are often caused by an incorrectly set PCTFREE value. To give a simplified example, with a database block size of 8 KB and a PCTFREE value of 25 (that is, 25% is kept free), 6 KB is available to store new rows in an empty block (insert operation). If a row has a length of 7 KB it is split in two pieces, with 6 KB stored in one block and 1 KB in a second block. If PCTFREE were set to 10% (7.2 KB available), the row would be stored in one block, so avoiding chaining.

Procedure

1. Monitor and adjust PCTFREE.

   Adjust PCTFREE to reduce or eliminate chaining. This requires a single-table reorganization only. Use SAPDBA to watch for average row length and number of chained rows early and adjust the table when it is still small (therefore, a shorter reorganization time is required).

   For more information about reorganizations, see Reorganizing Objects (Oracle) [Page 58].

2. Distinguish chained rows from “migrated” rows.

   A migrated row is caused when an update increases the size of a row such that it does not fit in the current block any more, leading to the entire row being moved (or “migrated”) to another data block. Since the row’s index entry still points to the original location, a pointer is set in the original location so that the new position can be found. Subsequent access to the migrated row now requires an extra read to locate the migrated row, causing performance deterioration. A table reorganization eliminates this problem since the index entries then point directly to the rows.

Result

You avoid chained rows and so improve performance for your Oracle database and the R/3 System. Poor performance can lead to downtime so avoiding chained rows also reduces downtime.

Managing Data Files (Oracle)

Use

To avoid exceeding the limit for the maximum number of files, you must manage the data files in an Oracle database.

Prerequisites

The following constraints limit the maximum number of data files in an Oracle database:

- The `db_files` parameter in the `init<SID>.ora` file is usually set to the value of the `maxdatafiles` option of the `create database` command. Currently, the SAP installation process sets `maxdatafiles` and `db_files` to 254. If you reach this limit, you can no longer add files to the database.
- There is a UNIX kernel limit for the maximum number of open files.
- There is an absolute Oracle maximum of 65533 files in a database and usually 1022 files in a tablespace. However, certain hardware platforms have a limit **lower** than this absolute maximum.

Procedure

1. Monitor your system regularly for the number of data files using SAPDBA or the CCMS.
2. Create sufficiently large data files.

   When creating data files, make sure that they are large enough to accommodate foreseeable growth in the tablespace. Otherwise, you have to repeat the procedure soon, and you might also eventually reach the limit.
3. If necessary to avoid reaching the limit, use SAPDBA to reorganize tablespaces [Page 58].

   If you are about to reach the lowest of the above limits and you decide to solve the problem by reorganizing a tablespace, you should use SAPDBA to identify tablespaces that consist of more than one data file. Then you can reorganize these so that the tablespaces are recreated with fewer data files, if possible with just one data file. SAPDBA offers the functionality to reorganize a tablespace including data files.
4. If you reach the limits specified in "Prerequisites" above, then do the following:
   – If you reach the limit for `maxdatafiles` and `db_files`, then do one of the following:
     - Increase the value of the `db_files` parameter, then shut down and restart the database.
     - Reorganize a tablespace consisting of several data files such that the number of files in use is reduced.

   – If you reach the UNIX kernel limit for the maximum number of open files, you have to change the relevant UNIX kernel parameter, then make sure that the change takes effect (such as shutting down and restarting the database, logging on again at UNIX level, restarting the UNIX system, and so on).

– In the very unlikely event that you reach the hardware-dependent limit of files in the database (65533 files or less), then you have to perform a reorganization.

Result

You avoid exceeding the file limits for your Oracle database, so reducing downtime for the database and the R/3 System.

Managing Database Files (Oracle)

Use

To improve availability and performance, you need to manage your database files and their distribution. An Oracle database consists of control files, redo log files and data files. Control files and redo log files can be mirrored at the Oracle software level.

We make recommendations here on file distribution for performance (that is, I/O load balancing) as well as for high availability. This is because a poorly balanced system might necessitate a reorganization of the database or redistribution of data files, so causing downtime (however, redistribution of data files requires only short downtime).

💡

Make sure the system is correctly balanced from the outset so as to avoid both performance and high availability problems later. Load balancing is a complex area so make sure that your intended alteration will actually bring about the desired improvement.

Procedure

1. Set up the control files and redo log files correctly.

   SAP recommends having at least three control files and two members (that is, copies) in each redo log group. From the high availability viewpoint, place the three control files on different disks. In the same way, separate the two members of a redo log file group on different disks. This makes sure that a single disk failure does not lead to the loss of all control files or of all members of a redo log group.

2. Use the R/3 disk space configuration program.

   You should use R/3's disk space configuration program (the R3 config tool) to estimate table and tablespace growth for each business application. This allows you to configure your disks correctly from an early stage, so avoiding problems later.

3. Inspect R/3 ABAP/4 TABLE buffers before DBMS.

   Focus on R/3 ABAP/4 TABLE buffers as performance levers because the DBMS is usually an insignificant factor towards the poor performance detected by the end user.

4. Use SAPDBA to reorganize tablespaces.

   If your system appears to be unbalanced it might be sensible to reorganize the tablespace and its data files using SAPDBA. Make sure that you have correctly interpreted the problem. For more information about reorganization, see Reorganizing Objects (Oracle) [Page 58].

Result

Your system is optimally balanced in terms of file distribution and disk I/O, so improving performance reducing downtime for the database and the R/3 System.

Reorganizing Objects (Oracle)

Use

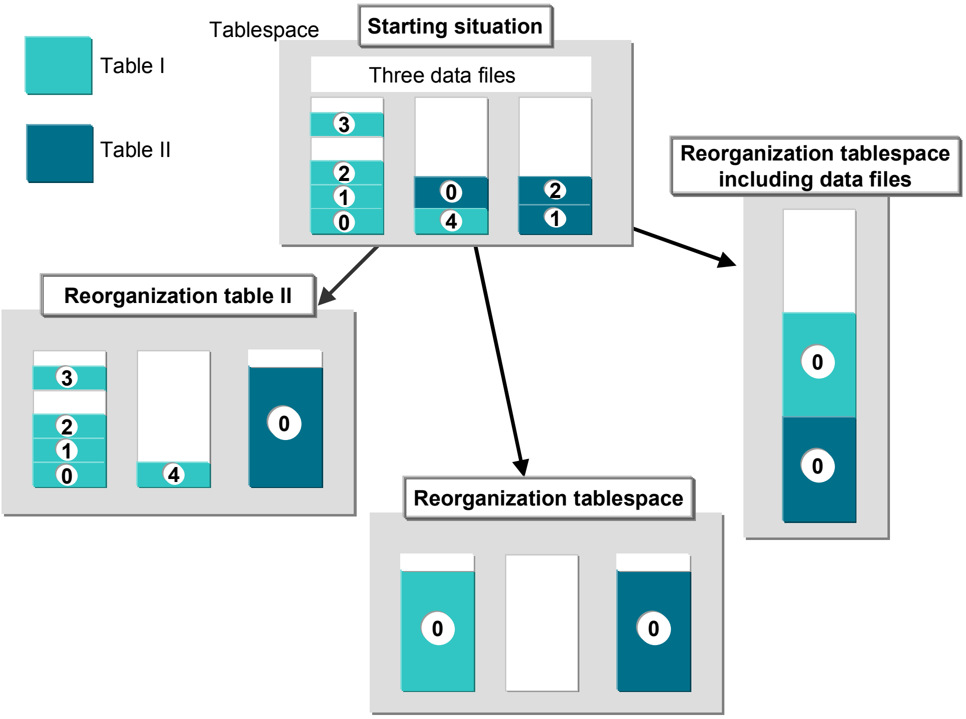
The following reasons might necessitate reorganization of the objects in your Oracle database:

- A tablespace might soon overflow and you decide to reorganize it rather than add a new data file.
- A database object might be severely fragmented.

Be sure to regularly monitor the database using the Computing Center Management System (CCMS) in the R/3 System or SAPDBA, so that you can anticipate problems requiring reorganization before they cause unplanned downtime.

The following diagram illustrates the kinds of reorganization possible (there are three data files involved in the reorganization, containing various extents from two tables):

Oracle: Reorganization Types



Prerequisites

When you decide to perform a reorganization, there are a large number of factors to consider if you want to minimize the downtime (note that some of the recommendations given below are performance-related since improving the speed of a reorganization can reduce downtime).

The duration of data reorganization depends on the database size and the objects to be reorganized and could take several hours. Index reorganizations are usually faster than table reorganizations. For example, the reorganization of an index of about 400 MB in size might take less than 10 minutes while for a 400 MB table around two hours might be needed.

Procedure

This procedure consists of recommendations about when and how to reorganize.

1. Reorganize only when necessary.

   Monitor the system closely and only reorganize when necessary. It is far better to reorganize individual tables and indexes rather than entire tablespaces. You can do a single table reorganization with the R/3 System running. However, you must perform a tablespace reorganization when the R/3 System is down. SAPDBA does not let you execute a tablespace reorganization as long as an R/3 application server is connected to the database.

2. Reorganize early when database is small.

   Do reorganizations early while the database is smaller. This can be done based on early usage patterns and future projection. For example, perform reorganizations immediately after an upgrade.

3. Use EarlyWatch [Page 169] to monitor storage parameters.

   One way of monitoring the system is to use the EarlyWatch service to detect storage problems early. EarlyWatch also gives recommendations to adjust parameters, so helping to avoid reorganization completely.

4. Schedule reorganizations alongside backups.

   It is a good idea to back up tablespaces before reorganization. Therefore it makes sense to schedule reorganizations immediately after regular backups to avoid the need for a separate backup. It is also a good idea to batch a number of reorganizations together to reduce total downtime.

5. Prepare properly for reorganization.

   Check using SAPDBA to make sure all required disk space/operating system files are available and pertinent parameters are set properly before shutting down the database host. This avoids unnecessary downtime due to improper settings or insufficient resources.

6. Export to disk instead of to tape in a reorganization.

   A faster reorganization can be achieved if you export to disk rather than to tape. Even better, although requiring a lot of disk space, is to export to non-compressed disk format. Furthermore, export to disk is more secure than export to tape since checking is carried out during the export to guarantee the integrity of the operation.

7. Maximize performance during reorganization to minimize downtime.

Temporarily set DBMS parameters to maximize reorganization performance during shutdown (for example, increase sort buffers, increase block I/O sizes). For more information about detailed parameter settings, refer to the Oracle documentation. Maximizing reorganization performance is useful because DBMS parameters configured for production time are rarely optimized for data reorganization. The reverse is also true, which means it is equally important to restore the DBMS parameters to their original values after reorganization is complete.

8. Use parallel index building if possible.

   If you have a multi-processor machine, use parallel index building to reduce reorganization time.

Result

Reorganization helps you to solve problems that might lead to downtime. However, be sure to continue monitoring the database so that you can anticipate future problems before they cause downtime.

Backup with Oracle

Purpose

You must always have a recent and consistent copy of Oracle database data that can be used to recover the database in the event of failure involving data loss.

See SAPDBA [Ext.] for more information about backup options, including a description of SAPDBA integration with Oracle Enterprise Manager (OEM) and Recovery Manager (RMAN) and of support for incremental backups.

The Oracle database management system (DBMS) supports the following types of backup:

Oracle Backups

Type of backup	What gets backed up
Database	Full backup, all data files are saved
Tablespace	Partial backup, data files of one or more tablespaces are saved
Data file	One or more data files are saved

You can perform offline or online backups, as follows:

- Offline

  During an offline backup the whole database, or parts of it, are unavailable for use. Offline backups are usually done at the database level. Offline backups at the tablespace or data file level are not advisable in an integrated environment like R/3, because the application modules require access to data in several tablespaces. If one of the main tablespaces is not available, most application modules cannot continue.

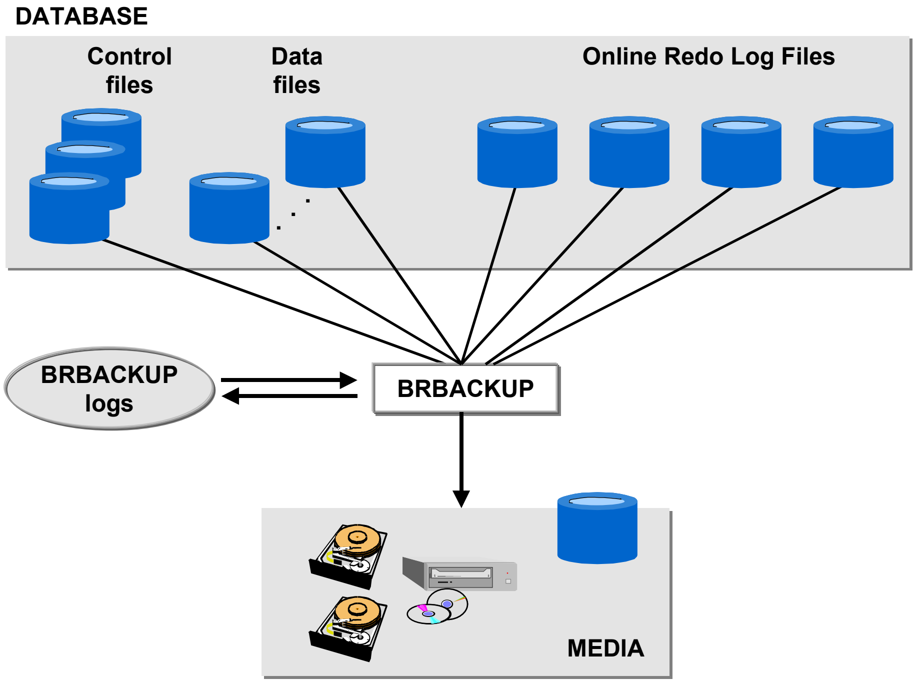
  For an offline backup at the database level, you must close the database. Then you should do a full backup, that is, you must back up all data files, all online redo log files, and at least one control file.

  The advantage of such a full backup is consistency. This backup could be restored and the database could then be opened without needing to do a recovery.

The major disadvantage is, obviously, that the database cannot be used while the backup is running.

BRBACKUP (the SAP backup tool) supports offline backups at all levels.

Oracle Offline Backup with BRBACKUP



- Online

  An online backup is more appropriate for large databases. An online backup can be a full or a partial backup. The backup includes all or some of the data files and a control file. For the backup to be usable, the offline redo log files generated while the backup was done also need to be available.

  The advantage is that the database is available for normal use while the backup is taken. Online backup has the following disadvantages:

  – Performance deteriorates

    The DBMS performance worsens because more redo log information is generated while the backup is going on. Therefore you should do online backups during times of low system load. For example, an online backup running alongside a background job with batch input causes the run time of the background job to increase considerably.

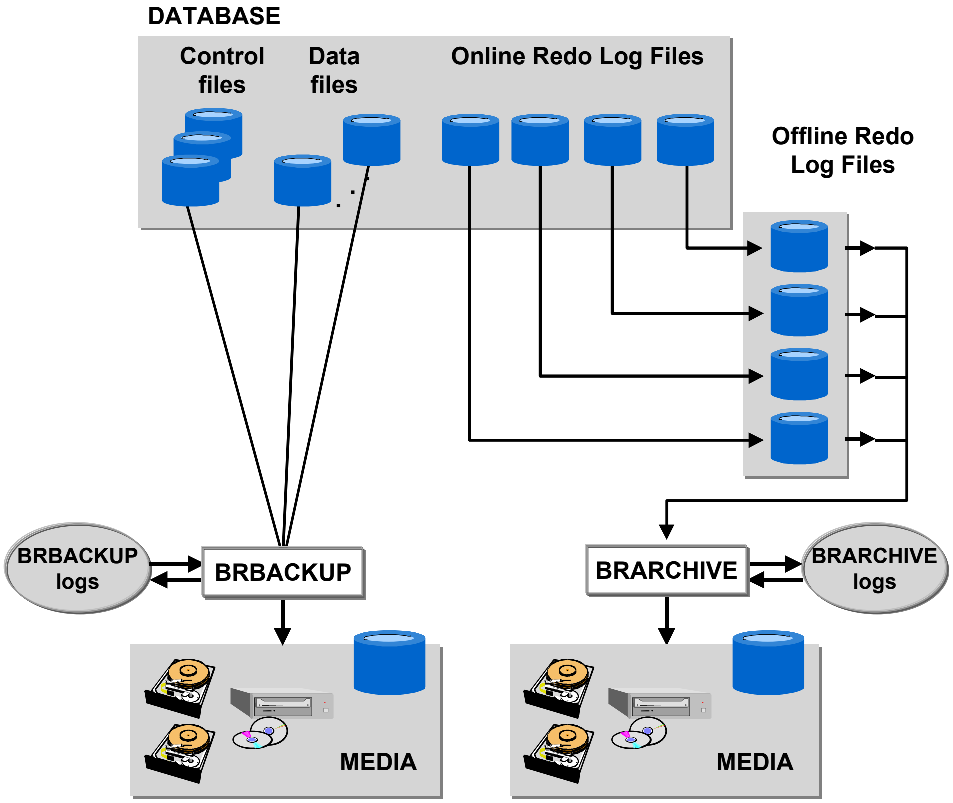
  – Backed up data files are inconsistent without the redo log files

    If used for a recovery, the online backup must be supplemented with the redo log files archived during the backup. To make sure that the last online redo log file from the backup is safely archived, you should perform a redo log switch immediately after the backup. An online backup done with SAP's BRBACKUP tool does this automatically.

Backup with Oracle

You could perform an online backup by first running BRBACKUP to save data files and then running BRARCHIVE immediately afterwards, to save the archived redo log files belonging to the backup.

Oracle Online Backup with BRBACKUP



Process Flow

1. You decide what kind of backups to do, as follows:

   – Whether to use offline or online backups or both

     Only offline backups cause system downtime. Use online backup if downtime cannot be tolerated or if your database is very large and would take too long to back up offline.

   – What level of backup to use

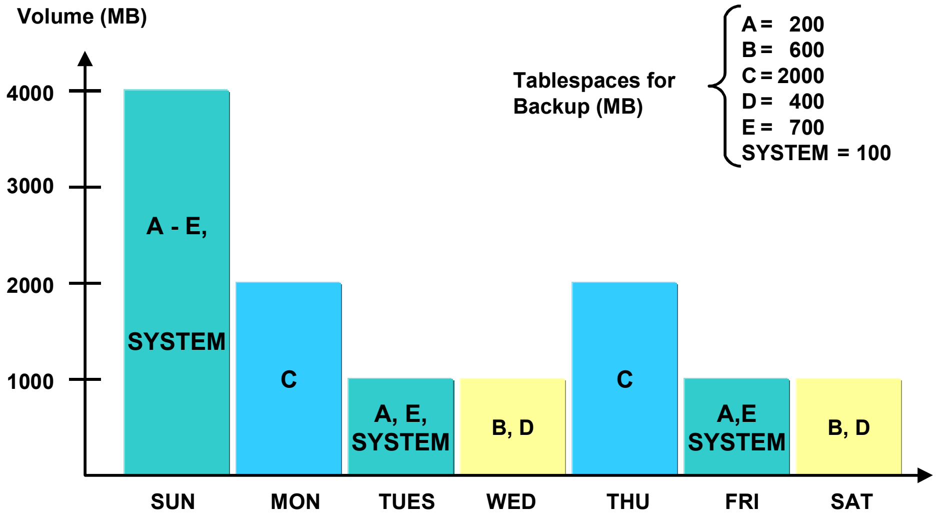
     If you have decided that a complete backup takes too long, use backups at the tablespace level. For example, if the database is comprised of 20 tablespaces, back up four of them every night Monday to Friday (that is, equivalent to one full backup a week). Then the worst case would be that a tablespace damaged on Sunday would have to be recovered from last Monday’s backup.

2. You consider the frequency of backups, if possible increasing the frequency.

   More frequent backups lead to shorter recovery time and therefore shorter downtime. A good compromise is to make less frequent full backups, (for example, one full backup on Sunday) and more frequent partial backups (for example, back up one third of all tablespaces Monday, Tuesday, Wednesday and again Thursday, Friday, Saturday).

Then the worst case would be to use a backup that is three days old. The following diagram shows an example of this approach:

Oracle: Example of Partial Backup Schedule



Increase backup frequency of heavily used tablespaces

SAP recommends backing up heavily used tablespaces more often, for example, by including them twice in a backup cycle. The reason is that heavily used tablespaces put more load on disk drives, so increasing the probability of disk failure. The more often such tablespaces are backed up, the more likely a recent backup is available to be used in a recovery.

3. You consider parallel online or offline backups to increase throughput.

   For example, if partial online backups of tablespaces are done (see example above), you can also schedule the backups of several (groups of) tablespaces to run in parallel, utilizing multiple devices. BRBACKUP supports parallel backups.

4. You adjust the backup frequency by performing tests.

   There is no rule of thumb to determine the backup frequency. Suppose, for example, a test showed that to apply 3 redo log files to a restored full backup took 15 minutes (that is, 5 minutes recovery time per archived redo log file) and assuming 20 redo log files are archived on average a day, then a recovery from a 3 day old full backup would take 3 x 20 x 5 minutes = 300 minutes (or five hours). If, as in this example, a recovery time of five hours is too long, more frequent backups can be taken (or other techniques such as mirrored disks can be used).

5. You consider backing up to disk first if you have enough disk space.

   A disk backup is usually faster than a tape backup because disk devices are generally faster. You can then copy your backups from disk to tape without incurring downtime. If possible, retain the disk backup copy since a restore from disk is faster than from tape. Note that this assumes that the disks are not mirrored. Other options are available with mirrored disks (see below).

6. You consider using mirrored disks [Page 65] to reduce the downtime for an offline backup.
7. You consider using hardware compression with backup.

   This cuts down backup time by as much as 50%. If the backup must go to tape directly and be done online, consider using multiple tape drives for parallel backups to shorten backup time.
8. You assess your tape devices.

   Think about what kind of tape devices you are using for backing up your database since this determines the downtime in the case of offline backups. To give you some idea of this, the capacity of tapes currently ranges from 2 to 30 GB and the speed of data transfer ranges from 1 to 10 GB per hour.
9. You verify that your backup tapes are readable for a restore.

   It is best to use a separate system from the live production system. To perform a test, you might want to restore the data files of the system tablespace, rollback segment tablespace, temporary tablespace and a tablespace of your choice, plus control files, online redo log files, and archived redo log files. Mount the database, take offline all data files that were not restored, recover the database, and then open it. If successful, this is proof that the restored files can be used.
10. If you use the Oracle backup tool RMAN, you consider using incremental backups

    Incremental backups are available only possible if you use RMAN. They can help you to dramatically reduce the size of backups. You should start with a level 0 incremental backup, because this backs up all data blocks of the data files. Then you can use incremental backups with a level greater than 0 to back up only data blocks that have changed since the last incremental backup of the same or a lower level.

    A recommendation made in earlier versions of this documentation was to skip index tablespaces during backups, because it might be faster to rebuild an index tablespace and the indexes than to restore and recover an index tablespace. This recommendation is **no longer valid**.

    SAP now uses Oracle8's cost-based optimizer. This means that one more step is now necessary after the rebuild of an index tablespace and its indexes. The extra step requires executing the command `analyze index <name>` … to calculate the statistics required by the cost-based optimizer. With this the total time is longer than the time for restore and recovery

Result

You always have an up-to-date backup of your database. This allows you to quickly recover the database [Page 66] in the event of a failure involving data loss. Therefore, the availability of your R/3 System is increased because downtime due to the absence of a suitable database backup is kept to a minimum.

See also:

SAP Tools to Back Up the Oracle Database (in SAPNet)

Mirroring with Backup (Oracle)

Use

You can use mirrored disks to reduce the time taken for an offline backup with the Oracle database. If mirror disks are used, you can still do online and offline backups in the normal way, but there are additional aspects discussed in this section that you need to be aware of.

The method described here has the disadvantage that the database is not mirrored during the backup. If you must have disk mirroring all the time (for protection against disk failures) this method can be safely deployed if three-way mirroring is used.

Procedure

1. If you want to perform **offline** backups in a mirrored disk environment, proceed as follows:
    a. Shut down the database and split the disk mirror.
    b. Restart the database on one disk set and perform offline backup on the other disk set.

       You can perform offline backups while the database is online. You have to first shut down the database, then split the disk mirror and then you can restart the database on just one disk set. Next, you usually mount the split-off disk set on another system and perform an offline backup.

       The downtime for this method is the time taken to shut down the database, split the mirror, and then restart the database.
2. If you want to perform **online** backups in a mirrored disk environment, proceed as follows:
    a. Put the tablespaces in backup mode and split the disk mirror.
    b. End backup mode and perform the backup on the split-off disk set.

       If the split-off disk set is mounted on another system – possible for both the approaches outlined above – the backup processing load is removed from the production system, so there is no performance loss during backup.
3. After the backup is finished, resynchronize the two disk sets. It can take a long time to resynchronize, leading to severe performance loss.

   Use BRBACKUP with mirrored disks

   BRBACKUP supports backup for mirrored disks, using the `backup_types` `offline_split` and `online_split`. BRBACKUP records information about backups in the database. For mirrored disks, BRBACKUP writes the information about the backup to the database running with the active disk set, while it backs up the files from the split-off disk set to tape.

Result

You have reduced the downtime for an Oracle offline backup or reduced performance loss during an Oracle online backup.

Recovery with Oracle

Purpose

To minimize downtime with your Oracle database, you must make sure that you can quickly recover the database in the event of a failure with loss of production data. There are the following types of recovery with Oracle:

- Instance recovery

  This is required if structures on disk are **not** damaged, but the database instance has aborted.

- Media recovery

  This is required if structures on disk (that is, control file or files, online redo log files, data files) are damaged. It consists of the following steps:

Phase	Description
Check	Analyze the problem and determine the proper course of action.
Restore	Restore the appropriate files, for example, backed up data files and archived redo log files.
Recovery	Recover the database as far as possible (that is, roll forward transactions from the logs). Open transactions are rolled back during database opening.

SAPDBA [Page 155] automatically supports all the steps in media recovery. If backups are done using SAP’s BRBACKUP tool, SAPDBA is able to select the backups needed, and restore the damaged files. SAPDBA supports a variety of recovery possibilities, including recovery to the current point-in-time, to a point-in-time in the past, and resetting of the database (that is, restore of a full database backup only).

Instance recovery is done automatically when the database instance is restarted. This section only looks at media recovery, during which the database is unavailable.

Process Flow

1. Devise a recovery strategy and rehearse it in practice

   SAP strongly recommends you to rehearse a disaster. You could, for example, use a test system to restore data files of some tablespaces (at least the system and rollback segment), control files, online and archived redo logs of a backup taken from the production system. Then perform the recovery, applying the redo log data.

   This exercise is very valuable to give you some idea how long a recovery would take.

2. Repeat the recovery rehearsal often enough (for example, twice a year) to take account of the effects of database growth on recovery time.

3. Use SAPDBA to perform the check, restore and recovery functions.

   This automates the process so that it is faster and there are fewer errors.

4. Consider using fast devices for recovery and adjust the frequency of backups.

   The downtime required for media recovery depends on the time spent for each of the three steps (check, restore, recover). This in turn depends largely on the devices used for restore and also on the backup scheme used. The less time you can afford to spend on recovery, the more often you must perform backups and the more important it is to use fast devices for backup and restore.

5. If available, use parallel recovery.

   On multi-processor hardware, consider using parallel recovery. However, on a single-processor machine, use the standard serial recovery procedure.

6. Only restore and recover what is absolutely necessary.

   Restore and recover only damaged files if a recovery to the current point in time is planned. Recovery is slower if more files than necessary are restored, because more processing has to be repeated.

Result

You can restore your database as quickly and efficiently as possible, so avoiding downtime caused by a failure with loss of database data.

See also:

SAP DBA (in SAPNet)

Upgrade with Oracle

Purpose

Upgrades of the database management system are usually done in line with an R/3 upgrade. An Oracle upgrade might involve one or more of the following activities:

- Upgrade of the Oracle software
- Changes to the Oracle Data Dictionary

  For example, executing scripts to create updated versions of Data Dictionary views

- Changes to Oracle internal structures

  For example, changes in the structure of database blocks. Changes to internal structures usually imply an upgrade of the Oracle software and changes to the Oracle data dictionary as a prerequisite. To make the changes to internal structures, you can usually perform either of the following:

  – A migration, where the changes are made when objects are accessed the first time after the upgrade

  – A full database export, recreate the database, and import

Most upgrades include an upgrade of the Oracle software and changes to the R/3 System Data Dictionary only. The elapsed time to expect for a database upgrade as part of an R/3 upgrade is around one hour. If upgrading to a new R/3 release or doing a large upgrade, this is a minor part of the overall upgrade. The database can not be used for normal operation during an upgrade.

> For up-to-date information on upgrading the Oracle database, you can use the alias “instguides” in SAPNet. Enter the following in the address line of your web browser:

```
http://sapnet.sap.com/instguides
```

Process Flow

1. You read the upgrade documentation.

   This helps to avoid unexpected problems during the upgrade.

2. You plan the upgrade carefully.

   Rather than simply starting the upgrade and hoping for the best, plan the procedures involved in advance. Make sure that all the resources (that is, people and equipment) are available.

3. You rehearse the upgrade using a test system.

   The best preparation of all is to rehearse the upgrade fully using a test system that is as similar to the production system as possible.

Result

You can now upgrade the database with the minimum possible downtime.

See also:

Migrating/Upgrading to Oracle Version x: UNIX (in SAPNet)

Migrating/Upgrading to Oracle Version x: Windows NT (in SAPNet)

High Availability for the Informix Database

Purpose

This section looks at database administration for the Informix database and makes specific recommendations on improving availability. Wherever possible, SAP recommends you to use SAPDBA [Page 159], which is the tool specially designed for administering your database.

For up-to-date information on Informix with the R/3 System, you can use the alias "dbainf." Enter the following in the address line of your web browser:

```
http://sapnet.sap.com/dbainf
```

Process Flow

6. You work out your approach to backing up your database. You must do this before you start production with the database. When using the database for production, you must back up the database as regularly as possible.

   Refer to Backup with Informix [Page 69].

7. You manage the space in your database. You do this both before you start production with the database and during production as the database grows.

   Refer to Space Management with Informix [Page 71].

8. You upgrade your database when required.

Refer to Upgrade with Informix [Page 81].

9. You recover your database if a failure occurs with data loss.

   Refer to Recovery with Informix [Page 80].

10. You consider using various tools and services offered as standard by SAP to increase the availability of your database:

    – SAPDBA

    – [Page 159]Computing Center Management System (CCMS) [Page 166]

    – GoingLive and EarlyWatch [Page 169]

7. You consider using advanced products and services to increase the availability of your database:

   – DB Reconnect [Page 174]

   – Informix High-Availability Data Replication (HDR) [Page 193]

   – Switchover Software [Page 216]

8. You review your procedures to increase the availability of your database. Refer to High Availability Procedures at Your Site [Page 247].

Result

Your Informix database is more available for production use.

See also:

BC R/3 Database Guide: Informix [Ext.]

Documentation in SAPNet

Backup with Informix

Purpose

You must always have a recent and consistent copy of database data that can be used to recover the database in the event of failure involving data loss. Informix provides the following tools for database recovery [Ext.] (including backup):

- `ON-Bar` (the most recent tool) together with a storage manager such as the Informix Storage Manager (ISM)
- `ON-Archive`
- `ontape`

For an introduction to backup with Informix, see the following:

- Informix Data Recovery [Ext.]
- Informix Whole-System and Storage-Space Backups [Ext.]
- Informix Logical-Log Backup [Ext.]

When ON-Bar was released, Informix changed the terminology to bring it into line with other databases. The term "archive" from ON-Archive and ontape is no longer used with ON-Bar. Instead, "database backup" is used.

In this section, we:

- Use "database backup" to include the term "archive."
- Concentrate on ON-Bar, giving cross-references only to ON-Bar sections in the documentation for Informix issued by SAP.

Process Flow

1. You develop a robust and secure backup strategy. Refer to:

   – Approach to Database Backup (ON-Bar) [Ext.]

   – Approach to Logical-Log Backup (ON-Bar) [Ext.]

   Consider factors such as the frequency of backups, the level of database backups, whether you want to use the Database Planning Calendar of the Computing Center Management System (CCMS) to schedule backups, whether you want to perform manual or continuous backups, how long you intend to keep the storage media with the backups before they are overwritten, whether you want to use unattended backups, and so on.

   SAP recommends you to:

   - Use full (level-0) database backups as often as possible, preferably once per day. Remember that you cannot use incremental (that is, level-1 or level-2) database backups independently of the corresponding level-0 backup. Full (level-0) archives are also required in special circumstances, for example, if you alter the configuration of the Informix server.
   - Avoid partial backups. This is because they are not independent in two ways. First, they require a backup of the logical log data from the time when the archive was taken. Second, from the recovery point of view, it makes little sense to only restore part of the database because R/3 is so integrated.

2. If necessary, you consider parallel database backup (known as a "dbspace backup" with ON-Bar) to speed up the backup. This can help you to avoid affecting the performance of your production system during working hours. ON-Bar and all storage managers fully support parallel backup. Informix allows you to create backups while the system is fully online and no extra risk is incurred.

3. Before you create a database backup, you check physical database consistency [Ext.]. Be sure to always have at least one database backup for which you have checked the consistency. If a backup is inconsistent, you might be unable to use it to restore the database.

4. You check and review your backup strategy. For example, this might include testing that you can use the backups to successfully restore the database from time to time, using a copy of the production system on a separate machine.

SAP recommends you to pay special attention to logical-log backup, because this often causes problems. If the logical log fills, the database stops processing, so stopping production with your R/3 System. For more information, see Preventing Emergency Logical-Log Backup [Ext.].

Result

You always have an up-to-date backup of your database. This allows you to quickly recover the database [Page 80] in the event of a failure involving data loss. Therefore, the availability of your R/3 System is increased because downtime due to the absence of a suitable database backup is kept to a minimum.

See also:

ON-Archive for Data Recovery [Ext.]

ON-Bar for Data Recovery [Ext.]

ontape for Data Recovery [Ext.]

Archive (ON-Archive and ontape) [Ext.]

Database Backup (ON-Bar) [Ext.]

Logical-Log Backup [Ext.]

Emergency Logical-Log Backup [Ext.]

Space Management with Informix

Purpose

This section looks at space management (including reorganization) of database objects (that is, tables and dbspaces). If you neglect space management, this can lead to downtime due to normal database growth when database objects fill up. If this happens, applications cannot write to the database and you have to quickly make more space available. You might need to bring down the R/3 System to tune and configure the database. Therefore, it is much better to anticipate the problem by monitoring and pro-actively managing the disk space in your database.

SAP recommends you to manage space on your Informix database using the Computing Center Management System [Page 166] (CCMS) in the R/3 System and SAPDBA [Page 159]. You need to monitor regularly and occasionally take timely action to avoid the problem leading to downtime.

The problems of extent overflow, dbspace overflow, and a high number of extents for a table are more likely to occur in some dbspaces than in others. The following dbspaces in an R/3 System are most likely to cause problems:

Dbspace	Comment
`PSAPBTAB`	Transaction data tables. Objects in this dbspace might expand very rapidly.

PSAPSTAB	Master data tables. Objects in this dbspace might expand very rapidly
PSAPCLU	Clustered tables, such as financial tables. Objects in this dbspace might expand very rapidly
PSAPPOOL	Pool tables, containing customization tables
PSAPPROT	Spool (that is, print) requests, protocols

Pay special attention to the following dbspaces in certain circumstances:

- PSAPSOURCE, PSAPLOAD and PSAPDDIC

  If you develop many new ABAPs, monitor these dbspaces closely.

- PSAPTEMP

  If you are running a large import or reorganization, you should closely watch this dbspace since it is used to store temporary data.

Process Flow

7. You manage extents [Page 72].
8. You manage dbspaces [Page 75], especially problematic ones.
9. You manage tablespaces [Page 76].
10. You manage chunks [Page 77].
11. If necessary, you reorganize tables, indexes, or dbspaces [Page 78].

    SAP recommends you to avoid reorganizations wherever possible. You can achieve this by correct configuration and sizing of the database together with proper monitoring.

    Only reorganize with a clear justification. One such justification is that database objects might soon reach the maximum available size limits (that is, number of extents).

Result

By managing the space in your Informix database, you can avoid unplanned downtime due to database objects filling up.

See also:

Management of Informix Database Growth [Ext.]

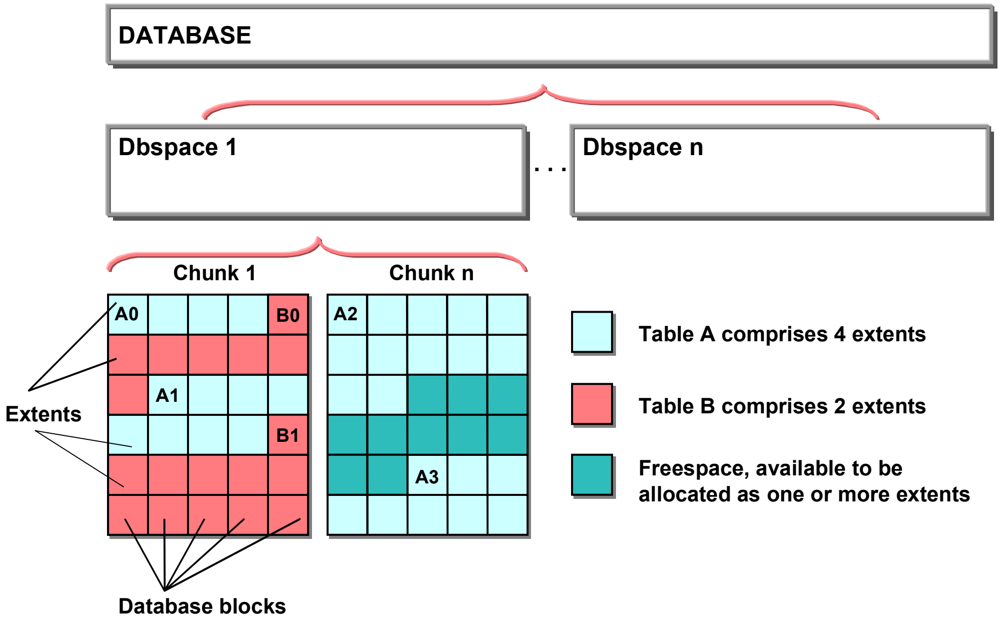
Managing Extents (Informix)

Use

You need to manage the extents of your Informix database to avoid "extent overflow". This occurs when a table reaches the maximum number of extents allowed.

Tables with a high number of extents can threaten continuous availability of the system, because they might in future cause downtime for reorganization (that is, if growth continues and an extent overflow occurs). The following diagram illustrates how storage is allocated in an Informix database:

Storage Allocation for Informix


DATABASE
Dbspace 1
. . .
Dbspace n
Chunk 1
Chunk n
A0
B0
A1
B1
A2
A3
Extents
Database blocks
Table A comprises 4 extents
Table B comprises 2 extents
Freespace, available to be allocated as one or more extents

Although some impact on performance is possible, extent overflow does not usually cause performance degradation because the R/3 System has been optimized to use indexes (however, a high number of extents might cause problems for batch access). Even tables with very many extents (for example, 150) might not necessarily suffer performance degradation. Furthermore, the Informix server has the following features to minimize this problem:

- Extent fusion

  Whenever possible, a new extent is allocated adjacent to an existing extent with the advantage that the number of extents is not increased and the table storage remains contiguous.

- Extent size doubling

  When a table is repeatedly extended, the next extent size is automatically doubled at regular intervals, that is, every 16th time that the table is extended with a distinct (that is, non-contiguous) new extent.

The following factors lead to a large number of small extents being allocated, which in turn can lead to extent overflow:

- Insufficient next extent size

  Every table has a next extent size defined for it. If the next extent size is too small in relation to the growth rate of the table, then the table must be extended using a large number of small extents.

- Insufficient gap size within chunks of a dbspace

  The chunks comprising a dbspace have gaps that can be used to satisfy the requirement for new extents on tables that need to be extended. If the dbspace in which the table resides has insufficient gaps and the table needs a new extent, there might not be enough contiguous space available to satisfy the requirement completely. Therefore, the table must be extended using a large number of small extents.

The interaction between the next extent size (this differs between tables) and the gap size (differs between chunks allocated to a dbspace), determines whether tables are able to grow normally, that is, with the full next extent size being allocated.

The limit for the maximum number of extents that one table can have (see table below) depends on the page size of the operating system and, to a lesser extent, on certain parameters specific to a table. If an object reaches the limit for the number of extents, you **must** reorganize it.

Page Sizes and Number of Extents for Informix on Different Platforms

Platform	Page Size (KB)	Maximum Number of Extents
HP, SUN, DEC, SNI	2	200 – 230
IBM, NT	4	400 – 460

Procedure

1. Monitor tables for number of extents. Refer to Managing Extents (Informix) [Page 72].

   Look for the following:

   – Tables with a high number of extents allocated, to make sure that the maximum limit for a table is not reached. Refer to Analyzing Tables by Fill Level, Size, and Extents with SAPDBA [Ext.]. The Computing Center Management System (CCMS) [Page 166] also provides information about extent allocation as well.

     Reorganize tables with a high number of extents (see next step).

   – Tables for which the normal next extent size could not be satisfied. Refer to Analyzing Tables for Critical Next Extent Size with SAPDBA [Ext.].

   Extend the dbspace in which such tables reside to avoid future problems. Refer to Managing Dbspaces (Informix) [Page 75].

2. Reorganize tables with a high number of extents.

   This lets you consolidate the table into a small number of initial extents and also increase the next extent size for future extensions.

   With Informix databases, you do **not** need to regularly reorganize. However, if a table has a high number of extents, this is a good reason to reorganize it.

   Refer to Reorganizing a Single Table with SAPDBA [Ext.].

Result

You avoid extent overflow and so avoid downtime for your Informix database and the R/3 System.

Managing Dbspaces (Informix)

Use

You need to manage the dbspaces of your Informix database to avoid "dbspace overflow". This occurs when there is no more freespace in the chunks allocated to the dbspace. Therefore, downtime of the R/3 System is caused because it is impossible to allocate new extents to objects in the dbspace, such as tables or indexes. You normally correct the problem by extending the dbspace (that is, adding a chunk). You can do this with SAPDBA while the database is online.

Another aspect to managing dbspaces is to make sure that the distribution of dbspaces across disk drives is optimal. This is best done during the installation phase.

Before dbspace overflow is reached, it is likely that extents cannot be allocated normally, that is, with the normal extent size required by larger objects. Therefore, extension of such objects leads to a large number of small extents as space in the dbspace becomes increasingly fragmented. You can detect and fix the problem at this stage before actual downtime occurs.

The following factors make dbspace overflow more likely:

- Operations that greatly extend a table

  Certain operations (for example, client copy or batch input) might extend a table excessively. Therefore, plan these operations with care.

- Unused space in a table

  An indirect cause of dbspace overflow is if tables in a dbspace contain a lot of unused space (this occurs, for example, after a table has many inserts followed by many deletes). If you really need to reclaim the space, the best solution is to reorganize the affected table(s) with SAPDBA. Refer to Reorganization of Tables, Indexes, and Dbspaces with SAPDBA [Ext.].

Procedure

1. Set up dbspaces optimally.

   Locate the three critical dbspaces (`logdbs`, `physdbs`, and `rootdbs`) on separate disk drives to reduce contention. Also, mirror the critical dbspaces to increase their availability. This is best done during the installation phase.

2. Use the R/3 disk space configuration program.

   This helps you to estimate table and dbspace growth for each business application. Therefore, you can configure your disks optimally from an early stage.

3. Monitor freespace in dbspaces.

   When freespace in a dbspace is approaching zero, increase the size of the dbspace in time to accommodate further growth. Refer to Listing Dbspaces with SAPDBA [Ext.]. The Computing Center Management System (CCMS) in the R/3 System also displays information on freespace.

4. Monitor dbspaces for chunks with inadequate gaps.

   To make sure that dbspaces have chunks with adequate gaps, regularly monitor the dbspaces with SAPDBA. Refer to Listing Dbspaces with SAPDBA [Ext.].

5. Use the alert monitor in the CCMS to identify when a dbspace needs to be extended.

Refer to Extending a Dbspace with CCMS (Informix) [Ext.]. This lets you extend the dbspace immediately.

6. Monitor dbspaces for rapidly growing objects, that is, objects that need more and more extents in a short time.

   Refer to Analyzing Tables by Fill Level, Size, and Extents with SAPDBA [Ext.].

7. Monitor disk space at operating system level.

   In addition to using the CCMS and SAPDBA, monitor available disk space at the operating system level. Plan for additional disks in time to accommodate dbspace growth.

8. Take the following actions if necessary to fix space problems in dbspaces:

   a. Add a chunk with SAPDBA [Ext.] (this extends the dbspace)

      Additional disk space is required, but the database remains fully online. This is preferable to the next action, reorganization.

      However, note that this does not resolve the problem of tables with a high number of extents. If you have this problem, you have to reorganize either individual tables or the dbspace and all its tables.

   b. Reorganize a dbspace and its tables with SAPDBA [Ext.]

      This lets you reorganize the dbspace and all its tables, and allocate more space to the dbspace if necessary.

      With Informix databases, you do **not** need to reorganize regularly. Make sure you have a good reason to reorganize. For more information, see Reorganizing Objects (Informix) [Page 78].

Result

You avoid dbspace overflow and so avoid downtime for your Informix database and the R/3 System.

See also:

Dbspaces with SAPDBA [Ext.]

Managing Tablespaces (Informix)

Use

You need to manage the tablespaces of your Informix database to avoid "tablespace overflow". This occurs when a tablespace reaches the maximum number of pages allowed.

The term tablespace refers to the total storage allocated to a table or table fragment (if the tables is fragmented). The maximum size allowed for a tablespace is determined by the largest page number that can be accommodated in a rowid. Since the page number in a rowid cannot exceed 16,277,215, this is the upper limit of the number of pages that a single fragment can contain.

Procedure

1. Monitor pages per tablespace for very large tables.

   Refer to Analyzing a Tablespace with SAPDBA [Ext.].

2. Take action to correct the problem if the tablespace is about to overflow, as follows:

   – Fragment the table

   – Detach the index from the tablespace (that is, recreate it in another dbspace).

   For more information, see the Informix documentation.

Result

You avoid tablespace overflow and so avoid downtime for your Informix database and the R/3 System.

Managing Chunks (Informix)

Use

You need to manage the chunks of your Informix database to avoid "chunk overflow". Chunks are the physical storage units used to build dbspaces. To extend a dbspace, you add chunks at the physical level.

If a large number of small chunks are created, it is possible that the Informix server limit for the maximum number of chunks in the database could be reached. The limit is either 2048 or the maximum number of open files per process, as allowed by your operating system (refer to the appropriate documentation). However, note that the limit is rarely reached.

If chunk overflow occurs, you have to reorganize and downtime occurs. Therefore, monitor your chunks to avoid this situation.

Procedure

1. Monitor the number of chunks for your database regularly.

   Refer to Listing Chunks with SAPDBA [Ext.].

2. When creating new chunks, make sure that they are large enough.

   Make sure that chunks are large enough to accommodate foreseeable growth in the dbspace. Otherwise, you have to repeat the procedure in the near future and the result is a large number of small chunks, with the danger of chunk overflow.

   Refer to Adding a Chunk with SAPDBA [Ext.].

3. If approaching chunk overflow, reorganize problem dbspaces with their tables.

   With Informix databases, you do **not** need to regularly reorganize. However, if you are approaching the Informix server limit for the maximum number of chunks, this is a good reason to reorganize.

   Use SAPDBA to identify dbspaces that consist of more than one chunk (see the first step above). Then you can use SAPDBA to reorganize these dbspaces so that they have fewer chunks, if possible just one chunk. Refer to Reorganizing a Dbspace and Its Tables with SAPDBA [Ext.].

Result

You avoid chunk overflow and so avoid downtime for your Informix database and the R/3 System.

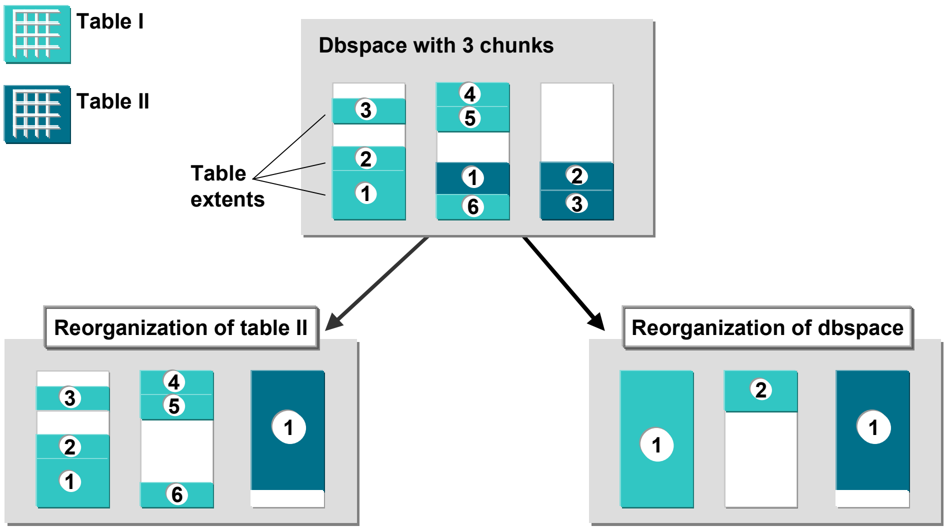
Reorganizing Objects (Informix)

Use

You can use SAPDBA to reorganize tables, indexes, and dbspaces. SAP recommends you **not** to regularly reorganize Informix databases. Therefore, make sure you have a clear reason for reorganization, which usually causes downtime (depending on the type of reorganization).

The following graphic illustrates the types of reorganization possible:

Types of Reorganization with Informix



The reorganization on the left is at table level, resulting in optimal storage of the reorganized table in a single extent. The one on the right is at dbspace level, resulting in optimal storage of all objects in the reorganized dbspace.

Prerequisites

Be sure to monitor extents, dbspaces, and chunks. This helps you to avoid reorganization where possible and identify when it is required. Refer to the following:

- Managing Extents (Informix) [Page 72]
- Managing Dbspaces (Informix) [Page 75]
- Managing Chunks (Informix) [Page 77]

It is best to detect storage problems early:

- Set up the database optimally and reorganize early when the database is small. You can use GoingLive [Page 169] to help you during the implementation phase. In general, reorganize early while the database is small. This can be done based on early usage patterns and future projection. For example, perform reorganizations immediately after an upgrade.
- You can use EarlyWatch [Page 169] to detect storage problems early. EarlyWatch also gives recommendations to adjust database parameters, so helping to avoid reorganization completely.

Procedure

1. Identify the problem, making sure that reorganization is the correct way to solve it.
   - Extent overflow

     If a dbspace has many tables with a large number of extents, it is often best to reorganize the dbspace, or particular tables within it. Extent overflow normally indicates that the dbspace is also short of space. Therefore, you can also extend the dbspace during reorganization.

     If the dbspace is simply short of space, you can add a chunk with SAPDBA [Ext.] to extend the dbspace.
   - Dbspace overflow and dbspace contains many tables with a high number of extents

     Reorganize the dbspace with SAPDBA. You can also extend the dbspace during this procedure.
   - Relocation of large tables

     If you want to relocate large tables to a dbspace of their own (not generally recommended but sometimes a good idea), reorganize the tables with SAPDBA, specifying a new target dbspace.

     You might then want to reduce the size of the original dbspace. For this you need to reorganize the dbspace.
   - Reclaim space from large tables that have shrunk

     Before you reclaim the space, be sure that the tables will not grow again. If you are sure of this, you can use SAPDBA to reorganize the tables with smaller extent sizes.
   - Chunk overflow

     An unusual reason for reorganization is that the dbspace has almost reached the maximum number of chunks. In this case, reorganize the dbspace with SAPDBA.
2. Choose the correct kind of reorganization:

   Reorganize at the correct level, that is, at table or index level, or dbspace level.

   SAP advises you to reorganize individual tables and indexes rather than entire dbspaces, if possible. In general, use SAPDBA for reorganization, with the R/3 System down. However, you can reorganize an index or extend a dbspace (that is, add chunks to it) when the R/3 System is up. You can also perform a minimal table reorganization with the R/3 System up.

   You can reorganize tables and dbspaces as follows:

- Reorganizing a Single Table with SAPDBA [Ext.]
- Reorganizing a Group of Tables with SAPDBA [Ext.]
- Reorganizing a Dbspace and Its Tables with SAPDBA [Ext.] (if you want to resize the dbspace afterwards)

3. Reorganize in such a way as to minimize downtime.

   When you decide to reorganize, there are a large number of factors to consider if you want to minimize the downtime. The time taken for data reorganization depends on the database size and the objects to be reorganized and could be several hours. Index reorganizations are usually faster than table reorganizations. As an example, the reorganization of an index of about 400 MB in size might take less than 10 minutes while for a 1 GB table around 1 hour might be needed (depending on hardware).

   💡

   Prepare properly for reorganization

   Check using SAPDBA to make sure the required disk space is available and relevant parameters are correctly set before shutting down the database host. Disk space needs to be available in the database itself for the “insert into select from” or “alter fragment” types of reorganization and in the file system for the “export/import” type. Proper preparation avoids unnecessary downtime due to improper settings or insufficient resources.

4. Schedule reorganizations together with database backups (`ON-Bar`) or archives (`ON-Archive` or `ontape`)

   You must back up or archive the database before reorganization. Therefore, it makes sense to schedule reorganizations immediately after regular archives or database backups to avoid the need for a separate archive or database backup. It is also a good idea to group a number of reorganizations together to reduce total downtime.

Result

Reorganization helps you to solve problems that might lead to downtime, such as extent overflow, dbspace overflow, or chunk overflow. However, be sure to continue monitoring the database so that you can anticipate future problems before they cause downtime.

See also:

Reorganization with SAPDBA [Ext.]

Recovery with Informix

Purpose

To minimize downtime with your Informix database, you must make sure that you can quickly recover the database in the event of a failure with loss of production data. With Informix, recovering the database includes routinely backing up the database and restoring the database after a failure. This section concentrates on database restore. For more information, see Informix Restore [Ext.].

Informix provides the following tools for database recovery [Ext.] (including restore):

- `ON-Bar` (the most recent tool) together with a storage manager such as the Informix Storage Manager (ISM)
- `ON-Archive`
- `ontape`

When `ON-Bar` was released, Informix changed the terminology to bring it into line with other databases. The term "archive" from `ON-Archive` and `ontape` is no longer used with `ON-Bar`. Instead, "database backup" is used.

In this section, we use "database backup" to include the term "archive".

Process Flow

1. You make sure that you have suitable and recent database and logical-log backups. This is absolutely essential if you want an up-to-date restore of the database. Refer to Backup with Informix [Page 69].
2. You work out a recovery strategy, including an approach to database restore.
3. You test database restore using a test system as similar as possible to your production system.

Result

You can restore your database as quickly and efficiently as possible, so avoiding downtime caused by a failure with loss of database data.

See also:

Restore [Ext.]

Upgrade with Informix

Purpose

You can take measures to reduce the downtime caused by upgrade of the Informix database. Upgrading the Informix database is normally performed before an R/3 System upgrade. The downtime involved depends on the level of upgrade you are performing. An upgrade from version 7.00 X to version 7.00 Y can take as little as 15 minutes whereas an upgrade from Informix version 6 to version 7 takes much more time since internal data structures are altered.

For up-to-date information on upgrading the Informix database, you can use the alias "dbainf" or the alias “instguides” in SAPNet. For example, enter the following in the address line of your web browser:

```
http://sapnet.sap.com/instguides
```

Process Flow

4. You read the upgrade documentation.

   This helps to avoid unexpected problems during the upgrade.

5. You plan the upgrade carefully.

   Rather than simply starting the upgrade and hoping for the best, plan the procedures involved in advance. Make sure that all the resources (that is, people and equipment) are available.

6. You rehearse the upgrade using a test system.

   The best preparation of all is to rehearse the upgrade fully using a test system that is as similar to the production system as possible.

Result

You can now upgrade the database with the minimum possible downtime.

See also:

Upgrade to Version x of the Informix Dynamic Server: UNIX (in SAPNet)

Upgrade to Version x of the Informix Dynamic Server: Windows NT (in SAPNet)

High Availability for the SAP DB Database

Purpose

This section looks at database administration for the SAP DB database and makes specific recommendations on improving availability.

Process Flow

11. You work out your approach to backing up your SAP DB database. You must do this before you start production with the database. When using the database for production, you must back up the database as regularly as possible.

    Refer to Backup with SAP DB [Page 83].

12. You manage the space in your SAP DB database. You do this both before you start production with the database and during production as the database grows.

    Refer to Space Management with SAP DB [Page 86].

13. You upgrade your database when required.

    Refer to Upgrade with SAP DB [Page 89].

14. You recover your database if a failure occurs with data loss.

    Refer to Recovery with SAP DB [Page 87].

15. You consider using various tools and services offered as standard by SAP to increase the availability of your database:

    – Computing Center Management System (CCMS) [Page 166]

- GoingLive and EarlyWatch [Page 169]

  Wherever possible, use the specially designed tool R/3-Database Manager (DBMGUI) [Page 163] to administer your database. There is also a command line interface for this tool.

8. You consider using advanced products and services to increase the availability of your database:
   - DB Reconnect [Page 174]
   - Switchover Software [Page 216]
9. You review your procedures to increase the availability of your database. Refer to High Availability Procedures at Your Site [Page 247].

Result

Your SAP DB database is more available for production use.

See also:

SAP DB documentation for Version 7.2

Backup with SAP DB

Purpose

SAP DB uses the following operational modes for backup:

- WARM (that is, online)
- COLD (that is, restricted single user administration)

You can back up the database and log in WARM or COLD mode. Only a backup in COLD mode causes system downtime.

There are different kinds of backup, including a complete data backup, an incremental data backup (backup of the updated pages only), a backup of the entire unsaved log, and an automatic backup of the full log segments.

Prerequisites

Distinguish between backup in operational mode COLD and WARM:

- COLD backup

  Backup activities in operational mode COLD can be started with the Database Manager (DBMGUI) [Page 163] or the tool xbackup only. The major disadvantage of these backups is that the entire database system is unavailable while the system is in cold mode This affects the backup types complete data backup, incremental data backup, and log backup. Backups in operational mode COLD are only consistent if the database system is first shut down normally (that is, using *Instance* → *Shutdown* → *Cold* → *OK* in the Database Manager (DBMGUI).

Log backup in operational mode COLD (as in WARM mode) only backs up the area of the log that was **not** previously backed up with *Instance* → *Backup* → *Log* or a switched on *AutoLog*. Therefore, the time taken to back up the log depends on the size of the log not yet backed up.

- WARM backup

  All backup activities in operational mode WARM can be started with the Computing Center Management System [Page 166] (CCMS) or the Database Manager (DBMGUI). Before doing a backup from CCMS, you must define the backup media with the Database Manager (DBMGUI).

  A backup in operational mode WARM is more appropriate for large database systems where availability is crucial. The database system is available for normal use while the backup is running. The backup types are complete data backup, incremental data backup, and log backup.

  Log backup in operational mode WARM (as in COLD mode) only backs up the area of the log that was **not** previously backed up with *Instance* → *Backup* → *Log* or a switched on *AutoLog*. Therefore, the time taken to back up the log depends on the size of the log not yet backed up.

You do not necessarily need a data backup in operational mode COLD to obtain a consistent copy of serverdb data. Data backups in operational mode WARM can also be consistent, but **only** when a checkpoint is made at the start of the backup (for example, choose *Instance* → *Backup* → *Complete/Incremental* → *Migration (with checkpoint)* in the Database Manager (DBMGUI) to perform a consistent data backup). Consistent data backups can be later used to migrate or to copy the database instance. In normal recovery situations, all data – that is, including recent changes in data – must be restored, which means that the data backup as well as the log backup have to be restored.

In terms of performance, there is no difference between backups in operational mode COLD or WARM. However, there might be a delay until a backup with checkpoint in operational mode WARM can start because it has to wait until all open transactions are finished (for example, batch jobs often do not write commits for long periods).

Process Flow

1. You work out your data backup strategy.

   You work out a comprehensive data backup strategy before you start productive operation of your database. For example, perform a complete data backup at the weekend, and an incremental data backup every day at midday or overnight. Use the DBA Planning Calendar in the CCMS to schedule these data backups.

   ⚠

   A data backup does **not** also back up the log.

2. You avoid the log filling up.

   To do this, you can enable the AutoLog feature to back up the log segments as soon as they are full. Use the DBA Planning Calendar in the CCMS to enable *AutoLog*. Log backups never write checkpoints, so can always be started without delay.

3. You determine the type of backup to be used.

In general, it is best to use backup in operational mode WARM when possible for regular backups since this reduces downtime. Use operational mode COLD for backup only when for a special reason operational mode WARM is impossible or the normal database operation is meant to be interrupted.

You use backup in operational mode WARM if your system cannot tolerate downtime or if your serverdb is too large to be backed up in the time available during scheduled downtime.

4. With backups in operational mode WARM, you decide if you need a checkpoint

   You can choose between a data backup with and without a checkpoint in operational mode WARM. You should normally perform a data backup without a checkpoint to save time. You must also perform a log backup in case your database fails and needs to be recovered.

   If you start a backup with checkpoint for migration purposes, you make sure no batch jobs are running.

5. You consider incremental data backup at page level.

   If you have decided that a complete data backup takes too long, use an incremental data backup, which only backs up updated pages. For example, perform a complete data backup at the weekend and incremental data backups every day. You can choose between incremental data backup at page level with checkpoint and without checkpoint.

   An incremental data backup with checkpoint can be helpful if you want to copy the database instance. This avoids having to recover the log backups after you have restored the complete and incremental database backups.

6. You use parallel media for complete or incremental data backups where possible, since this takes less time. However, the log can **not** be backed up using parallel media.

7. You choose the best method for log backup, according to your requirements.

   - You use the autosave log feature in the CCMS or Database Manager (DBMGUI) to save the unsaved log segments. If a log segment is complete, the log backup is started automatically. After the backup, the log segments are released. Therefore, the log fill level decreases.

   - You use the WARM log backup in the CCMS or Database Manager (DBMGUI) to save the entire unsaved portion of the log.

8. You keep more than one generation of backups. If a complete data backup is faulty, you then still have an older version available that could be used – together with subsequent incremental data backups and log backups – to fully restore the database.

   The time required for backup depends on the machine load and the throughput of the backup devices (for example, disks are faster than tapes).

See also:

SAP DB documentation for Version 7.2

Space Management with SAP DB

Purpose

Although this might not seem to be relevant to high availability with SAP DB, it can, if neglected, lead to downtime because the database runs out of space. It is far better to predict this problem than to recover from it. You can use the Database Manager (DBMGUI) [Page 163] to manage space in the SAP DB database.

Prerequisites

In the following scenarios, a SAP DB database stops processing, so that R/3 applications can no longer continue:

- Database Usage Level 100%

  This means there is no more space in the data area of the serverdb. You must quickly provide more space by adding a new data area – called a "devspace" – to the database. You can normally add a new devspace in WARM mode using the Database Manager (DBMGUI).

- Log Usage Level 100%

  During normal database operations (that is, in WARM mode), you can back up the log without affecting R/3 applications. If you do not back up the log, messages are sent to the `knldiag` file and to the alert monitor in the R/3 System. If the error message for a full log appears, R/3 applications cannot continue until you have backed up the log.

  To avoid the log filling up, use the automatic log save feature. To do this, use Computing Center Management System [Page 166] (CCMS) to enable the autosave log backup or choose *Instance* → *Backup* → *AutoLog* → *On* in the Database Manager (DBMGUI). This means that, as soon as a log segment is full, it is backed up to the specified device.

Process Flow

1. You regularly monitor the database and log usage levels.

   The recommended way to manage space on your SAP DB database is using the CCMS or Database Manager (DBMGUI). For routine monitoring, use the alert monitor in the CCMS. The CCMS and Database Manager (DBMGUI) present information in different ways and you should familiarize yourself with both to obtain the best possible picture of the state of database objects.

2. You install your database optimally for production.

   Optimal installation of your database in view of the disks available and anticipated requirements can prevent future problems. This means correctly sizing the serverdb (for example, size and number of data devspaces), using large values when setting the database parameters. This allows you to add a devspace in WARM mode. Otherwise, you would have to increase the parameters `MAXDATAPAGES` and `MAXDATADEVSPACES`, then stop and start the database, before you could add a new data devspace.

3. You monitor the database usage level.

   You should closely watch the expansion of the data in the database. You can use the CCMS or Database Manager (DBMGUI) to monitor the fill level.

4. If required, you add a data devspace.

As the result of your monitoring, you might decide to add a new devspace to the serverdb. Use the Database Manager (DBMGUI) to do this before the situation becomes critical. First, check whether the parameter `MAXDATADEVSPACES` is large enough to add a new devspace. Then check `MAXDATAPAGES` to see whether its size has been calculated from the number of pages of all data devspaces. If these parameters are large enough, you can add the devspace in WARM mode. Finally, make the new data devspace available. Under UNIX, grant the rights 660 to the owner `sqd<sapsid>`, group `dba`, and set the link to the directory

5. You monitor the log usage level.

   You can observe the log usage level with the CCMS or Database Manager (DBMGUI). Make sure the log usage level does not approach 100%. Once you have backed up the log, it is automatically free for re-use.

6. You back up log segments.

   If you back up the log segment, the log segment is not closed by a checkpoint. Configure the log segment size as follows using the Database Manager (DBMGUI):

   0 < log segment size <= 50 % of the whole log area

   If you do not configure the log segment size, the default value is a third of the whole log area. If you configure a value higher than 50 %, the default value is 50 % of the whole log area.

   You can back up log segments automatically. To do this, enable the automatic log backup feature using the CCMS or Database Manager (DBMGUI). As soon as a log segment is full, the database backs up that log segment and releases it for further use.

See also:

SAP DB documentation for Version 7.2

Recovery with SAP DB

Purpose

To minimize downtime with your SAP DB database, you must make sure that you can quickly restore the database in the event of a system failure in which productive data is lost. A database can only be restored in COLD operation mode using the Database Manager (DBMGUI) [Page 163].

Prerequisites

The sequence you must follow when restoring data is that you restore:

1. The last complete data backup, done in either WARM or COLD mode.
2. Any subsequent incremental data backups.
3. Any necessary log backups. The Database Manager (DBMGUI) prompts you for the correct sequence of logs.

Only the first step is mandatory, the second step can be performed if the relevant data is available, and the third step is performed if necessary.

Process Flow

1. You design a recovery strategy and test it.

   Think in advance what must be done if the database suddenly fails and the data must be restored. Plan the steps carefully and rehearse the procedure using a test system, if possible. This substantially reduces downtime.

2. You use the recovery report in the R/3 System.

   You can obtain a list of the data backups (complete and incremental) and log backups required for a recovery by using the recovery report in transaction `DB12` in the R/3 System. You should print this report regularly, so that you have an up-to-date list of the required backups, if a recovery is later necessary.

3. You always have the best possible set of data available.

   This means that the users need not spend extra time on manually restoring lost data, if a recovery is necessary.

4. You make sure that you can quickly identify the tapes required.

   To recover the database, you must know exactly which types of backup tapes you need (that is, complete data backup, incremental data backup, or log backup tapes). A data backup (including incremental data backups) can consist of several tapes. You do **not** have to start restoring data with the first backup tape nor do you have to finish with the last one. As long as a tape belongs to the same backup (complete or incremental, whether parallel or not) the sequence of the tapes is not important. You can use the Database Manager (DBMGUI) to display the tape labels.

   However, you must restore the individual data backups in the correct sequence. To restore the log, you must restore the tapes in the correct sequence (that is, in the sequence in which the log backups were performed).

5. You choose the fastest recovery strategy.

   Start with the recovery of data backups. It is normally faster to restore incremental backups (that is, updated pages) than to restore the log backup. To reduce the time taken still further, consider using parallel recovery, that is, with several tape devices working simultaneously in parallel. If you are using incremental backups as well as log backups, you must use the incremental backups first, followed by the log backups.

   When recovering a database, you always need to restore log backups too. Sometimes the relevant log entries are still on the log devspace (that is, they have not yet been backed up). In this case the log recovery does not take place explicitly. The relevant log entries are restored while the database instance is restarted. If there are log entries required for recovery but these are no longer in the log devspace, the system guides you accordingly.

See also:

SAP DB documentation for Version 7.2

Upgrade with SAP DB

Purpose

Upgrading the SAP DB database is normally performed before an R/3 System upgrade and obviously causes extra downtime for the R/3 System. The SAP DB upgrade always involves the following activities:

- Saving old DBROOT (that is, manually)
- Upgrading the SAP DB software
- Upgrading the SAP DB system tables

Process Flow

1. You read the upgrade documentation.

   You familiarize yourself with the upgrade documentation so that you know what to expect during the update.

2. You plan the upgrade.

   It is risky to start the upgrade without knowing the procedure, simply hoping that everything runs well. A lack of planning often leads to unexpected problems that can considerably increase the downtime of the R/3 System.

3. Before starting the upgrade, you stop the R/3 System and the database.
4. You perform the upgrade according to the documentation.

   The SAP DB software is upgraded with the DBUPDATE tool. DBUPDATE reads software from CD to disk and starts the installation script `bin/x_install` For more information, see Database Manager (DBMGUI) [Page 163].

5. After the upgrade, you start the database in WARM operational mode. With the Database Manager (DBMGUI) the system tables are loaded.
6. You restart the R/3 System.

High Availability for the DB2 UDB Database

Purpose

This section looks at database administration for the DB2 Universal Database (abbreviated to DB2 UDB) and makes specific recommendations on improving availability. With the introduction of DB2 version 5 the database has been renamed from DB2 common server (abbreviated to DB2/CS) to DB2 Universal Database. DB2 UDB runs on UNIX and NT platforms, and the information provided here applies to both platforms, unless explicitly stated.

This section does **not** apply to DB2 for OS/400 or DB2 for OS/390.

Process Flow

16. You work out your approach to backing up your database. You must do this before you start production with the database. When using the database for production, you must back up the database as regularly as possible.

    Refer to Archive and Backup with DB2 UDB [Page 91].

17. You manage the space in your database. You do this both before you start production with the database and during production as the database grows.

    Refer to Space Management with DB2 UDB [Page 91].

18. You upgrade your database when required.

    Refer to Upgrade with DB2 UDB [Page 95].

19. You recover your database if a failure occurs with data loss.

    Refer to Recovery with DB2 UDB [Page 95].

20. You consider using various standard tools and services offered to increase the availability of your database:

    – Computing Center Management System (CCMS) [Page 166]

    – GoingLive and EarlyWatch [Page 169]

    – DB2CC [Page 166]

      💡

      DB2CC, or "Control Center," helps you to perform your tasks in the most precise and efficient way. You can only operate this tool on platforms using Microsoft Windows (that is, with the operating systems Windows 98 and Windows NT). However, from this central point you can administer databases running on both NT and UNIX platforms.

9. You consider using advanced products and services to increase the availability of your database:

    – DB Reconnect [Page 174]

    – Switchover Software [Page 216]

    – Replicated Standby Database for DB2 UDB [Page 197] (known as "HACMP")

    – MIMIX Standby Database for DB2/400 [Page 197]

10. You review your procedures to increase the availability of your database. Refer to High Availability Procedures at Your Site [Page 247].

Result

Your DB2 UDB database is more available for production use.

Archive and Backup with DB2 UDB

Purpose

To achieve high availability with the DB2 Universal Database (abbreviated to DB2) without endangering the recoverability of your system, you need to balance the following factors:

- Routine downtime for offline backups

  Frequent offline backups mean the database is less available for routine production but reduce the recovery time.

- One-off extra time for database recovery after failure

  If your last offline backup is relatively out-of-date when the failure occurs, you need to roll forward more log files, causing extra downtime during recovery.

For a detailed description of archive and backup procedures with DB2 Universal Database (abbreviated to DB2 UDB), refer to your IBM documentation.

Process Flow

1. You get to know the full archive and backup functionality of the DB2CC [Page 166] tool.
2. You practice backup and recovery of the DB2 UDB database on a test R/3 System. In this way you can gain valuable experience to help you in a real emergency.
3. You use parallel backup to reduce the required time.

   DB2 UDB offers parallel backup writers to reduce the time taken to back up the database. By choosing separate backup devices, the time required for a backup decreases substantially. For example, a medium-sized 4-way J30 with SSA disks backs up at a rate of 15 GB per hour to 4 separate disks using 4 parallel backup sessions.

Space Management with DB2 UDB

Purpose

The R/3 System runs on DB2 Universal Database (abbreviated to DB2 UDB) using Database Managed Space (DMS). With this setup, DB2 UDB manages its data in large files or devices called “containers”. Data from several tables, all grouped in a table space, are evenly distributed across the containers of a table space. The R/3 System on DB2 UDB does not generally use System Managed Space (SMS), where each table resides in its own file(s). The exception is temporary table spaces, which use SMS (starting with R/3 Release 4.6).

For more information about DMS, see your IBM documentation.

Prerequisites

Extents and Page Size

All space allocation in the containers of a table space is done in multiples of the page size, called “extents.” Extents are contiguous regions of data within a container. The size of the extents is defined when the table space is created. You cannot change the extent size for a table space once it has been defined. However, you can redefine extent sizes using a procedure called “redirected restore.”

The extent size for a table space also denotes the number of pages that are written to a container before skipping to the next container. The database manager cycles repeatedly through the containers of a tablespace as data is stored. This ensures proper ”striping” of data pages across several containers. Striping is advantageous if each container resides on a different physical disk since the disks can be accessed in parallel. DB2 UDB always allocates new pages in a table space in units of one extent.

The standard page size for DB2 UDB is 4 KB. From version 5 of DB2 UDB, a page size of 8 KB is also possible, and from version 6, page sizes of 16 or 32 KB.

Extent Management

Tables in DB2 UDB table spaces consist of one each of the following objects:

- User table data
- Table index
- Table long field

Each object consists of an extent map that holds information about the allocated extents and the extents holding actual data. Each of the objects can reside in a different tablespace.

With the R/3 System, user table data and user table long data are grouped into one table space. The index objects of the user tables are stored in special index table spaces that only hold index objects. All this information is determined by the `CREATE TABLE` command during table creation.

Each table space also holds a space map that records the status of the extents in the table space. If new extents are needed during `INSERT` operations, free extents are located in the space map. The space map and extent maps of the respective table objects are updated to reflect the changes.

Rows are inserted into the table in a first-fit order. Using the free space map for the table space, the pages are searched for the first available gap that is large enough to hold the new row. When a row is updated, it is updated at its original location unless there is insufficient room left on the 4 KB page to contain it. In this case, a "tombstone record" is created at the original row location that points to the new location of the updated row. If no free page exists for this type of update or for inserts, a whole new extent is allocated. This strategy enables high performance database operation but makes table reorganization necessary under certain circumstances.

With the R/3 System, data table spaces also contain `LONG VARCHAR` data. For each data table space, an index table space is defined during the R/3 database installation. These table spaces only contain index data and do not contain `LONG` data or standard data pages of tables. With R/3, index table spaces are created with different extent sizes than the data table spaces. Both types of table space can have a maximum size of 64 GB. The table space `PSAPTEMP` can have a maximum size of 2 TB.

Process Flow

1. You monitor storage and, if necessary, reorganize rather than extend.

   By monitoring your tables and reorganizing them when necessary, you can often avoid extending table spaces.

2. When free space in a table space reaches a critical limit, you decide which of the following to do:

- Reorganize tables [Page 93] in a table space to reclaim unused space in the data pages
- Extend the table space [Page 94] by adding containers

Your decision depends on the usage characteristics of the table space.

It is better to reorganize tables rather than extend table spaces.

Reorganizing Tables (DB2 UDB)

Use

With DB2 UDB, you reorganize individual tables rather specific table spaces. However, DB2 UDB allows you to reorganize tables according to a specific index. This index provides favorable I/O characteristics in case of table scans that can not be performed directly out of the buffer pool (either because they are not yet in the buffer pool or because they are too large for the buffer pool).

Prerequisites

Think about the way the table is used and how it grows or shrinks over time before deciding how to proceed:

- Tables with fluctuating space requirements

  For example, the table space `PSAPBTABD` contains tables that have a fluctuating space requirement. The tables often grow until R/3 transactions delete the processed transaction data records, so significantly reducing the number of extents required for the table data. Since the unused pages of the table remain allocated to the table and can be re-used, the table can grow in future without problem. You can reorganize tables that you do not expect to grow in the near future to provide free space for other tables.

- Tables with multiple inserts, deletes and updates

  You might be able to identify tables that have had multiple inserts, deletes, and updates. By reorganizing, you can compress these tables and reclaim the space from partially used or unused pages for the free space map of the table space. This allows other tables to use the released space for expansion.

These measures can prevent unnecessary table space extension.

Reorganization is not generally suitable for table spaces that contain master data, for example, `PSAPSTABD`. With this type of table space, you can only gain a small number of reclaimed data pages by table reorganization. In the long term, you might have to extend the table space [Page 94].

Procedure

1. Starting with R/3 Release 4.0A, you can use the Computing Center Management System (CCMS) in the R/3 System to automate the identification of tables requiring reorganization and then reorganize such tables if required. CCMS calls the checking tool “DB2 runstats” to provide this functionality.

2. Use the DB2CC [Page 166] tool to reorganize tables. You can also use DB2CC to reorganize all tables of a table space (this is similar to the “reorganize tablespace” functionality of other database management systems).

3. The CCMS provides extended facilities for table reorganization based on the primary index to:

   – Reorganize all tables within a table space

   – Reorganize tables identified by the checking tool DB2 runstats. With this tool, you can select the tables to reorganize from the list of proposed tables.

     The estimated runtime of the table reorganization is displayed and a total runtime is calculated based on the table history. The calculated runtime of the table reorganizations might not be correct if the size of the tables or the performance of the database host machine has changed. If this is the case, you might encounter runtimes that are substantially longer or shorter.

Extending Table Spaces (DB2 UDB)

Use

With DB2 UDB, you extend table spaces by adding containers. If you want to add several containers, it is best to do this as a single operation, since each time you add a container the table space is “rebalanced.” When rebalancing occurs, DB2 UDB moves pages into the new containers until all containers have the same number of used pages. Rebalancing happens automatically during normal R/3 operation.

All information about added containers is stored in the transaction log and is reapplied during database recovery (rollforward). If you are rolling forward and the device/file of the container is not available, the rollforward fails. DB2 UDB offers the option of suppressing recreation of the containers. Refer to the most recent DB2 UDB documentation for information on this feature.

Procedure

Use your judgment for container allocation

There are no fixed rules for the number of containers per table space. If you expect a table space to grow substantially during R/3 System operation, use containers larger than those used in the standard installation.

- Your `PSAPBTABD` table space is 4 GB in size and is growing at the rate of 1 GB per year. You initially have 4 disks available for container placement. In this case, it makes sense to install the table space with one container per disk, each having a size of 1 GB. The table space can then be extended online as soon as space runs out. If you are worried about performance, put this new container on an additional disk to maximize I/O parallelism on this table space with high transaction rate.

- Your `PSAPSTABD` table space is 2 GB in size and is growing at a rate of 100 MB per month. Since the total rebalancing time increases with the number of containers, adding containers with 600 MB seems reasonable. With this rate of increase, you have to extend the table space twice a year.

Recovery with DB2 UDB

You choose one of the following types of recovery to restore the DB2 Universal Database (abbreviated to DB2 UDB):

- Crash recovery

  In the case of a system crash and a subsequent database restart, DB2 automatically tries to recover the database. For this, the `AUTORESTART` database parameter is enabled by default in your R/3 database.

  The database always tries to create a log file during crash recovery. If there is not enough log space, the recovery fails.

- Restore recovery

  This type of recovery enables the restore of a previous version of the database that was made with the DB2 backup command.

- Rollforward recovery

  A rollforward recovery follows the restore of the database with the application of the transaction logs created since the last DB2 backup.

  You can perform rollforward recovery if the database has been configured for `LOGRETAIN=ON`. This setting is mandatory for productive R/3 Systems.

Upgrade with DB2 UDB

Purpose

You use the following to upgrade the database software for DB2 Universal Database (abbreviated to DB2 UDB):

- On Windows NT systems, SETUP for DB2 UDB
- On AIX systems, DB2SETUP (or the SMIT tool)

It only takes a few minutes to upgrade the database software and database instances.

Prerequisites

Before starting the upgrade you:

- Stop your R/3 System and the database manager.
- See the release notes for the new database software.

Process Flow

1. To upgrade the database software, you only use the program temporary fixes (PTFs) from SAP. Whenever you receive a new database software level, you check with SAP whether this level has been tested and approved by SAP.
2. You perform the upgrade according to the documentation supplied by SAP.

High Availability for the DB2 for OS/390 Database

Purpose

This section looks at database administration for the DB2 for OS/390 database and make specific recommendations on improving availability.

Process Flow

21. You work out your approach to backing up your database. You must do this before you start production with the database. When using the database for production, you must back up the database as regularly as possible.

    Refer to Backup with DB2 for OS/390 [Page 97].

22. You manage the space in your database. You do this both before you start production with the database and during production as the database grows.

    Refer to Space Management with DB2 for OS/390 [Page 100].

23. You upgrade your database when required.

    Refer to Upgrade with DB2 for OS/390 [Page 105].

24. You recover your database if a failure occurs with data loss.

    Refer to Recovery with DB2 for OS/390 [Page 103].

25. You consider using various tools and services offered as standard by SAP to increase the availability of your database:

    – Computing Center Management System (CCMS) [Page 166]

    – GoingLive and EarlyWatch [Page 169]

10. You consider using advanced products and services to increase the availability of your database:

    – DB Reconnect [Page 174]

    – Data Sharing with DB2 for OS/390 [Page 210]

    – Replicated Standby Database for DB2 for OS/390 [Page 201]

    – Switchover Software [Page 216]

11. You review your procedures to increase the availability of your database. Refer to High Availability Procedures at Your Site [Page 247].

Result

Your DB2 for OS/390 database is more available for production use.

See also:

SAP R/3 Database Administration Guide: DB2 for OS/390

Backup with DB2 for OS/390

Purpose

This section describes how to reduce downtime when backing up your R/3 database in a DB2 for OS/390 environment. When the database is correctly backed up, you can avoid excessive downtime if you have to restore your database following database failure.

With DB2 for OS/390, you back up or recover tablespaces, not individual files or datasets. This allows DB2 to control consistency of the datasets belonging to one tablespace more efficiently. Also, DB2 records all backups in the catalog, so that it can automatically decide whether and which parts of the recovery log are needed.

Prerequisites

You need to set up backup procedures for each individual R/3 database. Be sure to consider:

- System availability requirements
- Rate of change of the database
- Database size
- Hardware and software resources

The appropriate backup procedure is a key factor for data recovery. It is generally better to have a recent tablespace backup available, so that a large amount of log data does not have to be restored during a recovery.

The main characteristics of a backup procedure are:

- Its frequency, for example, how often a tablespace is backed up.
- The tools used to produce backups.

The main types of backups are:

- Online Backup

  An online backup copies tablespace data while the tablespace remains online (that is, concurrent reads or writes are possible). Therefore, except for a small processing and disk access overhead, the online backup has no impact on the R/3 System. As it can contain uncommitted data, such a backup alone is never enough for tablespace recovery. Therefore, DB2 for OS/390 automatically applies the recovery log on top of the online backup, as needed.

  You can also create an "incremental" online backup, which only backs up data pages that have changed since the last backup.

  The DB2 `COPY` utility with the `SHRLEVEL(CHANGE)` option is an efficient tool for creating online backups.

- Offline Backup

An offline backup copies tablespace data while the tablespace is offline (that is, concurrent write activity on the tablespace is not possible). Therefore, all the data is committed, which means that this backup alone could be used for recovery, but only to the point in time at which the offline backup was taken.

Note that offline does not refer to either the DB2 subsystem or the R/3 System. That is, concurrent reads or writes can continue on all other tablespaces, as well as read only on the tablespace being backed up. However, it is often unacceptable to create offline backups of R/3 data (especially of frequently updated tablespaces) during normal operations. This is particularly true if the offline backup takes too long.

You can also create an "incremental" offline backup, which only backs up data pages that have changed since the last backup.

The DB2 `COPY` utility with the `SHRLEVEL(REFERENCE)` option is an efficient tool for creating offline backups. Also consider using the `FULL`, `CHANGELIMIT`, `COPYDDN`, and `RECOVERYDDN` options. However, the best option for this purpose is `CONCURRENT`, which can significantly reduce the time during which the tablespace is unavailable for write activity.

An offline backup of all R/3 tablespaces, DB2 catalog, and DB2 directory is a very restrictive way of backing up the data and should be used only where it is possible to stop the R/3 System for the required period. However, offline backups are very useful for recovery to a prior point-in-time.

DB2 and OS/390 offer a number of ways to implement such a backup.

Here is an example of how to create an offline backup:

a. Put the R/3 database into quiescent mode (that is, make sure that no users can perform updates).

b. Run the DB2 `COPY` utility.

To speed up the process, you can create several `COPY` jobs that run in parallel. The elapsed time can be improved significantly if you group the tablespaces so as to reduce path contention between direct-access storage devices (DASDs).

Instead of using the `COPY` utility, which operates on a tablespace level, you can make copies of all the volumes on which the R/3 data reside by using the RAMAC Virtual Array SnapShot feature (where available). It is the fastest method to copy the data, but makes the recovery more complex. For more information, see the IBM documentation *Implementing SnapShot*.

c. Restart the DB2 subsystem and databases to allow normal access.

Process Flow

1. You plan your backups as follows:

   – You get to know DB2 backup and recovery processes.

   – You assess the factors that influence the characteristics of the backup and recovery procedures.

- You develop procedures for all backup and recovery situations.
- You practice these procedures when normal operations are not affected.
- You dedicate a DB2 subsystem or DB2 data sharing group to the database for the R/3 System, because this simplifies point-in-time recovery.

2. You run your backups as follows:
   - You use dual logging for the active log, archive log, and bootstrap datasets.
   - You place the copies of the active log datasets and bootstrap datasets on different direct access storage device (DASD) volumes.
   - You do not discard archive logs from after the most recent consistent copy of any R/3 tablespace (or from after an older consistent copy if you require a point-in-time recovery to a prior point in time).
   - You consider producing multiple backup copies.
   - You retain level backups to extend the interval when a prior point-in-time recovery is possible as well as to avoid the impact of possibly damaged (that is, inconsistent) backup datasets.
   - You avoid backing up tablespaces that contain inconsistent data (that is, within a single page). Use the `DSN1COPY`, `DSN1CHKR`, and `CHECK INDEX` tools to detect such inconsistencies in the users and catalog or directory tablespaces.
   - You back up the DB2 catalog and directory, especially after the activities that involve a lot of data definition language (DDL), such as `R3load`, transport, or upgrade of the R/3 System.
   - To speed up recovery, you use more and larger active logs, consider archiving to disk, and make sure you have enough tape drives. Also, make sure that the sizes of buffer pools and log buffers are as recommended for the R/3 System.
3. After a successful R/3 installation, upgrade, system copy, or recovery to a prior point in time, you create an offline backup of the R/3 database. This is mandatory.
4. If your operations schedule allows, you create offline backups of the R/3 database occasionally. If there is not enough time available to back up the entire database, you create offline backups of the heavily updated and critical tablespaces. We strongly recommend this.

   Offline backups mean that a prior point-in-time recovery is very efficient, especially if the backup is created by a tool that copies both tablespaces and indexspaces (such as RVA SnapShot or DFSMSdss).
5. You regularly create online backups for any R/3 tablespaces, DB2 catalog tablespaces, or DB2 directory tablespaces used in the R/3 System. How often the backup needs to be created and whether it is full or incremental, depends on the rate of data change.

   Initially, we recommend that you run backup jobs every one or two days and specify `CHANGELIMIT(10)`. After you have categorized your tablespaces according to update intensity (that is, heavily, moderately, or lightly updated), you can change the backup frequency accordingly (that is, to daily, weekly, or monthly). You can use transaction ST10 to categorize tables by their access pattern.

Using the `CHANGELIMIT` option might result in infrequent full backups, which is not efficient for recovery. Therefore, make sure you have a full backup created periodically by specifying `FULL(YES)` or `CHANGELIMIT(0)`. Also, consider running the `MERGECOPY` utility that consolidates a full and several incremental backups into a new, more recent full backup.

See also:

SAP R/3 Database Administration Guide: DB2 for OS/390

High Availability Considerations : SAP R/3 on DB2 for OS/390 (from IBM)

DB2 for OS/390 Administration Guide (from IBM)

Implementing Snapshot (from IBM)

Space Management with DB2 for OS/390

Purpose

Space management might not at first seem to be an area of relevance for high availability. However, if neglected, it can lead to downtime. For example, if a database object needs to expand, but is not able to do so, applications cannot continue writing to the database and you quickly have to make more space available.

You can manage space for R/3 with DB2 for OS/390 using either of the following:

- DB2 itself, known as "DB2-managed data"
- Data Facility Storage Management Subsystem (DFSMS), which is system-managed storage, known as "SMS-managed data"

SAP does **not** support the DB2 method "user-managed data" for the R/3 System.

Storage Management Subsystem (SMS) is the IBM automated approach to managing auxiliary storage such as disk space. It uses software programs to manage data security, placement, migration, backup, recall, recovery, and deletion. Using these functions, SMS makes sure that current data is available when needed and obsolete data is removed from storage.

In this section, both SMS-managed and DB2-managed data are described. Both options provide a high degree of automation. We recommend you to use SMS-managed data if possible.

With DB2 for OS/390 the data to be managed consists of:

- R/3 and DB2 system data (tablespaces and indexspaces)
- DB2 bootstrap dataset (BSDS)
- Catalog and directory data
- Image copies (that is, database backups)
- Archive logs

All of these can be managed by SMS. However, with both SMS-managed and DB2-managed data it is still possible to run out of space:

- With SMS-managed data, an SMS storage group can fill up.
- With DB2-managed data, a DB2 stogroup (that is, a storage group) can fill up.

In both cases, a dataset can reach its maximum number of extents, when you must provide more space as described below.

Process Flow

1. If all the volumes in a DB2 stogroup (DB2-managed data only) are full:

   a. You add additional volumes by executing the following SQL statement:

   ```
   ALTER STOGROUP <FULL_STOGROUP> ADD VOLUMES (<VOLID_NEW1>,
   <VOLID_NEW2>, ...)
   ```

   b. You prevent this happening again by using transaction DB02 to check each DB2 stogroup for sufficient free space.

   A DB2 stogroup is a collection of one or more volumes (that is, disks). If all the volumes in a stogroup are full, any transaction that writes data to tables or indexes associated with that stogroup fails.

2. If an SMS storage group (SMS-managed data only) is full:

   a. You add additional volume(s) to the full storage group by using the DFSMS application or the appropriate DFSMS commands.

   b. You prevent this happening again by checking each SMS storage group for sufficient free space. Depending on the granularity of your SMS storage groups, you can significantly reduce the number of objects to check for available free space, compared to DB2-managed data.

   An SMS storage group is a collection of one or more volumes. If all the volumes in an SMS storage group are full, transactions fail when attempting to write data to tables or indexes in that storage group.

3. If the maximum number of dataset extents (both SMS and DB2-managed data) is reached:

   a. You reorganize the tablespace as follows:

      i. You increase the `PRIQTY` and `SECQTY` parameters of the tablespace using `ALTER TABLESPACE` (or `ALTER INDEX`). Set these parameters so that less than 20 extents are used.

      ii. You use `REORG TABLESPACE` (or `REORG INDEX`) for the relevant tablespace. DB2 for OS/390 supports online reorganization (that is, the R/3 System is running during this action).

   b. You prevent this happening again by regularly checking that the number of tablespace and indexspace extents remains well below the maximum of 255. You can do this using either of the following monitors in the Computing Center Management System (CCMS):

      - Database monitor [Ext.]
      - Alert Monitor [Ext.]

A DB2 tablespace consists of one or more datasets used to store DB2 tables. When a tablespace is created, DB2 allocates disk space as defined by the `PRIQTY` (primary quantity) parameter of the tablespace. Each time more space is needed, DB2 allocates additional disk space as defined by the `SECQTY` (secondary quantity) parameter. The same applies for indexspaces. The maximum number of extents is 255 (but prior to OS/390 version 2 release 4 the maximum number of extents was 119).

If the maximum dataset size of 2 GB is reached, DB2 automatically creates an additional dataset for this tablespace, again starting with `PRIQTY` followed by `SECQTY` as needed. This is repeated up to the maximum number of datasets per tablespace, which is 32 for segmented tablespaces and for each partition of a partitioned large tablespace (which can have up to 254 partitions).

4. You monitor your tablespaces and indexes carefully with a suitable monitor, such as those mentioned above. Check for scattering of rows in a tablespace or indexspace caused by updates. Scattering causes performance degradation.

5. If you need to reorganize the tablespace or index, you choose offline or online reorganization:

   - Offline reorganization

     The DB2 offline reorganization command unloads the data from the tablespace, sorts the data in clustering sequence, and then reloads the data with the desired free space. The indexes are synchronized with the tablespace data. During the unload phase, R/3 has read-only access to the data in the tablespace. During the other phases, R/3 has no access to the data. Due to the integrated nature of the R/3 applications this means that the R/3 System needs to be stopped during an offline reorganization.

   - Online reorganization

     The DB2 online reorganization allows R/3 read-write access to the data during most of the time it is running. The online reorganization makes a "shadow copy" of the tablespace and reorganizes this copy. After the reorganization of the shadow copy is complete, DB2 reproduces changes from the original tablespace to the shadow copy using the DB2 recovery log. During this action, R/3 has full read-write access to the data. During the very last updates to the shadow copy, DB2 only allows read-only data access.

     After this, DB2 switches from the original dataset to the shadow copy. During the switch phase, R/3 has no access to the tablespace. If you need to reorganize R/3 tablespaces or indexes, use online reorganization because this increases the availability of R/3 data.

6. If required, you separate tables.

   Most DB2 utilities, such as `REORG`, operate with tablespaces rather than tables. DB2 monitoring is better supported for tablespaces than for tables. Therefore, if a tablespace in a multi-table tablespace becomes very large, grows rapidly in size, has to be reorganized frequently, or is accessed very often, you can move it to a single-table tablespace. Moving tables from a multi-table tablespace to a single-table tablespace involves moving the table data. Depending on the table size, this can cause significant downtime.

Since Release 4.5A, all R/3 tables that are not R/3-buffered are in single-table tablespaces by default. Therefore, you very rarely need to separate tables.

See also:

SAP R/3 Database Administration Guide: DB2 for OS/390

Recovery with DB2 for OS/390

Purpose

To minimize downtime in the event of failure, you must make sure that you can quickly restore database data. Some recovery operations are done automatically by DB2 for OS/390 without any outside intervention, such as recovering the database to a consistent state before an operating system or database failure. In this case, automatic recovery happens at the next DB2 for OS/390 start.

DB2 provides the `RECOVER` utility for data recovery. This lets you recover DB2 objects such as tablespaces, indexes, partitions, individual datasets, and individual pages. With the `RECOVER` utility you can recover data to:

- The state captured in a particular backup (the `TOCOPY` option),
- The state at the time corresponding to a relative byte address (the `TORBA` option) or a log record sequence number (the `TOLOGPOINT` option). The `TORBA` option is used in non-data sharing and the `TOLOGPOINT` option in data sharing environments.
- The current state by not specifying any of the above options.

The `RECOVER` utility also has the `LOGONLY` option, which allows you to recover the data using the log only, starting with a backup that was created outside DB2 (for example, RVA SnapShot).

There are the following types of recovery:

- Recovery to the current state

  A recovery to the current state is generally less demanding and is usually needed more often than a point-in-time recovery. A typical example is volume failure in a direct access storage device (DASD), resulting in data loss. You need to find out which tablespace and indexes resided on the volume and recover only these tablespace and indexes, or even only partitions or individual datasets that are affected. The rest of the system is already at the current state and need not be recovered.

- Recovery to a prior point in time

  This type of recovery is used to reinstate the R/3 database to the condition it was in at a prior point in time. All changes after that time are lost. You must carefully consider the decision to set the system back in time. Typically, a recovery to a prior point in time is needed when an application program logic error introduced unwanted and irreversible changes into the system.

  This type of recovery is explained in "Process Flow" below.

You can speed up any recovery by splitting the job into multiple parallel recovery streams to reduce disk contention, but note the restrictions documented in the IBM documentation *DB2 for*

OS/390: Tool Guide and Reference. Note that the `REUSE` option of the `RECOVER` and `REBUILD` utilities significantly reduces the overall recovery elapsed time.

Process Flow

1. You consider the following when choosing a method for a point-in-time recovery:
   - How soon the data must be available again
   - Which point in time you want to use for the recovery
   - Whether offline backups are available
   - Whether indexspaces were included in an offline backup
   - Whether quiesce points are available
2. You choose one of the following methods for recovery to a prior point in time:
   - Recovery with conditional restart

     This method causes the least interruption to production operation. It requires neither offline backups nor quiesce points and this makes it the best choice in high availability environments. It can also bring the system closest to the time when the database is known to be consistent, so reducing unnecessary data loss.

     For more information, see the SAP documentation *SAP R/3 Database Administration Guide: DB2 for OS/390*.
   - Recovery to a consistent offline backup

     This is the simplest and fastest method, but it is also the most restrictive, for the following reasons:

     - It requires creating offline database backups, which means planned system downtime that some R/3 installations cannot tolerate. Note that all backups used in a recovery have to be created during the same downtime.
     - It can set the system further back than necessary, depending on the backup frequency. For example, assume that offline backups of the R/3 database are scheduled weekly on Sundays. If the data was damaged on Friday, all the changes made from the previous Sunday to Friday would be lost when Sunday's offline backup was used for the recovery.
   - Recovery to system quiesce point

     This depends on the existence of "system quiesce points". These are the points in time when there are no uncommitted update transactions in the system. Such a point is specified by the corresponding relative byte address (RBA) or log record sequence number (LRSN).

     Recovery to a system quiesce point is better than to a consistent offline backup, because establishing a quiesce point is less disruptive to the R/3 System than creating an offline backup. Therefore, it is more likely that recovery is possible to a point that is closer to the required time, so reducing data loss. However, establishing frequent quiesce points can significantly reduce performance.

Upgrade with DB2 for OS/390

Purpose

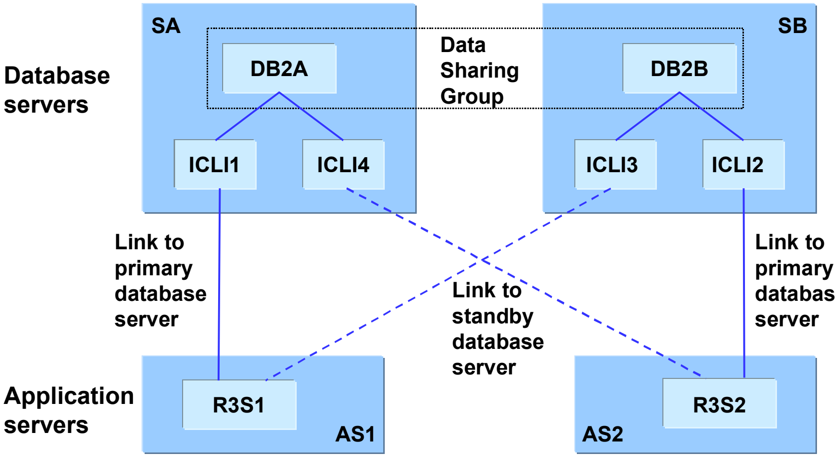
You normally upgrade the database before the R/3 System. In a DB2 data sharing environment you can upgrade the DB2 version with minimal downtime for R/3.

The first step in a DB2 release upgrade is to run a DB2 catalog maintenance utility to upgrade the DB2 catalog structure to the new release level. This allows different releases of DB2 (for example, DB2 V5 and DB2 V6) to share the catalog in toleration mode. This step can take up to 30 minutes and the R/3 System is down during this period.

You can then start the R/3 System again and upgrade each member of the data sharing group separately without downtime for the R/3 System. The R/3 application server that is connected to this member can switch over to another member of the data sharing group. See also Data Sharing for DB2 for OS/390 [Page 210].

Prerequisites

This procedure is based on the start configuration shown below:



There is DB2 data sharing group with two members, DB2A and DB2B. Both members are initially on DB2 version 5 and are to be upgraded to DB2 version 6 without R/3 downtime. On each database server there are two integrated call level interface (ICLI) server instances for primary and standby connections.

Process Flow

1. You stop ICLI server instance ICLI1.

   The application server R3S1 connected to DB2A switches over to DB2B and connects to the ICLI server instance ICLI3. Wait until all work processes of application server R3S1 have switched over.

2. You stop ICLI server instance ICLI4.

   Nothing happens since no application server is connected to ICLI4.

3. You stop DB2A.
4. You upgrade DB2A to version 6.
5. When the upgrade is completed, you start DB2A and the two ICLI server instances ICLI1 and ICLI4.
6. You stop the ICLI server instance ICLI3.

   The application server R3S1 switches back to DB2A and connects to the ICLI server ICLI1. Wait until all work processes of application server R3S1 have switched back.
7. If you intend to upgrade DB2B also at this time, you skip this step. Otherwise, you start the ICLI server instance ICLI3 to prepare for possible failover of R3S1.

   Both members are now running. DB2A is running version 6 and DB2B is running version 5.
8. Once you decide to also upgrade DB2B, you stop ICLI server instance ICLI3 and continue with the next step.
9. You stop ICLI server instance ICLI2

   The application server R3S2 connected to DB2B switches over to DB2A and connects to the ICLI server instance ICLI4. Wait until all work processes of application server R3S2 have switched over.
10. You stop DB2B.
11. You upgrade DB2B to version 6.
12. When the upgrade is completed, you start DB2B and the two ICLI server instances ICLI2 and ICLI3.
13. You stop the ICLI server instance ICLI4.

    The application server R3S2 switches back to DB2B and connects to the ICLI server instance ICLI2. Wait until all work processes of application server R3S2 have switched back.
14. You start the ICLI server instance ICLI4.

    The R/3 System is fully operational throughout the above procedure. However, the performance of the database server is temporarily affected at times. This procedure can also be used for the upgrade of the OS/390 operating system.

See also:

SAP R/3 Database Administration Guide: DB2 for OS/390

High Availability for the DB2/400 Database

Purpose

This section looks at database administration for the DB2/400 database and makes specific recommendations on improving availability.

Process Flow

1. You work out your approach to backing up your database. You must do this before you start production with the database. When using the database for production, you must back up the database as regularly as possible.

   Refer to Backup with DB2/400 [Page 107].

2. You recovery your database if a failure occurs with data loss. Refer to Recovery with DB2/400 [Page 107].

3. You consider using various tools and services offered as standard by SAP to increase the availability of your database:

   – Computing Center Management System (CCMS) [Page 166]

   – GoingLive and EarlyWatch [Page 169]

3. You review your procedures to increase the availability of your database. Refer to High Availability Procedures at Your Site [Page 247].

Result

Your DB2/400 database is more available for production use.

See also:

BC R/3 Database Guide: DB2/400 [Ext.]

Backup with DB2/400

Purpose

You can automate the daily backup for the DB2 database running on the AS/400 platform using the operating system scheduler.

Process Flow

You use the command `SAVR3SYS`, which saves all the information required for an R/3 System.

See also:

Backup and Recovery [Ext.]

Recovery with DB2/400

Purpose

To minimize downtime in the event of failure, you must make sure that you can recover your DB2/400 database.

Prerequisites

You can recover the DB2/400 database with operating system tools but data definition language (DDL) operations – such as `create table`, `alter table` and so on – are **not** automatically recovered. Therefore, SAP recommends you to perform upgrades in mode *A_off*.

Process Flow

You use operating system tools to perform the recovery for DB2/400.

See also:

Backup and Recovery [Ext.]

High Availability for the MS SQL Server Database

Purpose

This section looks at database administration for the Microsoft (MS) SQL Server database and makes specific recommendations on improving availability.

Process Flow

1. You read the SAP database administration documentation for MS SQL Server [Ext.].
2. You consider using various tools and services offered as standard by SAP to increase the availability of your database:
   – Computing Center Management System (CCMS) [Page 166]
   – GoingLive and EarlyWatch [Page 169]
3. You consider using advanced products and services to increase the availability of your database:
   – Microsoft Cluster Server (MSCS) on Windows NT [Page 220]
   – Microsoft SQL Server Standby Database [Page 206]
   – Comprehensive Microsoft SQL Server High Availability Solution [Page 214]

Result

Your MS SQL Server database is more available for production use.

Network System Key Issues

Purpose

This section discusses network availability, its impact on the R/3 system, and how to maximize network resilience for greater R/3 System availability. For most SAP customers, network availability is not unique to the R/3 System, since it is required by any other systems that operate in the network environment.

Implementation Considerations

For detailed technical guidance when implementing a specific product or feature, contact the appropriate source, such as your SAP consultant or your network supplier.

Integration

High availability for the R/3 System should be part of a system-wide strategy. Therefore, you should also consider other components of the system, such as the R/3 System itself, the database management system (DBMS), the hardware and operating system services, and so on.

Networks

Definition

This section describes the basic components of a network supporting the R/3 System and their general relevance for improving system availability. You can think about networks in the following ways:

- Architecture

  This approach looks at the way servers and clients are brought together. For example, how is the database host linked to the application hosts, or the frontends to the application hosts? See "Structure" below.

- Communication layers (using the OSI networking model)

  This approach breaks the various aspects of a computing network into distinct conceptual layers, from the hardware layer at the bottom up to the application layer at the top. This section looks at the lower layers of this model. See "Integration" below.

Structure

A typical R/3 System environment contains components comprising the following architectural layers:

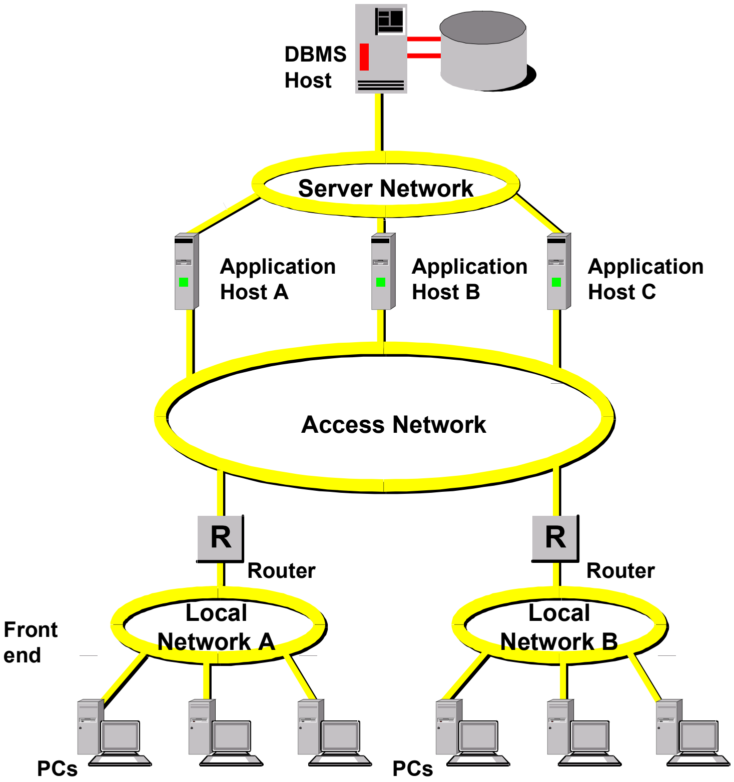
- Database services
- Application services
- Frontend clients (that is, SAPGUI)

These components usually run on different host machines in a network environment and they communicate using various network protocols. The flow of data and requests follows the three-tier architecture where the clients talk only to the application hosts which in turn talk to the database hosts. A database host can serve one or more application hosts and each application host can serve one or more frontend clients.

The diagram below shows the network for a typical R/3 installation in which a single database host, a number of application hosts and a number of frontends (that is, the SAPGUI platform, usually PCs) are connected together.

R/3 Architecture: Network View

Networks



This shows that the network can be broken down as follows:

- Server network

  The network connecting the application hosts to the database host is critical to system availability. If the database host can no longer communicate with other system components, the R/3 System cannot continue operating. Therefore, special measures to increase network resilience are warranted here.

  For more information, see Server Network [Page 115].

- Access network

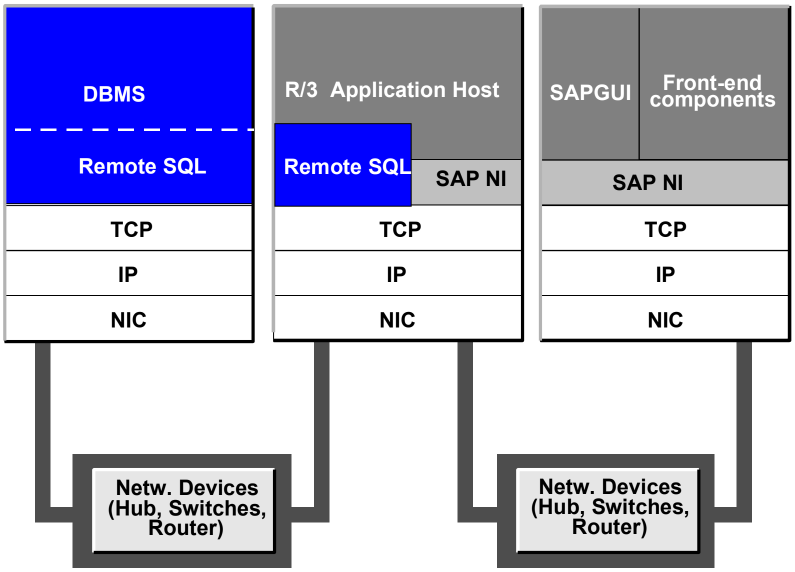
  The network connecting the application hosts to the frontends is often the main “company network” or the “company backbone”, and generally has some built-in redundancy. You should make sure that the connection of the application hosts to the company network has some redundancy, that is, these hosts should be connected across at least two paths to avoid the situation of a single point of failure.

  For more information, see Access Network [Page 120].

Integration

You can also think about networks by looking at the communication level. The diagram below illustrates the communication layers in an R/3 System network:

R/3 Communication Layers



SAP NI refers to SAP Network Interface layer.

It is important to realize that you should take measures at all of the relevant communication layers to improve availability. There is no single “availability layer” at which you can easily take all the necessary measures. The approach taken is also different for the server and the access network (see “Structure” above).

The upper layers (above the transport layer) are not discussed in this documentation since they are application-specific. Communication between the application and database hosts, for example, is performed by the remote SQL function of the database. For more information, contact your database vendors.

The R/3 database interface is located above the remote SQL interface and for this a reconnect function was implemented with R/3 release 2.2F and 3.0A. Refer to DB Reconnect [Page 174].

The communication layers are discussed further in the following sections:

- Physical Layer [Page 112]
- Network and Transport Layers [Page 114]

See also:

Integration of R/3 Servers in TCP/IP networks (in SAPNet)

SAP Software in PC Networks (in SAPNet)

Physical Layer

Definition

This section discusses the network hardware in some detail. It is always best to fix a problem on the lowest possible level of the network since this leads to a shorter system downtime and greater transparency for the R/3 applications.

Structure

The physical layer can be divided into the following main parts:

- Network infrastructure

  This includes cabling as well as active components such as hubs, switches, routers (these are not just pure hardware but have built-in intelligence).

- Network Interface Card (NIC)

  Every server or client machine in the network has at least one of these cards through which all network traffic must pass.

From the high availability viewpoint, the resilience of the network depends greatly on the technology used (Ethernet, Token Ring, FDDI) and the topology (bus, ring, meshed and the degree of redundancy built-in).

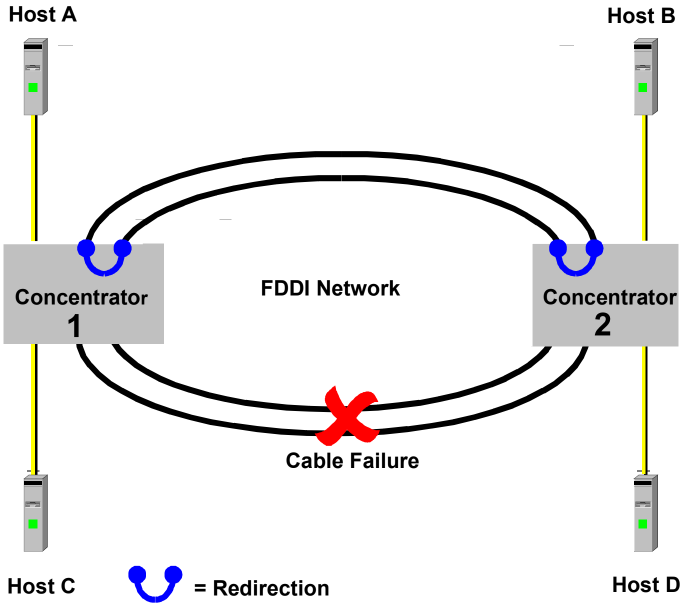
Network Infrastructure

This section discusses a reliable method of protecting your infrastructure, FDDI, and the building of redundancy into the network topology.

FDDI

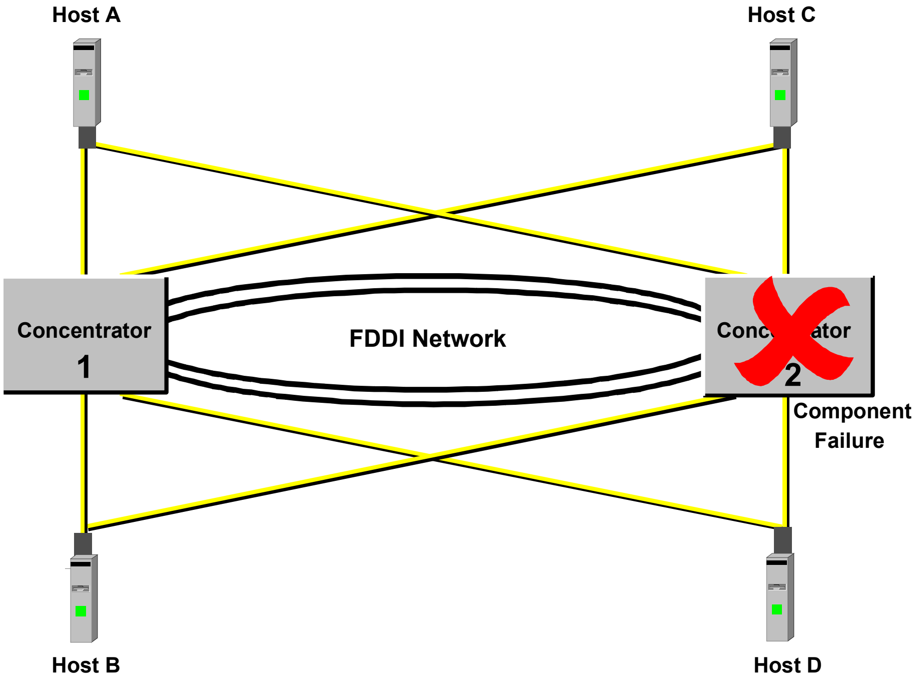
Redundancy is an integral part of FDDI, making it a good solution if you want to increase the availability of your network. It protects your network against the common causes of failure in the physical infrastructure. Error correction with FDDI occurs very rapidly and transparently to the higher layers of the network (IP, TCP). The following diagrams show how FDDI works in the event of failure:

FDDI: Broken cable



The dual-ring system shown in the above diagram protects against cable failure. Only one ring is required for the data flow. In the case of cable failure (cable broken, damaged or plug is out of socket), the adjacent active FDDI components detect this and bridge the two FDDI rings to restore an effective functional ring.

FDDI: Component failure



The dual-homing concept shown in the above diagram protects against component failure. Client or server computers with dual-attachment have access to both of the FDDI rings (one ring using the first concentrator and the second ring using the second concentrator). If a concentrator fails, the data flow is automatically redirected through the remaining device.

Redundancy in Network Topology

This method of achieving higher network availability by providing additional physical networks and additional paths can be used with all network technologies. The bridges (at OSI layer 2) or routers (at OSI layer 3) handle redundancy. Switchover to an alternate path in the event of failure takes longer than a corresponding FDDI switch (see above) and is often not transparent to the higher network layers. The TCP/IP session often fails and has to be re-established.

SAP recommends that you build redundancy into your network (that is, into the access network, enabling frontends to access the application hosts). Even if the switch takes some time, it is a means to bypass a damaged network path without operator interference and can be combined with application-level reconnects.

Network Interface Card (NIC) and Switchover Software

Even if the rest of the network is protected against failure using FDDI technology or redundancy in the network topology, the network interface card (NIC) remains a single point of failure in the physical part of the network connection. The NIC can in fact be considered as part of the hardware of the individual computer that it serves. If the NIC fails, one solution is simply to switch to a backup machine. If you want to eliminate the NIC as a single point of failure, a second (standby) NIC can be installed. Switching to a second NIC in the event of failure is faster than switching to a backup machine.

The handling of multiple NICs on a single computer is not easy. Special software products can be used to perform the takeover, by which the standby NIC takes over the address of the primary NIC in order for connectivity to continue uninterrupted.

The best solution is undoubtedly the use of proprietary switchover. In this scenario, the standby NIC is connected to the same physical subnet as the primary one. The switchover software monitors its functionality by sending “heartbeat” messages. If a primary NIC fails, the standby NIC is automatically activated with the address of the primary NIC and the connection resumes. The switchover occurs very rapidly indeed so that applications with TCP/IP connections across the damaged NIC might not even notice it. The degree of resilience depends on the switchover software product used (that is, whether the TCP/IP connection is maintained or breaks depends on the switchover software).

See also:

Switchover Software [Page 216]

Network and Transport Layers

Definition

The central concept here is TCP/IP (Transmission Control Protocol/Internet Protocol). Closest to the physical layer is IP, controlling the network layer and higher still is TCP, controlling the transport layer.

Structure

The IP and TCP layers are built up as follows:

- Network Layer – IP

  The network layer controlled by IP is the next level up from the link layer and is responsible for finding routes through the network and sending information to the next hop in the route. Knowledge of the topology and a list of “best routes” through the network for individual computers are stored in a routing table located on each network node. The system administrator usually defines this information statically in general-purpose computers.

  One method to achieve higher availability in this layer is to implement special routing protocols that can be used to interchange information about possible routes between the network nodes. This allows the build-up of routing information dynamically, also enabling alteration of it if the structure of the network changes. In this way, network routers can pro-actively maintain information about their network.

  Most general-purpose computers today do not have routing protocols available (they are held in routers). This means that the routing tables of the computers are configured statically by the system administrator. The administrator in this approach determines how each network adapter directs traffic through the network.

- Transport Layer – TCP

  TCP is the transport protocol that maintains end to end connection between communicating partners (that is, computers).

  TCP uses a sophisticated retransmission and error detection scheme. However, it can take TCP a considerable time (several minutes) to react to loss of the underlying network because it is designed to cope with transmission delays in WANs. TCP does not know about the underlying network structure, alternate routes etc.

Example

An example of a typical failure scenario is as follows. When the IP layer receives a data package to be sent, it searches the internal routing table and then takes the first route matching the desired criteria. If this route uses a failed network adapter, the TCP layer (see above) is either directly informed (if the adapter driver reports an error) or finds out by making numerous failed retransmissions (which can take up to 9 minutes). The IP layer itself in most implementations does not attempt to use alternate routes, even if present. In some cases (for example, NT), the IP layer searches for an alternate route **after** the TCP session has failed. Normally, the routing table has to be deliberately changed to specify an alternate route (manually or by a script defined for this purpose).

Note that, as stressed earlier, correction of faults in the higher connectivity layers (that is, network and transport) is invariably much slower than correction at the physical layer. The duality inherent in FDDI (dual-ring and dual-homing) makes it clearly the preferred solution for maintaining very high network availability.

Server Network

Definition

The server network is the network between the database host and the application hosts.

Use

Whether or not you use a separate server network and, if so, how it is configured depends on your high availability requirements as well as the size of the network:

- Small installations – no server network

  Small installations often use a central system, in which the database management system (DBMS) is located on the same machine as the application service, and therefore no server network is required.

- Medium and large installations – server network

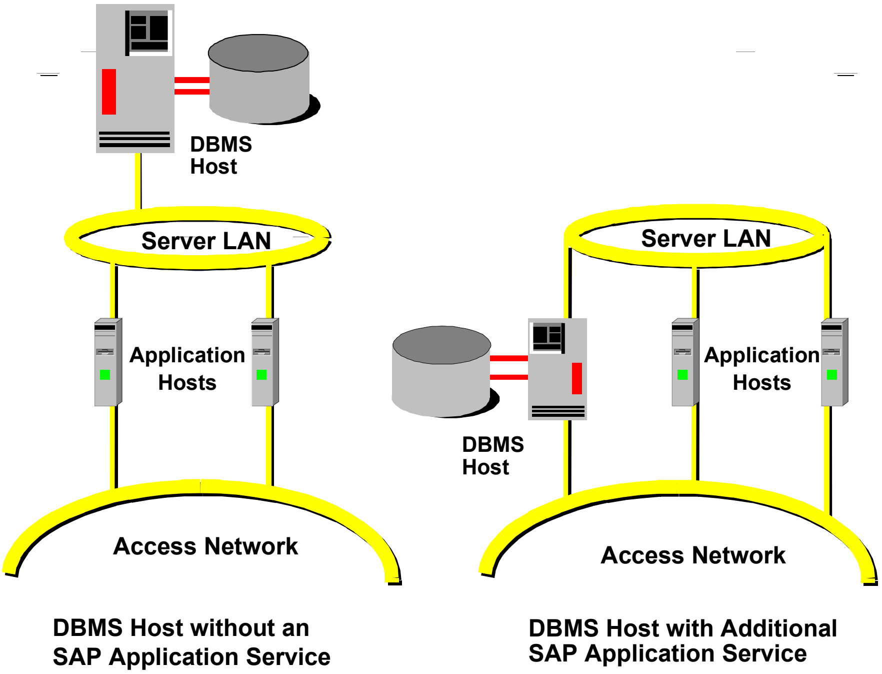
  With larger installations, in which several distinct machines are used to provide R/3 database and application services, a fast network is used for communications between the various hosts. Whereas failure can often be tolerated if restricted to other parts of the overall network, if the server network fails, the R/3 System as a whole can no longer provide proper service. Therefore, reliability of the server network is extremely important for high availability, as well as for general system performance.

  For performance, administrative and security reasons, it is best to use a separate local area network (LAN) segment (that is, a server LAN) for the server network. This enables you to more efficiently administer and configure the server network for higher availability if required.

Structure

Distinguish the server network from the access network [Page 120], which links the frontends together. A typical server LAN looks as follows:

Server LAN



The network load between the application and database hosts is typically ten times higher than that between the application host and front ends. The actual load depends on the applications used, user profiles, customer data and general system tuning. Experience shows that low speed networks such as Ethernet operating at 10 M bits per second (around 1 MB per second) are sufficient only in small installations. A network technology offering a higher bandwidth, such as FDDI, is preferable in most cases for the server network.

💡

Use a separate LAN segment for the server LAN.

For performance, administrative and security reasons, it is best to use a separate LAN segment for the server network. This enables you to more efficiently administer and configure the server network for higher availability.

💡

Keep hosts close together

Connect hosts directly to the server LAN rather than using decentralized application hosts.

💡

Use high-speed LAN with built-in redundancy

The investment in leading LAN technology such as FDDI to provide redundancy provides improved availability for your system.

💡

Structured cabling

Use structured cabling with standardized high-quality cables to build your network.

💡

Facilitate management of the network

To simplify network administration, you should equip all active components so that they can be monitored and managed easily.

💡

Take additional measures to improve network availability

Measures such as Uninterruptible Power Supply (UPS) should be taken to protect active components. Switchover software should be used to monitor the Network Interface Card (NIC).

💡

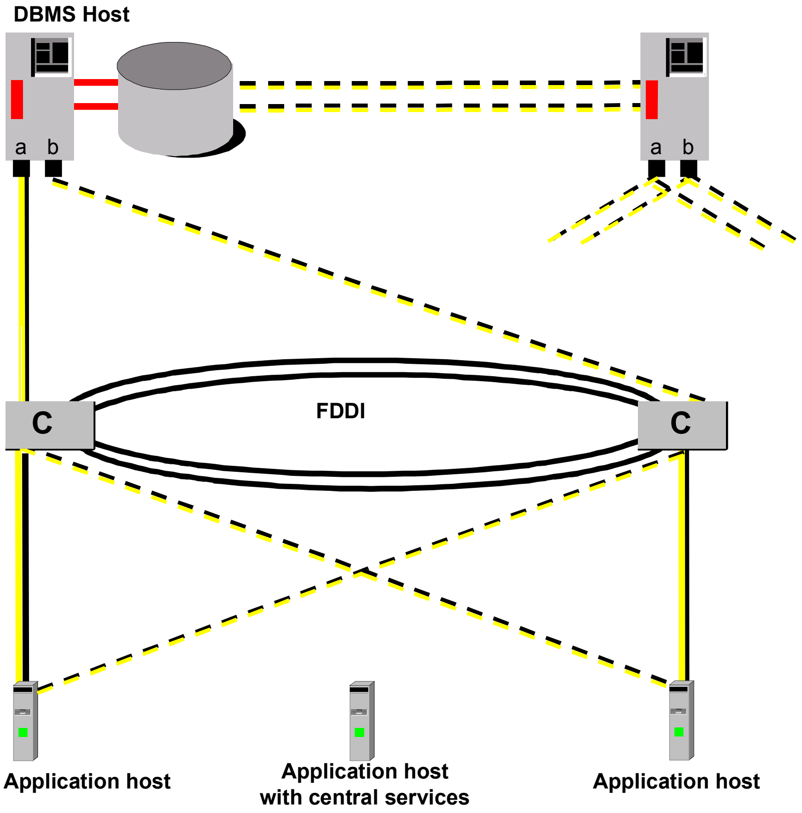
Implement **one** server LAN

For administrative reasons, SAP recommends that you implement a single server LAN (redundant and high speed) rather than multiple different subnets to connect the database host and application host.

Example

The following diagram shows an example configuration of a server LAN with enhanced availability features:

Server LAN with Enhanced Availability



Based on this diagram, we make the following recommendations if you want to further improve the availability of your LAN:

Double attach key hosts to FDDI ring

In the above example, the most important application hosts (that is, those offering critical services such as database and update) are attached via Dual Homing connections. Therefore, a media failure or an outage of an FDDI concentrator would be handled transparently for TCP/IP and all the upper layers of the network.

Install second NIC for database host

To eliminate the NIC (Network Interface Card) in the database host as a single point of failure, you should install a second NIC. This should be in standby mode (not active and without a separate IP address). The task of switching the network connection from the first NIC to the standby NIC should be handled by appropriate switchover. Refer to Switchover Software [Page 216].

Install second NIC for selected application hosts

It is not normally worthwhile to install a second NIC on all application hosts. You should choose application hosts that carry essential, non-duplicated R/3 Services (for example, enqueue service) for a second NIC, if no other measures have been taken to make this service redundant.

See also:

Server Network in a DB2 for OS/390 Environment [Page 119]

Server Network in a DB2 for OS/390 Environment

Use

A possible connection of the server network in a DB2 for OS/390 environment is the Enterprise System Connection (ESCON) channel.

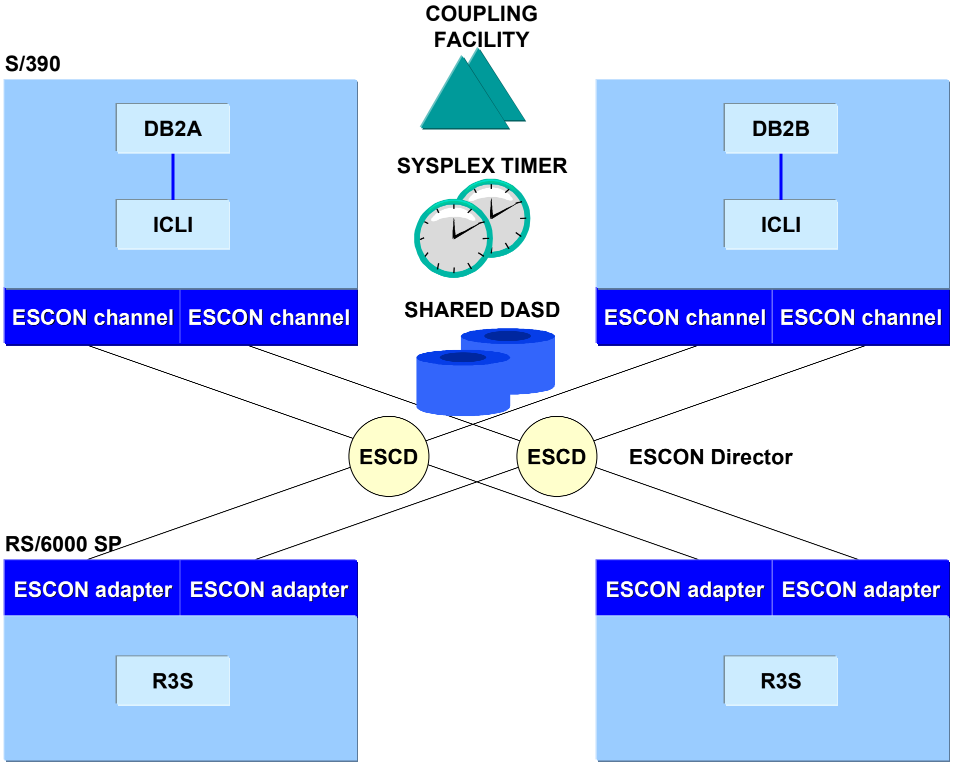
Features

An ESCON connection is a point to point connection and consists of an ESCON fiber cable, an ESCON channel feature on OS/390, and an ESCON adapter on AIX. Optionally, an ESCON Director can be used. An ESCON Director is a switch inserted into the ESCON connection between AIX and OS/390.

Activities

A server network can be protected against failures if you set up the following configuration:

Server Network with ESCON Connection



It is possible to have two ESCON connections between the database server and the application server. In this case, you have to configure two ESCON adapters on application server site and two ESCON channel features on OS/390.

Access Network

Definition

The access network is the network between the application host(s) and the frontends, also known as the company network. In the access network, it is important that the connection between the R/3 frontends and the application host on which the dialog service resides is reliable. Distinguish this from the server network [Page 115], which links the application hosts to one another and to the database host.

Structure

The topology of the access network normally has some degree of built-in redundancy. If connection to a particular application host fails, clients connected to the failed host can reconnect to other hosts, which then take over the dialog service workload.

The exception to this rule is for the host supplying the message service, which should be connected to the access network using a second NIC to deliver high availability. Refer to the recommendation “Install second NIC for database host” in Server Network [Page 115], where the same suggestion was made for the important connection between the database host and the server network. If the message host cannot be reached from the frontends, the frontends configured in the standard way (using the R/3 SAPGUI load-balancing mechanism) are not able to connect to the R/3 System as a whole. Therefore, it is important to provide an extra level of availability here.

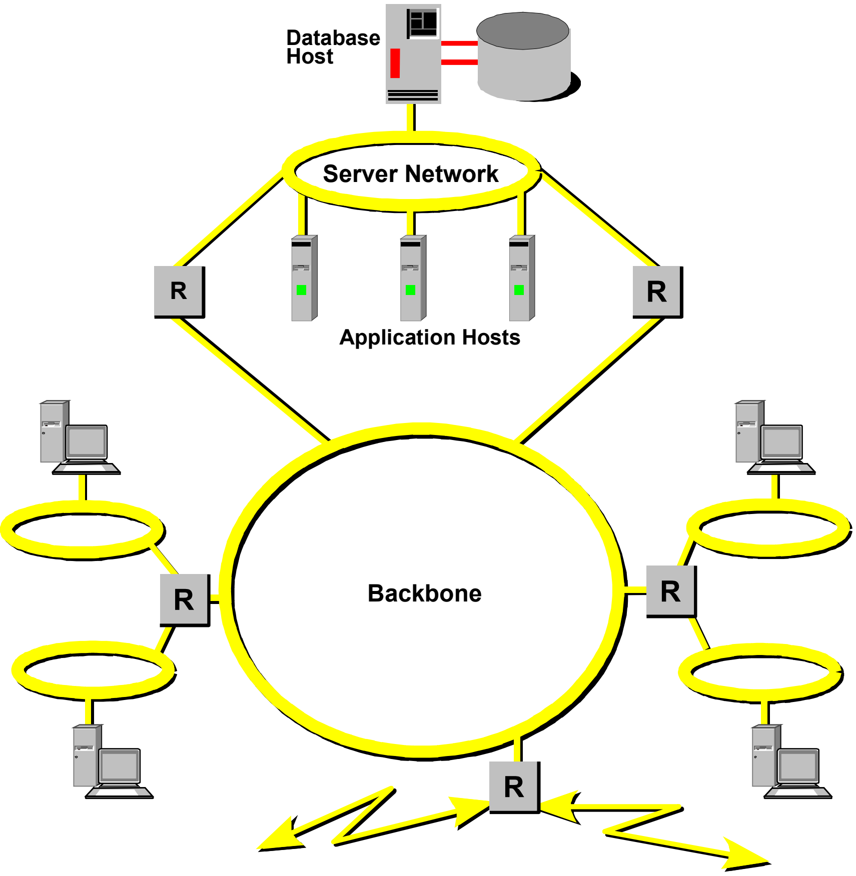
Install a backbone and make it reliable

If there is a backbone installed (as SAP recommends), you should take measures to make it reliable. That is, active components should be reliable and the backbone should be dedicated exclusively to communication between routers and switches. Do **not** plug computers directly into the backbone structure. See below for more details.

Access to Application Hosts Across the Server Network

One approach to connect the frontends on the access network to the hosts on the server network uses routers and is shown in the diagram below.

Backbone Access Through Routers



From this diagram, the following recommendations arise:

Connect backbone to server network using duplicate routers

Access Network

This solution avoids single points of failure. Use multiple routers to avoid having a single point of failure, and configure these to monitor one another’s status.

Configure routers to pass through only R/3 traffic

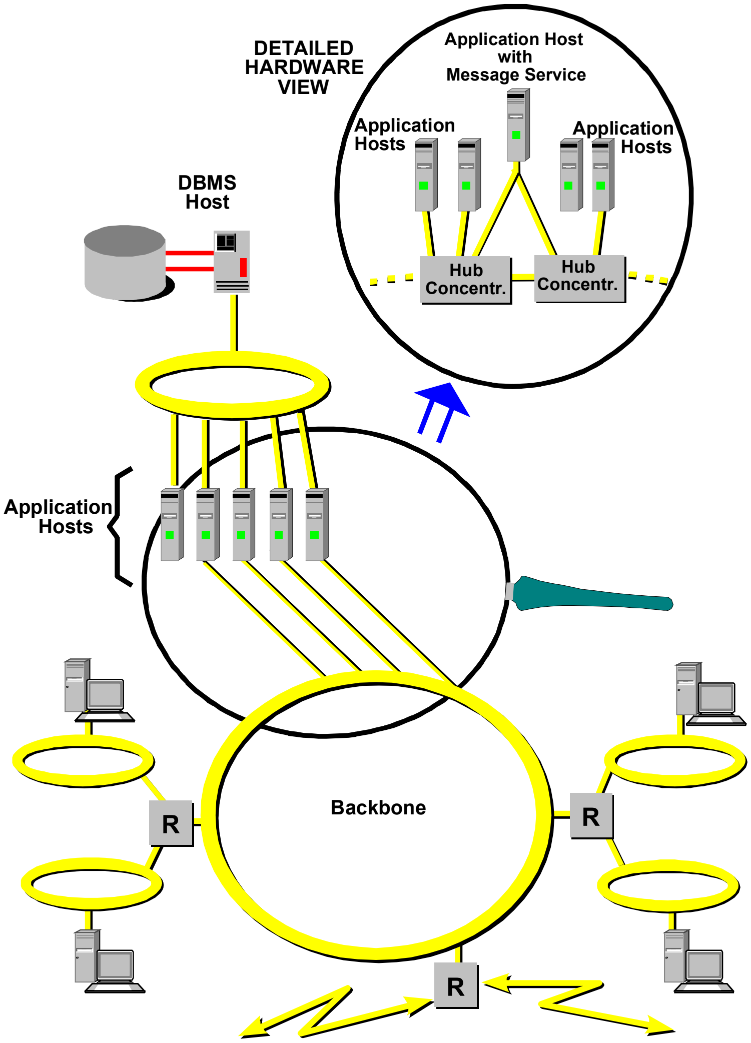
This approach shields the server network from the rest of the data traffic in the company network and improves system security.

The above solution introduces traffic from the access network onto the server network where it is then passed to the appropriate host for processing. From the performance point of view, the impact of the access network traffic is minor, since traffic between the application hosts and frontends normally comprises only 10% of overall network traffic.

Separation of Server and Access Network

However, if you wish to completely separate traffic on the access network from the server network, you can use the configuration shown in the diagram below:

Separate NICs for Server and Access (that is, Backbone) Networks



Notice that this configuration uses application hosts equipped with additional NICs (Network Interface Cards). From this diagram, the following recommendation arises:

Make sure application hosts are redundantly connected to access network

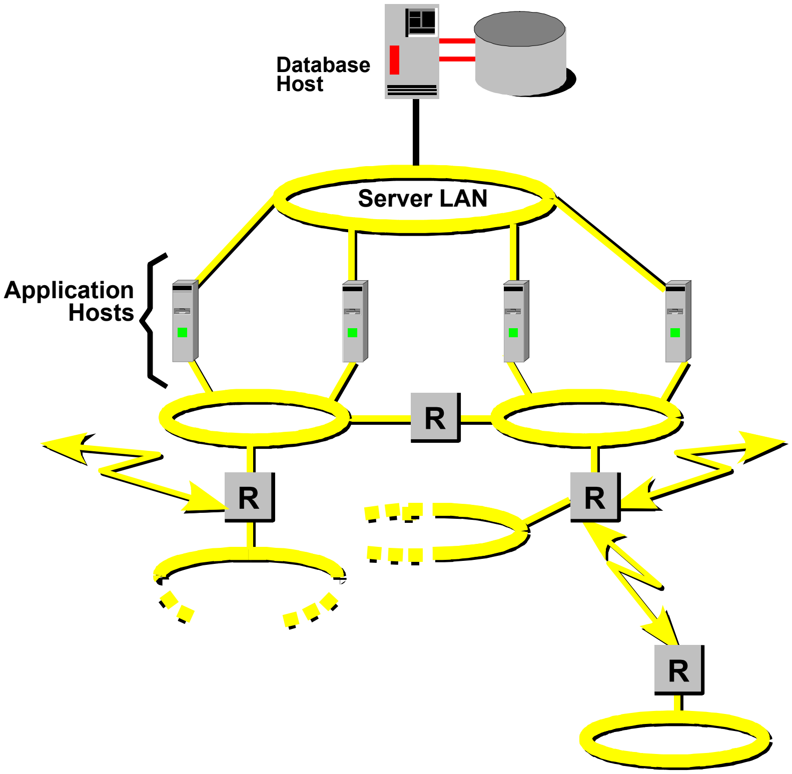
To avoid having a single point of failure, make sure that the application hosts are connected using multiple devices (not just using a single hub/router) to the access network.

If You Do Not Have a Backbone Network Topology

If your network does not have a backbone, the following diagram shows a typical configuration:

Access Network Without Backbone structure (Meshed Access Network)

Access Network



In a meshed access network such as this, consisting of multiple interconnected subnets (often grouped into administrative domains, for example), the application hosts can be connected to different subnets. For the correct functioning of the R/3 System, it is important to make sure that the NICs of the application hosts attached to the different subnets of the access network are interconnected (either directly or through a bridge or router). Redundancy in this configuration is provided by routers that allow for redundant paths through the network and back each other up in the case of failure.

Wide Area Networks (WAN)

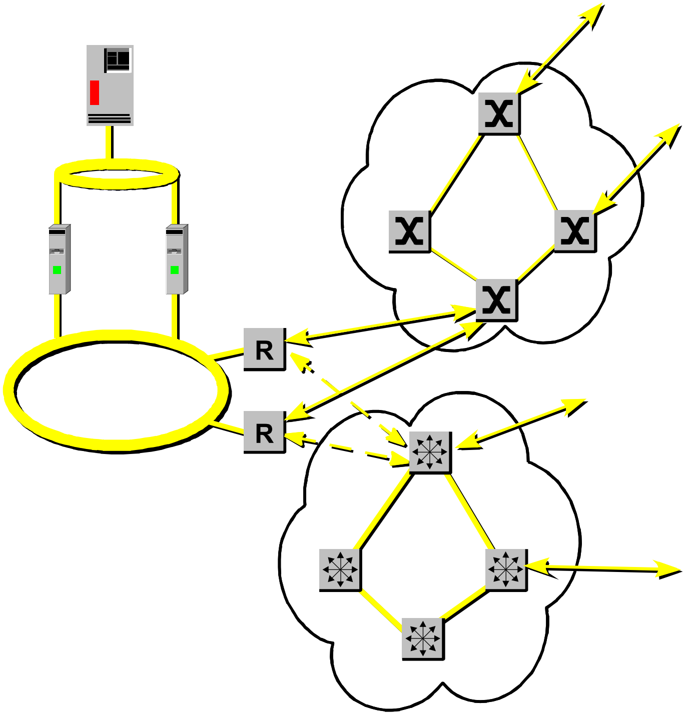
The same basic principles as described above also apply to WAN communication. Redundancy needs to be built into the network if access from decentralized locations is to be guaranteed.

Make sure multiple access paths at central location of R/3

Make sure that the following redundancy is built into the network (see diagram below):

- Multiple access points (that is, routers) should be used
- Different WAN infrastructures should be used (X.25, ISDN, and so on, at least as a backup solution). If multiple routers use the same type of WAN connections, there is likely to be a single point of failure in the area of the network provider (for example, access point).

Availability of WAN Connections: Multiple Access Points

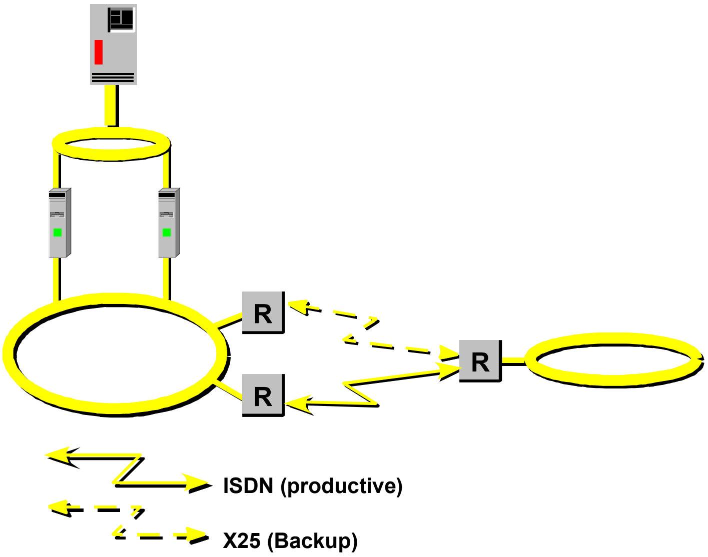


Provide for multiple WAN paths at the decentralized location

You should provide a second WAN infrastructure for backing up the productive connection (for example, you could use X.25 to backup ISDN, or vice versa).

The following diagram illustrates this:

Availability of WAN Connections: Backup Connections to Decentralized Locations

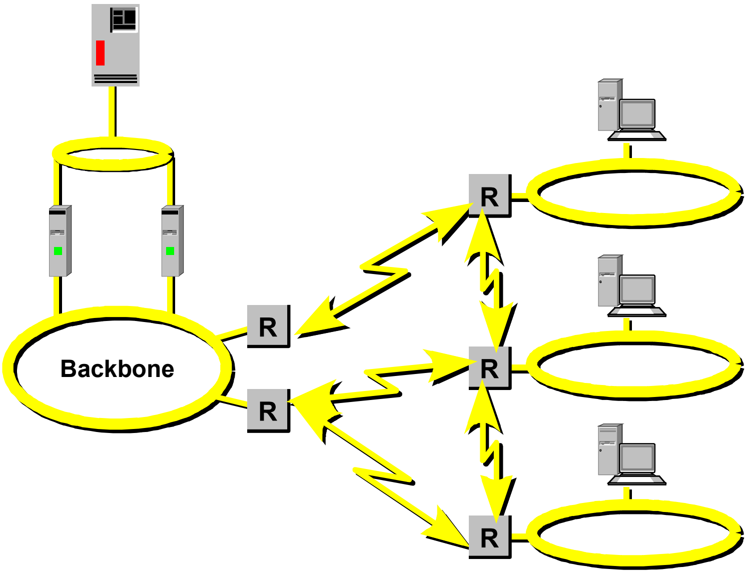


Access Network

Set up redundant WAN paths to other decentralized locations

You should consider setting up communication paths between the different decentralized locations. If the direct communication fails for a particular decentralized location, temporary connection through another decentralized location might then be possible.

Availability of WAN Connections: Redundant Paths

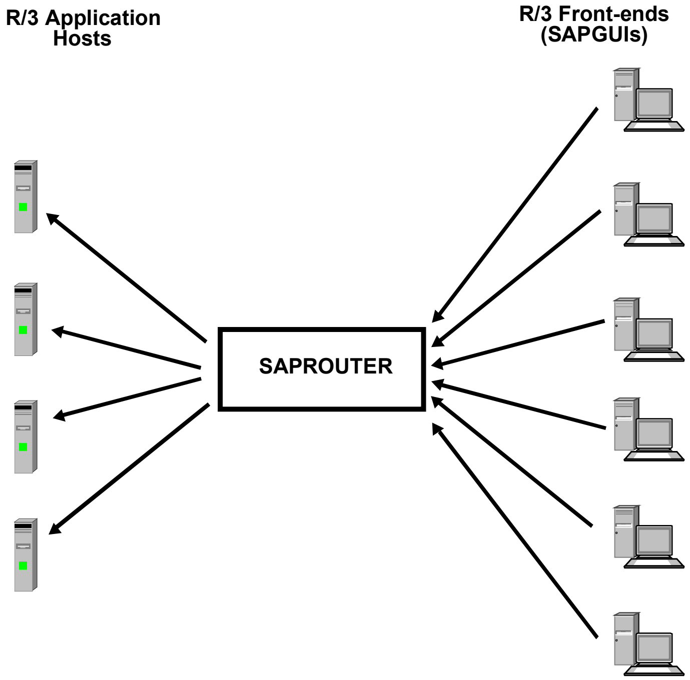


SAPROUTER with Wide Area Networks (WANs)

In WAN configurations, the communication between R/3 frontends (SAPGUIs) and the R/3 System is often implemented using SAPROUTERs. The SAPROUTER is an application program included in the standard R/3 software shipment that fulfills the task of a “proxy” (or application level gateway) for the SAP datastream.

Compared to a direct frontend connection, the use of intermediate SAPROUTERs offers you additional security features and shields the handling of WAN links from the R/3 application hosts. A single SAPROUTER can handle a large number of parallel GUI sessions with multiple application hosts, as shown schematically in the following diagram:

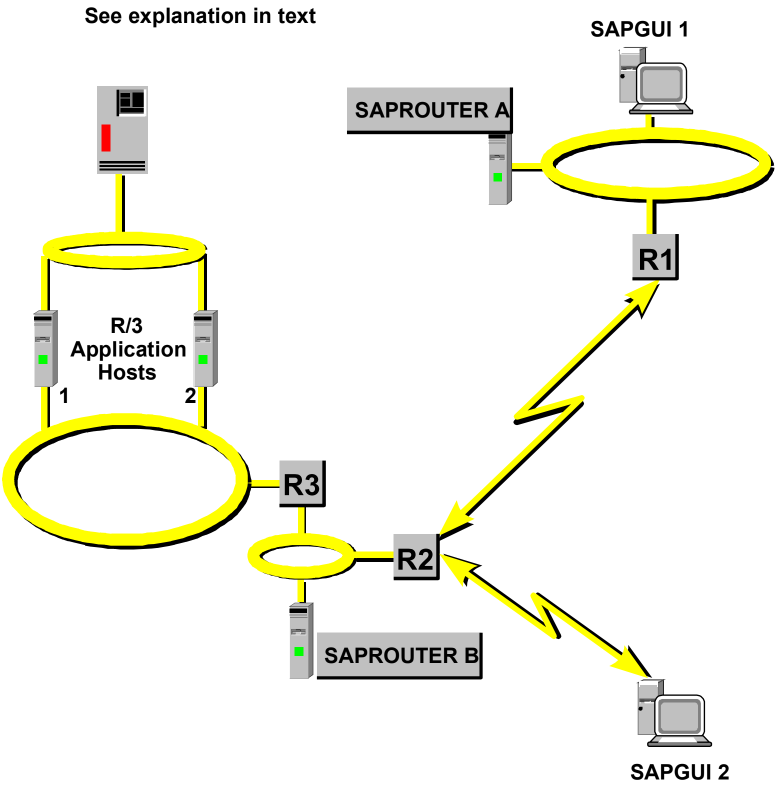
Single SAPROUTER Handling Multiple SAPGUIs and R/3 Application Hosts



When routing network traffic, the decision whether or not to use a SAPROUTER (or a chain of SAPROUTERs) is taken at the frontend (that is, on the SAPGUI). The location of the SAPROUTER processes included in the communication path between the frontend and the application hosts has to be defined in SAPGUI start parameters (some kind of “source routing”). A SAPROUTER is often installed on separate machines (so-called “firewall computers”) although it can also be installed in an application host machine. The following diagram illustrates typical use of SAPROUTERs with R/3:

SAPROUTERs in WAN Setup with R/3

Access Network



To reach either of the R/3 application hosts, traffic from the SAPGUIs is routed as follows:

- SAPGUI 1

  Using SAPROUTER A, routers R1 and R2, SAPROUTER B and router R3.

- SAPGUI 2

  Using router R2, SAPROUTER B and router R3.

If the entire communication traffic for a given group of frontends is redirected through a single SAPROUTER (or a chain of SAPROUTERs), this SAPROUTER becomes a single point of failure for the given group of frontends. For example, all the frontends in the above diagram depend on SAPROUTER B. If this SAPROUTER fails, none of the dependent frontends is able to connect to the R/3-System (without changing the SAPGUI start parameters). Measures to avoid this problem are discussed in the next section, “Measures for Improving SAPROUTER Availability”.

The use of a direct SAPGUI session without SAPROUTERs (one possibility, if a SAPROUTER fails) is often not allowed because the security measures taken to protect the internal network do not allow for direct sessions in most cases.

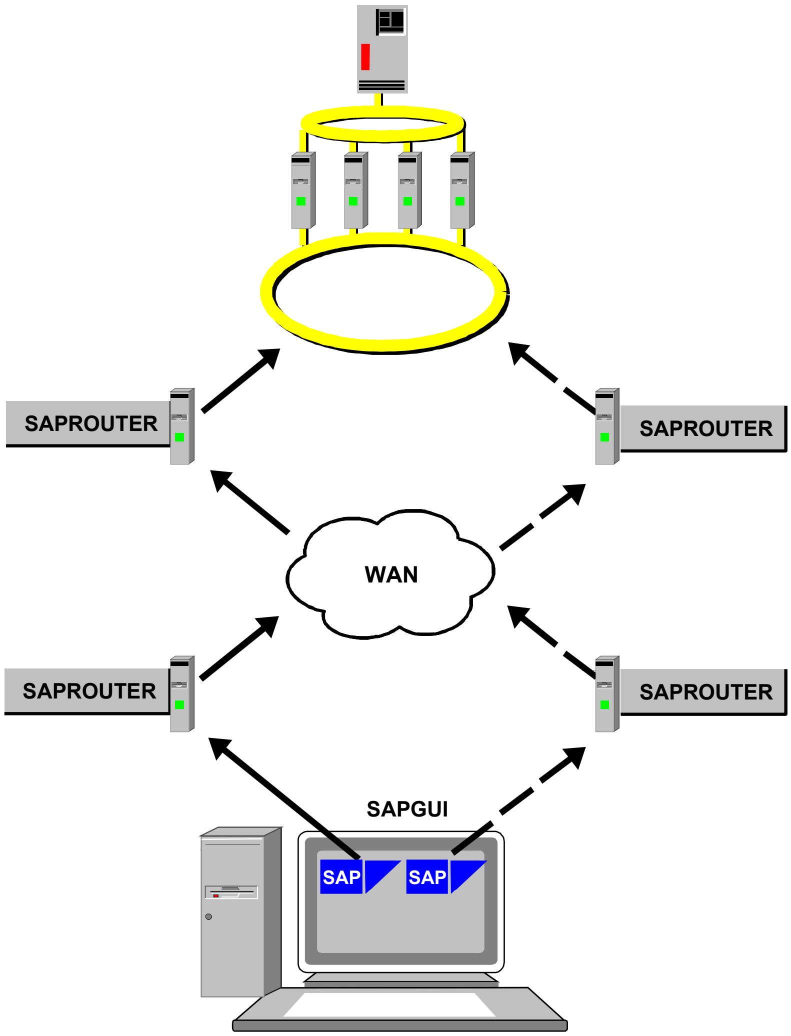
Measures for Improving SAPROUTER Availability

There is no load balancing mechanism between multiple SAPROUTER processes (as exists for multiple R/3 application hosts, for example). Instead you can take the following measures:

- Implement standard high availability measures for the machines running the critical SAPROUTER processes. See, for example, Switchover Software [Page 216].

- Provide for a redundant SAPROUTER route (that is, two SAPROUTER processes on each of the affected machines, or still better, two SAPROUTERs on two separate machines). To make use of these two alternate routes, there must be two separate SAPGUI icons on each of the frontends (the parameters relating to each icon describe one of the alternate routes). If the route described by one icon fails (shown with solid lines on the left-hand side of the diagram below), the user can click on the second icon to use the alternate route (shown with dashed lines on the right-hand side of the diagram below). The following diagram illustrates this approach:

Redundancy using SAPROUTER



Hardware and System Software Key Issues

Purpose

This section describes high availability for the hardware and system software that support your R/3 System.

Implementation Considerations

For detailed technical guidance when implementing a specific product or feature, contact the appropriate source, such as your SAP consultant or your network supplier.

Integration

High availability for the R/3 System should be part of a system-wide strategy. Therefore, you should also consider other components of the system, such as the R/3 System itself, the database management system (DBMS), the network, and so on.

Cluster Technology

Use

This section discusses the software and hardware aspects of cluster architectures relevant for R/3 Systems. A cluster consists of a small number of host machines, known as cluster nodes. The nodes can be single-processor machines or multi-processor machines, for example, Symmetrical Multi-Processor (SMP). Cluster technology is only used for R/3 System application hosts or DBMS hosts.

The following aspects of cluster technology are relevant to R/3 Systems:

- High Availability
- Scalability

This section focuses on high availability.

Features

In general, cluster hardware for R/3 Systems can be divided into the following categories:

Cluster Hardware for the R/3 System

	Shared Storage	Message Based
I/O Attached	Physically shared disk systems	Virtual shared disk systems
Memory Attached	Shared memory systems	—

The physically or virtual shared disk systems are relevant for current R/3 Systems.

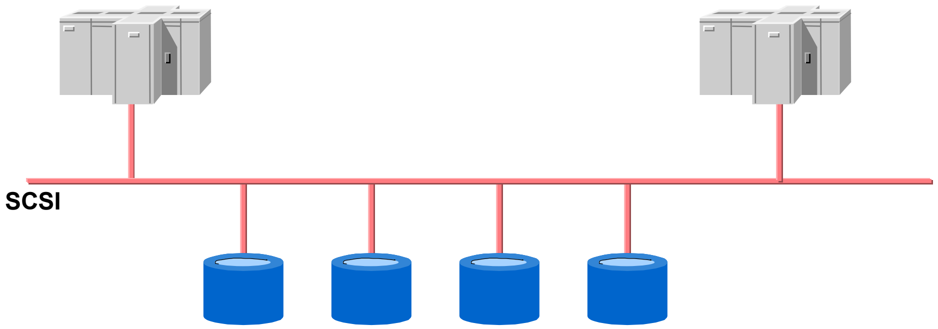
The new architectures using shared memory clusters – for example, “Scalable Coherent Interconnect Technology” (SCSI) – where several cluster nodes share memory by means of a software or hardware layer, might be used in the future with R/3 Systems.

Shared Disk Systems

- Physically-shared disk system

  In this type of system, the SCSI disk controller is directly connected to two (or more) computers, as shown below:

Physically-Shared Disk System

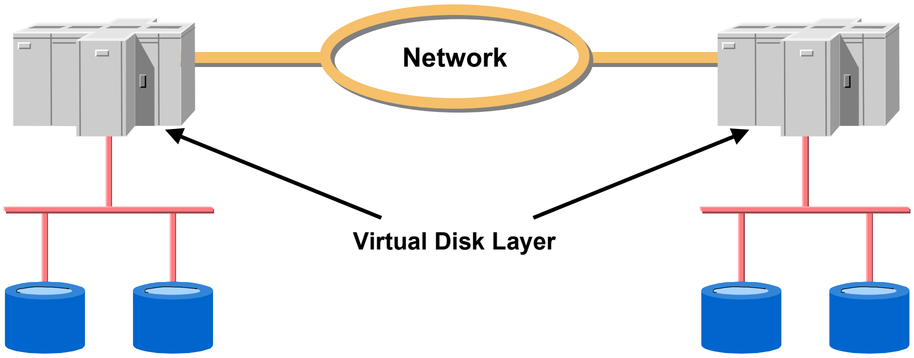


- Virtual-shared disk systems

  In the case of virtual-shared disk systems a software layer (the virtual disk layer) enables access to disks connected locally on another host machine.

  Some MPP (Massively Parallel Processing) systems, the main characteristic of which is a large number of processors, use a virtual disk layer. The virtual disk layer in an MPP system theoretically allows the sharing of disks between a large number of cluster nodes, as shown in the following diagram:

Virtual-Shared Disk System



Concurrent and Non-Concurrent Disk Access

The access to the shared disks can be concurrent or non-concurrent. If concurrent access is allowed, this is handled by distributed lock manager software, which is usually tightly integrated in the operating system.

An example of non-concurrent access is the use of switchover software. Refer to Switchover Software [Page 216]. Examples of concurrent access are Oracle Parallel Server and Sysplex failover support for DB2 for OS/390. Refer to Replicated Database Servers [Page 208].

Cluster Communications

From the high availability viewpoint, a high-speed communication network between the cluster nodes – faster in terms of bandwidth and latency than FDDI networks – is not necessary for the server traffic in an R/3 System. For more information, see Network System Key Issues [Page 108].

High-speed communication networks are needed for low-latency communications between the cluster nodes (for example, distributed lock manager communications).

Clusters or SMP Machines?

From a high availability viewpoint the question of the advantages and disadvantages of cluster systems versus SMP systems can be simply answered. SAP recommends you to run the R/3 System on a cluster of SMP machines. The use of a cluster of SMP machines (that is, the cluster nodes are SMP host machines) for an R/3 System increases both availability and performance.

Consider using switchover software with a cluster

The use of switchover software increases the availability of an R/3 System running on a cluster. Refer to Switchover Software [Page 216].

Disk Technology

Hard disks lie at the basis of data storage and conceptually represent a critical single point of failure in an R/3 System. Disk backup schemes – including database backup, described in the sections on database backup and recovery in Database Key Issues [Page 45] – only serve the very minimum of data availability requirements. Disk data in an R/3 System must often be made significantly more available. This section gives you an overview of suitable disk technology and practical guidelines to help you improve the availability of your disks in an R/3 System environment.

SAP advises you to consult your hardware vendor for more detailed information on specific solutions. This is because the area is complex and the information in this documentation, although written for the R/3 System, is of a general nature.

The Need for High Availability of R/3 System Data

You need to be clear about the minimum status of R/3 System data availability that your business can tolerate after a disk hardware crash. The following helps you achieve this:

- Minimum high availability requirements

  If your high availability requirements are quite low, you might be able to afford the unplanned downtime taken to recover the R/3 System database from backups following a disk failure. You then need to consider the following steps:

  – Replace the failed disk (is one immediately available?)

  – Restore backup data from storage media (often slow tape devices)

  - Recover the database to guarantee data integrity in your R/3 System

  The restore and recovery operations depend on the size of the database and the quality of backups available (for example, replaying transaction log data is more time-consuming than using more recent backups).

- Maximum high availability requirements

  Installations requiring a more rigorous level of availability can identify two differing levels of online data redundancy:

  - One level redundancy

    Data remains online and fully available despite a single failure in a disk drive. This means that an extra copy of the data is constantly maintained for use in the event of a failure.

  - Two level redundancy

    Data remains online and fully available despite two independent disk failures. This means that two extra copies of the data are constantly maintained.

    There are particular problems with very large databases (for example, excessively long times to backup or restore the database) that can be alleviated by using two levels of disk redundancy. Backup is made easier and the chances of having to do a time-consuming restore are much reduced.

    Use two levels of redundancy if downtime must be avoided at all costs

    For highly sensitive applications you should consider migrating to two levels of redundancy for your disks (for example, three-way mirroring, see below).

Performance and Costs of High Availability Disk Systems

Few customers are willing to sacrifice system performance for high data availability. However, the requirement for high availability implies redundancy, which in turn implies that data must be checked and/or written multiple times to disk. To maintain performance in a high availability environment, you often need to consider improving the underlying hardware in terms of:

- Speed of data handling
- Intelligent disk controlling
- Increased disk caches
- Data bus throughput
- Total disk capacity
- CPU computing power

  Consider high availability for disk systems seriously, despite costs

  The costs involved in upgrading hardware often pay back very well. Remember that, if the system is not available, you incur costs due to loss of income, support and service required.

Data Availability in a Failure Scenario

The crucial discriminator of quality for high availability purposes is what happens in the event of failure. How does the disk system in a high availability environment react (in terms of software and hardware), if the redundant part of the data must be accessed to continue processing?

It is almost inevitable that online performance is degraded in case of failure because a heavy workload is imposed on either the disk hardware itself and/or the host machine CPU due to the following points:

- Data must be re-constructed from check information
- On-disk redundancy of the data must be re-created (mirror rebuild)
- A failed disk must ultimately be replaced and then recommissioned.

Depending on the kind of disk technology used, a period of decreased performance must be expected. Choosing an unsuitable disk device for high availability purposes can hinder you from accessing the data in case of failure, simply due to unacceptably impaired performance.

Note that it is a common feature of disk systems with one level of redundancy that the risk of data loss is increased as soon as one failure has occurred. This is mainly for the following reasons:

- Active drive redundancy is lost
- Recovery work puts additional heavy work load on the system
- Complex situations might arise in hardware or software

Mean Time Between Failures

A key characteristic of a hard disk is its “Mean Time Between Failures” (MTBF) given in hours (as an approximate rule, 100,000 hours = 11 years). This is the mean statistical average of the time a disk device operates until it fails. For example, assume a disk device with an MTBF of 50 years. This implies that a company with 25 disks of this brand suffer roughly one disk failure every two years. Statistically there is absolutely no guarantee whatsoever, that one specified disk does not crash within the first week of its life. Statistics only tell you that this event has a comparatively low probability.

Note that there might be a significant difference between the MTBF of one disk spindle itself and the MTBF of a disk system taken as an entity. A disk device contains a lot of components that might all fail. For example, many disk devices contain several disk spindles (JBOD and RAID systems). In this case the total effective MTBF of the device goes down roughly in proportion with the number of spindles in the array.

Specified values of MTBF for a given device are often calculated from the statistics of pure hardware crashes. In reality however, data unavailability can arise from more complex problems that are not included in these figures. Therefore the MTBF can only serve as a guideline to compare pure hardware resilience.

How to Achieve High Availability

Since every sub-component of a disk system can fail, how can you achieve a maximum of system availability? In general, high availability is achieved by hardware and data redundancy (software is considered as data in this sense).

As a guide to hardware aspects, consider the following:

- Use hardware with high MTBF (“Mean Time Between Failures”)

- Avoid single points of failure
- Use fault-tolerant hardware and/or redundant components

There are two basic methods of redundant data storage:

- Mirroring, where identical data are physically stored twice (or multiple times)
- Check bits, where check information is computed from input data and stored to retrieve the full set of data in case part of it is lost

The disk-based components listed below must be considered as critical to the availability of your R/3 System. Depending on whether the component is read or write intensive, you need to select a suitable method for protecting the data (see discussion below).

- R/3 System user data (normally write-intensive but depends on application)
- R/3 System system data (normally write-intensive but depends on application)
- Software components:
  - R/3 System (normally read-intensive)
  - DBMS (read-intensive)
  - DBMS log files (very write-intensive)
  - Operating system (read-intensive)
  - Operating system swap space (write-intensive)
- Root file system (normally read-intensive)

The pros and cons of the different mechanisms of redundant data storage and the hardware aspects are discussed in detail below.

General Options for High Availability Disk Technology

There are two principal technologies for storing data redundantly in high availability disk systems:

- Hardware-driven redundancy, that is, RAID

  This comprises the whole range of RAID technology (“Redundant Array of Inexpensive / Independent Disks”):

  - Hardware-emulated mirroring on RAID disks (RAID level 1, see diagram below)
  - Error-correcting technology (check bit computing) on RAID systems (levels > 1)

  An intelligent disk (that is, controller, storage processors, and microcode) acts independently from the host machine CPU to make sure that data can be retrieved from the device even if one of the disk spindles fails.

- Software-driven redundancy, that is, LVM

  This is the regime of add-on software for the operating system, namely “Logical Volume Manager” software (LVM). CPU cycles are stolen from the host machine to run LVM software, so ensuring that data is written to physically redundant (mirrored) locations on disk(s). Underlying disk technology can be either several physically independent (standalone) disks or an array of disks sharing common hardware resources (disk tower, rack or cabinet, JBODs) but with no intelligent RAID controller installed on the device.

Hardware-Driven Redundancy with RAID

RAID stands for “Redundant Array of Inexpensive/Independent Disks”. While appearing logically to the operating system as a single disk drive (that is, user-transparent), a RAID device internally consists of an array of disk spindles with one single intelligent disk controller. The controller and its software (the microcode) handle the data distribution onto the disk array, the redundant storage of data and the computation of check bits inside the array. Often RAID arrays contain optional storage processors (CPUs) to assist in performing data handling inside the array.

Garth Gibson and Randy Katz set out this classification of RAID “levels” in the SIGMOD paper of 1988. They defined RAID levels to distinguish different methods for creating check information and distributing the data (and check bits) across the disk spindles. Note that level numbers do not imply any rating of redundancy quality or performance whatsoever. An overview of the RAID levels most relevant for the R/3 System is given in the table below:

RAID Levels Relevant for the R/3 System

Level	Description	Advantages / Disadvantages
0	**Striping** Data segmented and distributed across several disks	+ Increase in performance due to parallelism in read and write − No redundancy (not a high availability solution)
1	**Hardware Mirroring** Data written twice (or three times) to different disk spindles within the disk array	+ Good performance in read-intensive applications (data can be read in parallel from several disks) − Slower in writes (multiple writes required) − Spindle costs doubled
10	**Striped Mirroring** Combined level 0 and 1. Data mirrored onto and striped across several disks (sometimes known as RAID 1/0)	+ Good performance in reads (RAID 1) − Write performance improved compared to RAID 1 because of parallelism − Spindle costs doubled
5	**Check Bit** Check bit calculated from data, check bit and data distributed (that is, striped) across multiple disks	+ Good performance in reads due to parallelism (like RAID 0) − Costs only slightly increased compared to disks without high availability solutions − Write performance penalty due to check bit calculation

Systems with RAID levels 3 and 4 have also become commercially available in recent years. However, they normally do not suit OLTP (online transaction processing) application software like the R/3 System. In addition to the initial RAID definitions, further vendor-specific levels have been introduced. Several hardware vendors supply devices with configurable RAID level, so adding a further degree of flexibility to the configuration of the disk system.

Performance of RAID Systems

The general benefit of RAID systems (that is, hardware-driven redundancy) is that data storage is effectively handled solely by the disk hardware. There is no additional overhead on the host machine CPU since optimization of data storage occurs independently in the array.

The performance of disk devices in general depends on the following factors, several of which are particularly relevant to RAID:

- Ratio of read-to-write access
- I/O size
- Access pattern (random or sequential)
- Parallelism in data bus transfer and disk spindle access
- Possible bottlenecks
- Caching

Read-to-write Ratio

Write performance is a critical point in any disk system that tries to hold information redundantly. This is simply due to the time overhead implied by:

- Computing the check bit from the data
- Storing the additional check bit
- Writing the data to physically different locations (that is, maintaining mirror copies)

Therefore, the read-to-write ratio is of special importance to high availability disk systems. Decision support systems typically show a very high read-write ratio (nearly all reads, with write access often by periodic batch update) whereas R/3 Systems with heavy OLTP workload are often far more write-intensive.

RAID 5

The write-penalty of RAID 5 is well known. Generally, RAID 5 performance is very sensitive to the characteristics of the application as follows:

Characteristics of RAID 5

Access mode	Comments
Small I/O	Fast in random reads (parallelism)
Large I/O	Slow, cannot be parallelized
Writes	Slow
Small writes	Inefficient, need a read-modify-write sequence (that is, read stripe, modify appropriate stripe region, compute new parity and write back)
Large writes	More efficient, calculate check and rewrite the entire stripe
Reads	Relatively fast (parallelism), fastest in small random chunks

The type of access mode that dominates in a given R/3 System depends on how the system is set up and cannot be stated in general terms. However, note that RAID 5 provides low performance in write-intensive environments.

RAID 0 and 1, RAID 10

Striping (pure striping is RAID 0) is an efficient way to parallelize access and speed up reads and writes by concurrent I/O. It can handle small I/Os very effectively. However RAID 0 alone does not provide any redundancy and is not a high availability configuration. Note that striping is also used in RAID 5.

The read performance of RAID 1 (hardware mirroring) can be increased by reading from either side of the mirror. Its performance can be further increased in combination with RAID 0, which is then called RAID 1/0 or RAID 10 (striped mirroring). Due to its simplicity in implementation (compared to RAID 5), RAID 10 is a robust technology. Its chief drawback is the 100% increase in spindle costs, because mirroring requires double the disk space. RAID 10 also seems to provide performance advantages in most read/write environments in comparison to RAID 5, but this needs to be verified in concrete OLTP benchmarking for a given installation.

RAID with Write-caching

Write-caching is an appropriate way to compensate for write-penalties encountered with RAID arrays. This is because the host machine CPU only needs to wait for the write cache to accept the data, as the RAID device then takes over the writing of data from cache to disk. Large write caches can completely avoid the write penalty. Note that caches can be critical single points of failure. See below for further discussion of caching.

Failure Scenarios with RAID

In the event of failure, some degradation of performance is to be expected with high availability solutions for disk systems. The following points are specific to RAID systems:

- Overall RAID 5 performance is degraded, because controller computations are required and access requests are queued in front of the controller, causing a bottleneck (writes are especially affected).
- Read performance of RAID 5 is heavily degraded, because data must be recomputed from check bits.
- RAID 1 (and 10) read performance must not necessarily be degraded, because all information is stored twice and can still be directly accessed.
- RAID 1 (and 10) performance is degraded during mirror rebuild, when the failed disk has been replaced.
- Replacement of disks can often be undertaken “hot”, without shutting down the system (automatic swap to a dynamically configured standby disk is possible).
- RAID systems employ an internal mapping system to store the data in the actual physical location and the map is a critical single point of failure.
- Redundancy rebuild might last an extended amount of time, depending on the present workload and the amount of information lost.
- A failed RAID drive spindle must ultimately be replaced so service might be needed.

Controller and Microcode

It is the task of the controller and its microcode to create the check information and/or distribute data and check bits across the disk spindles in the array. It is clear that both hardware and software are critically important single points of failure. Configuration inconsistencies and incompatibilities between microcode and hardware can cause unforeseen problems and risks. Due to its internal complexity, RAID 5 has been prone to microcode problems, while RAID 1 and 10 have been relatively robust in this respect.

When upgrading RAID microcode, test using a non-production system

If RAID microcode needs to be upgraded it is strongly recommended to check its performance and reliability on the contents of an uncritical disk first. Updating microcode on a productive R/3 system without previous checks might put vital data at risk.

Summary of RAID

This section gives you a quick overview of the advantages and disadvantages of RAID systems followed by an approximate guide as to when RAID might be a good idea.

- Advantages of RAID systems
  - No host machine overhead as hardware works independently of CPU
  - Large theoretical storage capacity (generally true for disk arrays, either RAID or JBOD)
  - Hot spindle swapping possible
  - Configuration flexibility
  - Potential for high performance in read-intensive environments
  - Financial costs per effective megabyte only moderately higher in RAID5 than without redundancy
  - Performance benefits of RAID 10
- Disadvantages of RAID systems
  - Write penalty of RAID 5
  - Performance dependence on I/O specification in RAID 5
  - Little control over data placement for performance tuning
  - Microcode is a critical single point of failure

RAID can be a good solution in the following situation:

- Application almost 100% CPU-bound
- Read-intensive application
- Costs must be kept right down
- Hot spindle swapping required without system administrative action

Software-Driven Redundancy with LVM

The use of standalone disks with software-driven mirroring using LVM (“logical volume manager”) is the chief competitor to hardware-driven redundancy using RAID systems.

Disk Devices for Software Mirroring

There are two major types of disk devices used with LVM:

- “Single Large Expensive Disk” (SLED)

  Such standalone disk devices have their own controller, own SCSI interface, own power supply and so on. Each standalone disk is a physically separate device.

- “Just a bunch of disks” (JBOD)

  This uses an array of several disks sharing one rack, power supply and a standard disk controller (instead of a RAID-like controller) to handle the data I/O. There is only one physical SCSI path to the whole device. This is why, in contrast to SLEDs, JBODs offer a high limit in terms of total storage capacity. Devices like these appear as a simple standalone disk to the outside world.

Software Mirroring with LVM

LVM (Logical Volume Manager) software mostly comes as an add-on to or part of the operating system. It provides a general address space for a number of distributed hardware disks and/or partitions. When used for mirroring, the software associates a single logical volume (can be either a file system or a raw-device) with multiple physical copies in a way that is transparent to users and applications (there are also RAID 5-like configurations of LVM, but these are generally CPU-intensive). Commercial relational database systems support LVM software.

The two (or three) mirror copies in a mirror disk system should be placed on disk spindles that are physically as independent as possible (different standalone disks or at least different spindles in an array). If sectors of a disk (or the disk as a whole) containing one copy of the data should fail, the data still remains accessible via another copy (on another disk).

Although not necessary for mirroring, some LVM software is also capable of handling redundant SCSI paths to the device (that is, switchover to the redundant path if one path fails). It is recommended to take advantage of this feature for high availability.

Performance with LVM

One disadvantage of LVM is that, as part of the operating system, it steals CPU cycles from the host machine. This is in contrast to hardware-driven redundancy solutions like RAID where no CPU overhead is involved. In mirroring mode LVM instructs the operating system to physically send data to the storage media twice (or three times in three-way mirroring). The loss in overall performance is claimed to be in the order of only a few percent. This means that the LVM overhead only impacts your R/3 System strongly if it is almost 100% CPU-bound.

With LVM (as with RAID) performance degradation depends on the read-to-write ratio, with write-intensive applications causing the most significant deterioration in performance, as described below:

- Write access is degraded due to the need to write data multiple times. Write performance is dependent on whether file updates are configured to be handled in “asynchronous” or “synchronous” mode. Asynchronous mode means that control returns to the executing program after the file has been written once. After one copy of the file has already been written to disk, it is safe to write the mirror copy asynchronously “in background”.
- Read access can be improved with LVM, since either side of the mirror can serve the read request.

LVM (as with RAID 0) can be configured to stripe data and so parallelize disk access for improved performance. This is especially effective if work can be striped to several independent disk controllers and/or to different SCSI channels with different disks attached. The applications that benefit most from striping have a large number of random access and large sequential read/writes (if your version of LVM enables you to parallelize a large I/O request internally). Also, applications with large database tables on different disks that can be accessed in parallel normally benefit from striping.

Three-way Mirroring using LVM

Whereas LVM software is normally configured for two-way mirroring, providing a one level redundancy solution (that is, one disk failure can be tolerated), it is also possible to configure it for three-way mirroring (that is, one original plus two redundant copies of the data). The result is two levels of redundancy, since two disk failures can be tolerated. This configuration can be used for R/3 Systems with very strict high availability requirements and allows the following operation modes:

- Three data disks online in normal operation

  Redundancy is not lost after a single disk failure, since two simultaneous disk failures can be tolerated (that is, two levels of redundancy).

- Two disks online, one split off for backup

  R/3 System stays fully online with one level of redundancy during backup (that is, one additional disk failure remains tolerable during the backup period).

Three-way mirroring with LVM can be used to enable an offline database backup with a very short interruption to operations. The DBMS must be very briefly shut down, described in Backup with Oracle [Page 60] To do this kind of backup, the third mirror copy is split off for the backup, allowing the two remaining copies to continue providing normal database service. Three-way mirroring provides the best data availability but is also the most expensive solution in terms of performance and financial costs. Data needs to be written three times to disk in total with a corresponding impact on CPU performance. The costs per megabyte of storage space is increased by a factor of approximately three, because three times the normal disk space is required.

Three-way mirroring is one of the few remaining options to perform backups in an acceptable period of very large R/3 System databases (VLDBs) that are permanently under heavy transaction processing load.

Failure Scenarios with LVM

After a failed disk or logical volume has been replaced, LVM must resynchronize the mirrors. Direct impact on CPU performance is said to be minor but all I/O on the failed logical volume is inevitably impeded. Since R/3 System data is highly integrated, a general loss of performance is to be expected in this situation. In this respect, LVM is similar to RAID, which also runs into a bottleneck during failure, although the problem with RAID is situated in the disk array rather than in the host machine.

A drawback of current LVM-based disk systems is that hot disk swapping is generally not as well supported as with RAID. Even if the host machine often need not be completely shut down, the R/3 System must normally be quiescent during disk replacement. It is sometimes possible to configure hot spare spindles on each side of the mirror to emulate some kind of hot swapping mechanism with LVM. Automatic swapping is hardly supported with LVM configurations.

Summary of LVM

This section gives you a quick overview of the pros and cons of LVM systems followed by a short guide as to when LVM might be a good idea.

- Pros of LVM systems

  - Two or three-way mirroring possible
  - Backups possible by splitting off one mirror while data stays online at full performance (with additional redundancy if three-way mirroring used)
  - Each disk has its own full set of hardware components (that is, component failure can only affect one disk), more hardware redundancy
  - Total physical control on data placement for performance tuning
  - Multiple independent disk controllers, no bottleneck, concurrent I/O possible, no single point of failure
  - LVM read, performance boost by reading from either side of the mirror
  - Striping possible, that is, concurrent I/O on different data channels or disks

- Cons of LVM systems
  - LVM in principle leads to performance degradation because of host machine CPU overhead, but only of the order of a few percent
  - LVM write, performance degraded because operating system needs to send data multiple times to disks
  - Each standalone disk consumes one SCSI target address, total capacity on bus is limited, if SLEDs are used (JBODs can be used to increase capacity)
  - Failed disk replacement less convenient, might require shutting down the R/3 System
  - Probably more administrative action required (compared with RAID systems), if swapped disk returns to service
  - Hardware costs per MB increased, since twice as many disk spindles are used (100% cost increase for SLEDs, less for JBODs)

    LVM might be a good solution in the following situation:

    - Application well under 100% CPU-bound
    - Control on data placement required for performance tuning
    - You wish to eliminate all single points of failure
    - Optimal write performance to disk required

Comparison of Software-Driven and Hardware-Driven Redundancy

The decision as to whether you employ software-driven (LVM) or hardware-driven (RAID) redundancy can be a difficult one and depends on the details of your particular installation.

Analyze your installation thoroughly before choosing a solution

This documentation can only give you guidelines as to whether software or hardware redundancy is preferable for your installation. You should make the final decision only after consultation with your hardware vendor, having taken the details of your installation fully into account.

The following describes two extreme cases to illustrate the issues involved:

- Maximal CPU performance required, write access to disk less critical

  Raid systems (preferably RAID 10) are normally preferable due to the following factors:

  - Data handling is done by hardware situated internally on the disk array
  - CPU performance is unaffected
  - Rebuild does not put load on CPU in case of failure
  - Ease of service with disk replacement (disks can be swapped “hot”)
  - RAID 5 offers low prices per usable effective megabyte of disk space but low write performance
  - Devices with a large write cache are preferred

- Maximum write performance to disk required, CPU performance less critical

  LVM mirroring on standalone disks are normally preferably due to the following factors:

  - CPU cycles for LVM can be sacrificed
  - LVM striped over several disks to parallelize output and maximize performance
  - LVM configured to read from either side of the mirror
  - Preferably use device with large write cache
  - Possible options to split off third mirror for backups in three-way mirroring
  - For minimal disruption in case of failure have hot disks ready on each side of the mirror

  If you plan to use both types of disk redundancy, LVM and RAID 5, then SAP generally recommends adopting disk technology according to the predominant access mode for the data (that is, read or write intensive). See the list above in “How to Achieve High Availability”.

As an example of this principle, SAP recommends placing transaction logs (for example, Oracle redo log files or Informix dbspace LOGDBS) on mirrored disks using LVM since this data requires heavy write access. The remaining database data should be placed on disks using RAID 5 technology, assuming the write access to this data is not so heavy.

It is also possible to split R/3 System data according to its access requirements but this requires some expertise in R/3 System installation tuning. For further information, see the SAP database installation guides.

Caching on Disk Devices

Disk devices can greatly benefit from caches. However, the performance benefits must be balanced against the availability implications in the event of failure, as discussed below.

Performance Benefits

Disk I/O performance can be greatly improved by utilizing caches. The access times of disks and SIMM modules (used in caches) differ by a factor of up to one million (around 10 milliseconds with disks compared to 10 nanoseconds with cache SIMMs). Write and read caches are available, as described below:

- A write cache accepts the data sent by the R/3 System very rapidly, so freeing up the CPU for new work. The slow process of actually writing data onto the disk spindle is then handled internally on the disk device.
- A read cache can serve R/3 System read requests without actually accessing the disk if the cache is large enough to allow a high hit rate, that is, identical data items are requested multiple times with high frequency.

Consider caching as a means of improving high availability disk systems

Generally large disk caches should be able to compensate the write penalty encountered with high availability disk systems. Disk devices with cache sizes up to 4 GB are currently available. However, the quantified payback of the cache increment is highly dependent on the R/3 System usage pattern.

Failure Scenarios with Caches

Particular attention needs to be paid to write caches since they hold data that has not yet been permanently stored on the disk spindle itself. These represent a single point of failure and so deserve fault tolerant measures to increase their redundancy, as follows:

- To protect against power failure, data can be retained by using cache with non-volatile SIMMs or a battery backup system for powering the SIMMs.
- Mirrored SIMMs can be used so that a redundant copy is held in case of SIMM hardware failure.

For read caches redundancy precautions are in principle far less important since data is still permanently available from the disk itself.

Bus Architecture

Bus speed is not going to be a bottleneck for performance, as long as the limiting factor lies in the actual write process of the data onto the disk spindle. However, large caches could shift the bottleneck more to the bus architecture side. Then a high-speed, state-of-the-art bus architecture helps to maximize the performance of disk data storage with high availability options. However, note that maximum performance requires that the disk device supports the bus transfer rates internally.

Exploit OS support for redundant bus architectures if available

In high availability solutions for the R/3 System, SAP recommends taking advantage of operating system support for redundant bus architectures. For example, if two separate channels (SCSI paths) exist to connect to a given disk, a suitable operating system device driver can be configured to try a second channel if the first one fails. Some LVM implementations support this high availability feature.

The following sections discuss SCSI bus technology and fiber channel (FCS) technology with respect to high availability. There are additional vendor-specific bus architectures with high performance, which might be suitable for high availability purposes. However, note that standardized equipment has clear benefits in terms of general compatibility. Consult your hardware vendor for detailed advice about the most suitable bus architecture.

SCSI

SCSI stands for “Small Computers System Interface”. Key factors for SCSI bus architectures include the following:

- SCSI-1 versus SCSI-2 (both ANSI-standards)
- Single-ended versus differential bus
- Width of data bus connection (8, 16 or 32 bit)
- Speed of data bus connection (5, 10, 20, 40 MB/sec on the specified bus width)

Note that the distinction between single-ended and differential SCSI bus is made at the electrical level. This implies that, although connectors might be the same, these systems are not compatible and can harm each other electrically when connected without conversion facilities.

Generally SAP recommends using the bus technology that offers you the highest robustness, bus width/speed, and device connectivity. Therefore, differential bus SCSI-2 at largest widths and highest speed are the option of choice. Differential busses offer the benefit of improved robustness and less noise above SCSI-1 systems. Generally SCSI-2 can also be run at highest transfer speeds.

Note the following:

- A given bus can run at 5 MegaTransfers per second (conversion to MB/sec depends on bus width) and still claim to be SCSI-2.
- Bus transfer rates are normally only given as theoretical, maximal values that you can use to compare different architectures. In practice the realistic and sustained transfer rates are approximately 30-50% lower. Only the latter, corrected values are useful for quantitative estimation of the expected performance.
- Bus architectures also differ in their maximal bus length and number of maximally connectable devices. These are important figures when planning the expansion of your disk system for future requirements (see below).

SCSI bus termination

Large installations often use shared SCSI buses to which several SCSI disk devices can be attached. Refer to Cluster Technology [Page 130]. Attaching disks properly is a high availability issue in terms of permanent SCSI bus termination. Any SCSI bus has to be terminated properly on exactly two ends to make sure of correct bus operation. It is common practice to provide SCSI termination on board the SCSI disk device directly (on the SCSI controller). However, this means that the concerned disk becomes a single point of failure. If the disk providing one of the terminators on board has to be taken off the bus for repair or service, the SCSI bus is “open” or unterminated, leading to unavailability of all SCSI disks on the shared bus.

It is easy to avoid this problem by always terminating a SCSI bus externally, that is, by attaching all disks through Y-cables and plugging terminators directly into the cable outside any device.

Then, even if one of the disk devices on the bus has to be taken offline, the bus still remains fully operational.

Fiber Channel Standard (FCS)

FCS can supply massive performance benefits. FCS is an ANSI-standardized, high-speed interface that can span large distances (up to several km) and sustain transfer rates of up to 100 MB/sec (speed and distances are hardware-dependent). A larger number of connectable devices and support for error detection and correction at the hardware level are also provided. The benefits of this technology are maximized if a given disk can also support these high transfer speeds internally.

Capacity Options

The maximal number of devices on the bus determines how much disk capacity can be attached to your R/3 System directly. FCS can potentially support more devices than SCSI. In many cases, however, CPU and bus performance sets limits to the maximum possible number of disks.

It is important to consider capacity and growth options at an early stage when choosing new disk equipment. A system with future growth potential performs better, more reliably and is easier to expand as required. A system with its capacity fully utilized at the outset causes problems when trying to accommodate increased needs.

In this context, note that disk arrays (either JBODs or RAID systems) have the advantage of supplying a large amount of disk space on few SCSI channels, that is, the potential storage capacity is very large. In this respect it is a drawback of single standalone disks that each disk consumes one SCSI channel, restricting future growth potential (at least theoretically, since disk performance might already impose more stringent limits).

Disk Replacement Procedures

Even in a high availability disk system, failed disks must ultimately be replaced. The degree of service disruption required to finally return to normal operation varies greatly.

> 💡
>
> Retain at least one spare disk in case of failure
>
> This is a simple recommendation but can save a large amount of time if failure occurs. The disk should be ready to “plug in and go” (that is, configuration is not required). However, it is worthwhile testing this before a real failure occurs.

The following terms are used to qualify the way in which the replacement can be provided:

- Cold swap – system stop, power off
- Warm swap – system stop, power on
- Hot swap – system running, but operator action needed
- Automatic swap – system running and automatic online disk replacement

The replacement method is determined by the fundamental design of the disk device. That is, devices with lower levels of "swappability" do not generally provide upgrade options. The “hot swap” feature increases the cost of disk space by up to 50% per megabyte. This should, however, pay back in terms of reduced downtime and service costs.

The ability to do a hot swap is common with disk arrays (either JBODs or RAID systems). In general, RAID systems automatically start resynchronizing after a failed disk has been hot-swapped, without any subsequent administrative intervention. LVM working on JBODs can also take advantage of hot-swapping of failed spindles but possibly requires shutting down the application and some administrative action. For LVM-driven mirrors, it is best to keep a hot spare disk on either side of the mirror to minimize disruption in case a disk fails.

Journaled File Systems (JFS)

Journaled file systems (JFS) have been designed with failure resilience, decreased boot time after system crashes and increased online performance in mind.

Consider JFS for high availability of your R/3 System

It is recommended to use journaling file system architectures instead of normal file systems if high availability is an issue for your R/3 System, since it can considerably reduce the reboot time after failure. It also facilitates many administrative tasks and might improve runtime performance.

System failures (for example, node power failure) can lead to inconsistencies in the file system after a crash. This is why UNIX systems must check and, in case of corruption, reconstruct the file system during startup. Normally this procedure includes a full scan of the entire file system, greatly extending total downtime (especially for large systems).

The obvious advantage of JFS lies in the fact that only files that are corrupted and need to be reconstructed are addressed for checks at when the system is started. Checking the file system only requires a short time.

While accessing a file in normal system operation JFS employs a synchronous logging mechanism similar to the redo-logging employed by relational database systems. Meta-data that contain the changes to the file are written to a reserved area and applied to the real file only after a “commit”. Corrupted files can be reconstructed from the logs after a system failure has occurred.

Due to the internal architecture of JFS, many administrative tasks on the file system can be performed online, such as increasing storage space, defragmenting and backing up. JFS also often delivers higher runtime performance. These features make JFS the file system of choice for high availability requirements.

Checklist for Single Points of Failure for Disk Systems

Check the disk aspects of your R/3 System for single points of failure

In order to render your R/3 System highly available, you should aim to avoid single points of failure as far as possible. You should identify and prioritize possible points of failure within your R/3 System in terms of how likely they are, what effects failure would have and how expensive a solution providing redundancy would be. Finally, when a given solution is implemented, you should test it to ascertain how reliable it actually is.

Possible single points of failure in the hardware of a disk system include the following:

- Power supply
- Fan and cooling

- Internal/external cabling
- SCSI path from host machine to device
- SCSI bus terminators
- SCSI disk devices with internal SCSI terminators
- Internal system bus
- Write-cache:
  - Non-volatile SIMMs or battery backup serve to address power failure
  - Mirrored SIMMs to address SIMM failure
- Read-cache, non-volatile SIMMs optional
- Battery power for the device to store cache to disk in case of power failure
- Controller
- Micro code
- Disk-internal storage processors
- RAID internal storage maps
- Disk spindles
- Spindle mechanism

Possible single points of failure in the disk-based data are the following:

- R/3 System user data
- R/3 System system data
- Software components:
  - R/3 System
  - DBMS and log files
  - Operating system and swap space
- Root file system

Uninterruptible Power Supply

Use

The local electricity utility generally provides power supplies. Although this is often highly reliable, the consequences of even very occasional failure mean that you should give high priority to the problem of reliability of supply.

> Install uninterruptible power supply (UPS) as a fundamental aspect of ensuring high availability
>
> UPS is relatively cheap and you should install it early on when building high availability into your R/3 System.

Features

Consider the following to protect your system from power supply problems:

- Continuity of supply

  You can take the following measures to cope with complete failure of supply:

  - Install generator systems to provide continuous power in the event of supply failure. This is the most comprehensive solution to problems of supply continuity.
  - Install a battery backup system that can sense the failure of external power and switch to battery supplied power in the event of such a failure. Battery backup systems offer a limited time (approximately 10 to 15 minutes) during which the system can be sustained. In this scenario, a script is normally provided to shut down the system gracefully as the battery nears the end of its power. This graceful shutdown decreases recovery time, compared to a sudden system outage. If power returns prior to the shutdown of the system, the system switches to regular power and continues working without interruption.

- Quality of supply

  Additionally, you can install a power conditioning system in areas where the power supply is reasonably reliable, but might be subject to significant fluctuations in current. Variants of UPS are available to protect the hardware against various power supply problems (for example, spikes, surge, sags, noise, brownouts, blackouts, and harmonic distortion).

R/3 with ALE and Internet

Purpose

This section describes two emerging technologies with relevance for R/3 Systems from the high availability standpoint. Both enable R/3 Systems to communicate with distributed systems. ALE links distributed applications on R/3 Systems while the SAP Internet Transaction Server (ITS) facilitates communication with Internet-based applications.

Implementation Considerations

For detailed technical guidance when implementing a specific product or feature, contact the appropriate source, such as your SAP consultant.

Integration

High availability for the R/3 System should be part of a system-wide strategy. Therefore, you should also consider other components of the system, such as the R/3 System itself, the database management system (DBMS), the network, hardware and system software, and so on.

R/3 and ALE

Use

The ALE feature available from R/3 Release 3.0 supports the construction and operation of distributed applications. ALE includes a business-controlled message exchange with data consistency supported across loosely coupled SAP applications, as shown in the example at the end of this section.

The application integration occurs by means of synchronous and asynchronous communications rather than by means of a central database.

The basic concept of ALE is to guarantee distributed yet integrated R/3 operation. The applications run independently with their own set of data in the distributed environments, as shown in the example.

Features

The key issue for high availability in an ALE scenario is to make sure that the distributed R/3 System applications can recover gracefully from a failure in the system.

Independent systems imply data redundancy. Data must therefore be distributed and synchronized across the entire system. The basis for this data exchange is the "Intermediate Document" (IDoc) used in the EDI interface. The IDoc is the data container for the exchange between distributed systems, either between SAP systems or between SAP systems and other types of system. Data is passed on immediately after update by the application or by a batch job.

There are the following types of communication used in the ALE environment:

- Asynchronous

  The asynchronous communication used for the data exchange means that the application does not have to wait for a response from the target system. If, for example, the connection fails just before an update, the update can be sent later, controlled independently by the communication layer.

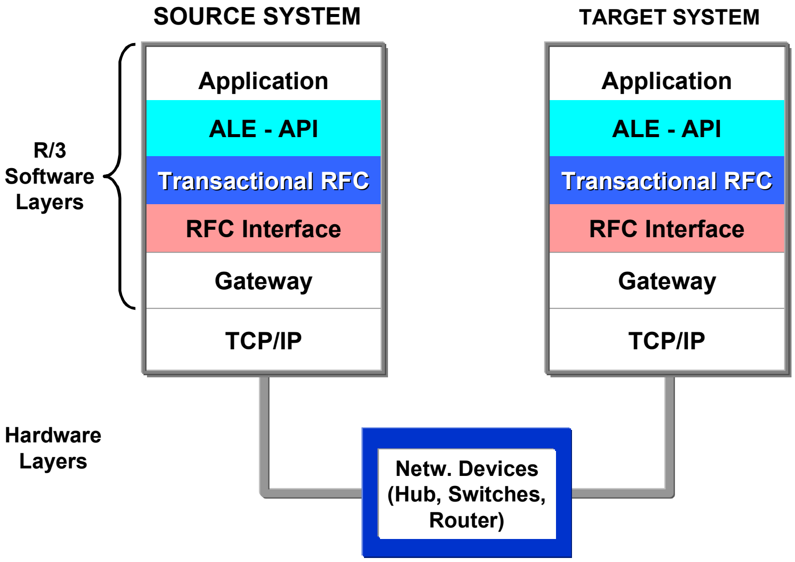
  The failed target system clearly can not receive any messages. Instead, messages destined for the failed target system are stored in queues on the sending system, until the target system is once more able to receive messages.

- Synchronous

  Synchronous communication is only used for information retrieval. In this case, there is no data distribution with update procedures. The synchronous request is performed via a direct function call to the target system. Such data is not exchanged via IDocs.

The diagram below shows two ALE-linked R/3 Systems. It illustrates how the ALE application programming interface (API) and the remote function call (RFC) software fits into the other components of the R/3 System:

ALE: Hardware and Software Layers for High Availability



The upper software layers – application and ALE - API – are protected from disturbances at lower levels of the software hierarchy and similarly from failures in the hardware layers. For information about high availability at the network level, see Network System Key Issues [Page 108].

Activities

The following table shows what happens when a failure occurs in either the source or target systems in the above diagram:

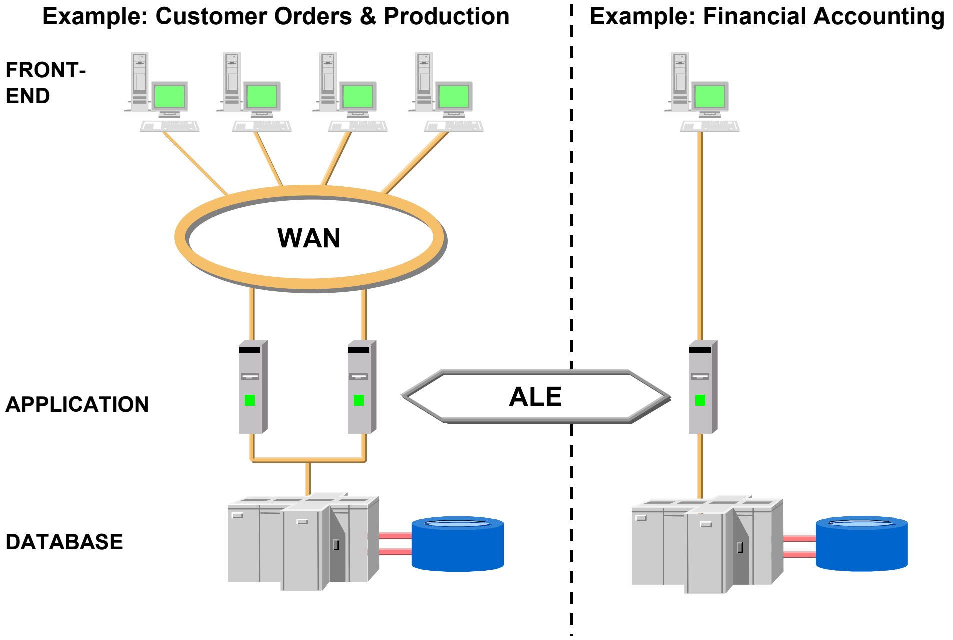
Handling of Failures in ALE-linked R/3 Systems at the RFC Level

Failure	Handling
Connection between ALE source and target system	Transactional RFC layer recovers by starting a batch job.
ALE source system application host	Client can log on again and enter new transaction. Refer to “Logon Load Balancing” in R/3 System Failures [Page 27].
ALE target system application host	RFC logon load balancing
Gateway failure	Restarted by dispatcher. See “R/3 Service Failures” in R/3 System Failures [Page 27].

To find out how to make sure of high availability for other components of ALE-linked R/3 Systems, see the relevant section of this documentation.

Example

Example of ALE Scenario


Example: Customer Orders & Production
Example: Financial Accounting
FRONT-END
WAN
APPLICATION
ALE
DATABASE

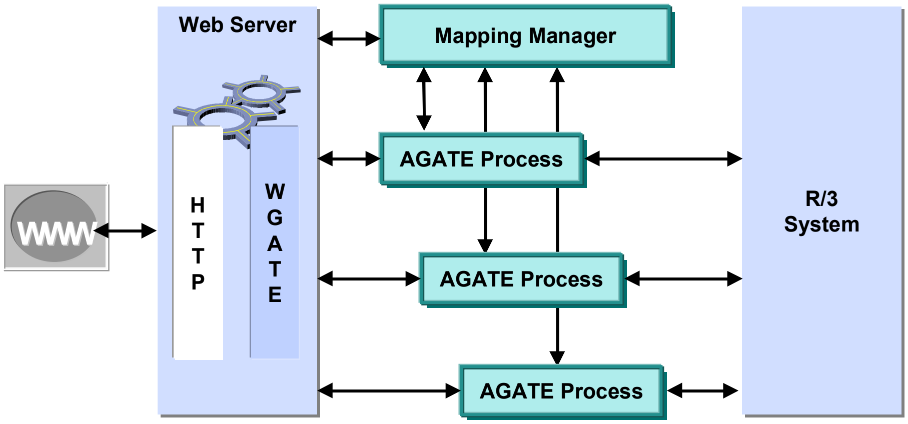
R/3 with Internet Applications

Use

You can implement Internet Application Components (IAC) in the flexible R/3 System architecture very rapidly. The application and processing logic of the IACs is implemented using the standard ABAP development environment. This means that the Internet applications run within the R/3 System. To link the R/3 System to the Internet, an additional software component known as the SAP Internet Transaction Server (ITS) is required, and this is shown shaded in the diagram below:

Integration

R/3 System to Internet Connection: SAP Internet Transaction Server (ITS)



Features

The ITS consists of the following components:

- WGATE

  This is a gateway program that guarantees the integrity of the data flow between the HTTP server and the AGATE component via TCP/IP. The WGATE modularizes the interface to the HTTP server and is implemented as an exchangeable DLL.

- AGATE

  This is a gateway program that guarantees the connection to the R/3 System via the DIAG interface and the GUIRFC interface as follows:

  – Incoming data from the WGATE is converted into a DIAG datastream, which is then passed to the R/3 System using the SAPGUI protocol.

  – Output data from the R/3 System is dynamically converted into HTML documents, which are then passed to the WGATE.

  – Additionally, data can be exchanged between AGATE and R/3 via the GUIRFC Interface.

  The AGATE is implemented as an executable file and uses multithreading to process multiple requests.

- Mapping Manager

  The mapping manager is implemented as a Microsoft Windows NT service and is therefore administered via the NT service manager. For each new session, the mapping manager assigns an AGATE process to the WGATE. After the TCP/IP channel between the WGATE and the AGATE has been established, these communicate directly with one another. The mapping manager supervises the AGATE processes, checking on workload and current status. To enable this supervision, a permanent TCP/IP connection between the mapping manager and the AGATE processes is established.

Activities

The components in the above diagram (that is, WWW browser, WGATE/HTTP server, AGATE, mapping manager and R/3 System) can run on physically different hosts and so you need to consider each independently in terms of high availability:

- WWW Browser

  This component is connected to the HTTP server via a WAN connection across the Internet. Since the WWW browser can run on any host in the WAN, it cannot be localized and you need therefore give it no further consideration.

- Connection WWW Browser to HTTP Server

  This connection needs to be considered in the same way as any WAN connection. Refer to "Wide Area Networks (WAN)" in Access Network [Page 120]. Internet connections are provided by a variety of hosts (acting as routers), which are supplied by a number of Internet service providers. These service providers are responsible for providing a corresponding level of availability for the connections

- Web Server (HTTP Server and WGATE)

  Web servers currently run under Windows NT 4.0 on hosts with Intel processors. If such a host fails, the connection to the WWW browser also fails. Since the Internet user can no longer communicate with the IAC, the AGATE times out and the session is canceled (that is, the session is automatically logged out from the R/3 System). In the event of failure in the web server, the corresponding WWW service must be started again with the HTTP server manager. To make sure of high availability for the web server, you must make sure that a pre-configured standby host is ready to take over in the event of host failure. Refer to Switchover Software [Page 216].

- Mapping Manager

  The mapping manager supervises the status of all AGATE processes. It runs together with the AGATE processes on a single host machine. When the host is started, the mapping manager service is automatically started (depending on the setting in the NT service manager). The mapping manager then automatically starts the corresponding AGATE processes, depending on the settings in the NT registry (see parameter `MaxAGates`, specifying the number of AGATE processes).

  If an AGATE process fails, its current sessions are aborted. The mapping manager detects this and automatically restarts the AGATE processes. Other AGATE processes continue normally since they are not affected. If the mapping manager itself fails due to any reason (operating system, hardware or software failure), all AGATE processes also fail. You must then restart the mapping manager (either manually via the NT service manager or by re-booting the host machine). The mapping manager therefore represents a single point of failure, analogous to the message service in the R/3 System.

- AGATE

  The AGATE runs as multiple processes on a single NT host machine. It is fault tolerant in that failure of a single AGATE process does not impact the remaining AGATE processes or their associated sessions. In future, SAP is planning to allow execution of the AGATE processes on different NT hosts. This provides additional fault tolerance in the event of host failure and also contribute to scalability. The connection to the AGATE via TCP/IP using the mapping manager interface will remain in this future scenario.

- R/3 System

For general information about high availability in your R/3 System, see R/3 System Key Issues [Page 10].

Tools and Services for Downtime Management

Purpose

This section describes the tools and services provided by SAP for use with R/3 Systems. You can use them to manage your R/3 System downtime for improved availability.

Most of the tools and services described in this section are relevant for database administration. However, the Computing Center Management System (CCMS) and the Consulting Services apply to many different aspects of the R/3 System.

Implementation Considerations

For detailed technical guidance when implementing a specific product or feature, contact the appropriate source, such as your SAP consultant.

Integration

High availability for the R/3 System should be part of a system-wide strategy. Therefore, you should also consider other components of the system, such as the R/3 System itself, the database management system (DBMS), the network, the hardware and system software, and so on.

SAPDBA: Oracle

Use

SAPDBA for Oracle is an integrated database administration tool developed by SAP for managing Oracle databases for R/3 Systems under different host systems. It is particularly suitable for large databases.

The information in this section is a summary and is not intended to give you exact instructions. Always refer to Using SAPDBA [Ext.] before performing tasks with SAPDBA.

You can use SAPDBA to monitor and tune your database, even if you do not have special knowledge of the database and its tools. SAP recommends you to familiarize yourself with SAPDBA since it can be used to reduce unnecessary database downtime as follows:

- Unplanned downtime by making fault diagnosis and database recovery easier
- Planned downtime by letting you monitor the database and perform preventive maintenance at a convenient time

Integration

SAPDBA uses the following standalone programs:

- `BRBACKUP` to back up the database

- BRARCHIVE to archive offline redo log files from disk to another device
- BRRESTORE to restore files from backup or archive media

Many of the monitoring functions in SAPDBA can also be performed using the database monitor in the Computing Center Management System (CCMS) [Page 166] of the R/3 System. SAPDBA runs with the UNIX and NT operating systems.

Features

SAPDBA has the following functions relevant to high availability:

- *Tablespace administration*

  The following functions are available on the *Tablespace administration* menu:

  – *Alter tablespace Add datafile*

    Use this function to add a file at the operating system level to an existing tablespace (for example, when a tablespace has overflowed). This has to be done while the database is online. The space in the new file is immediately available for use by objects in that tablespace, for example, an extent can be allocated for a table in the new file. SAPDBA provides defaults for the location and size of the new file and both can be adjusted if desired.

  – *Freespace and fragmentation of all tablespaces*

    This gives a list of freespace available by tablespace. It can be used to indicate whether a reorganization instead of, or as well as, addition of a file to the tablespace might be necessary.

  – *Check space for objects in all tablespaces*

    This checks whether n extents can be allocated for an object in a tablespace without leading to tablespace overflow and repeats this check for all tablespaces.

  – *Check space for objects in tablespace*

    The same as the preceding function but for a single specified tablespace.

- *Reorganization*

  The following reorganization functions are available:

  – *Check extents and fragmentation*

    This function contains the following sub-functions:

    - The check for objects with at least n extents, to monitor for objects approaching the limit for the number of extents
    - The functions to validate an index to check for unused space in an index (see fragmentation)
    - Various statistics about fragmentation

  – *Alter/show storage parameters*

    You can use this function to change maxextents and next extent size settings for tables and indexes.

  – The actual reorganization functions:

These functions enable you to perform various types of reorganization. You can reorganize tables or indexes, or lists of tables or indexes, and you can also reorganize tablespaces, with or without the datafiles.

The database has been set up so that tables reside in “data” tablespaces (the name has “D” at the end) while the corresponding indexes reside in separate “index” tablespaces (the name has “I” at the end). When a table is reorganized, the table is exported and imported, and the corresponding index is also dropped and recreated. When an index is reorganized, the index is dropped and recreated but the table is not affected.

Therefore, a reorganization of a data tablespace, that is, export and import of all tables residing in a tablespace, also includes the reorganization of the corresponding index tablespace, but not vice versa.

The reorganization including data files lets you rearrange the layout of data files in a tablespace. The goal usually is to have fewer, bigger files.

During a reorganization several options can be used to determine the effects of the reorganization, for example, compress all extents of each object into one, change storage parameters manually, reduce object size (helpful after massive deletes to free up space), and so on.

- *Backup database*

  This function backs up data files, control file and, for full offline backups, online redo log files. It can be used to set parameters for database backups (online or offline, device, tablespace to be backed up, and so on) and actually execute the backup, with SAPDBA calling `BRBACKUP`. In this way, SAPDBA offers an interface to call `BRBACKUP`, but note that `BRBACKUP` can also be run on its own, that is, standalone. By default `BRBACKUP` uses parameters as defined in the profile `init<SID>.sap`. The parameters specified in the SAPDBA screen are passed to `BRBACKUP` and overwrite the settings found in the parameter file for the current run of `BRBACKUP`.

- *Backup offline redo logs*

  This archives the offline redo log files to tape. The menu is analogous to the *Backup database* function. As with the *Backup database* function, SAPDBA basically offers an interface, this time calling `BRARCHIVE` interactively (`BRARCHIVE` can also be run on its own, that is, "standalone"). `BRARCHIVE` also uses the profile `init<SID>.sap` by default. Settings in the SAPDBA session overwrite the values found in the parameter file for the current run of `BRARCHIVE`.

- *Restore/Recovery*

  This function can be used to do a reset (restore and optional manual recovery to a chosen point in time), full restore and recovery (recovery to the current point in time or a chosen point in time), partial restore and full recovery (restore of parts of the database and recovery to the current point in time, including check of database for missing files, restore of required database files and archived redo log files), and restore of all files of a tablespace or just one datafile (implying a recovery to the current point in time).

- Command line operations

  An example is the `-next` command line option to adjust the `next` parameter (specifying the next extent size) of the database object storage definitions.

Actions such as reorganization are performed with script files generated and executed when the function is called in SAPDBA. All files related to such an action are placed in a separate directory, usually created under the directory `$ORACLE_HOME/sapreorg`. The directory name is the date and time when the action was started, for example, `9511141653` for an action started on 14th November 1995 at 4.53 in the afternoon.

Files related to backup/archive of offline redo log files are usually placed in directories under `$ORACLE_HOME/sapbackup`. The directory name consists of an encoded timestamp plus codes describing the action, for example, `bcqpvusg.ant` for a backup done on 31st October 1995 at 9.26 in the evening, where `ant` means `a`=all files `n`=online `t`=tape.

BACKINT

`BACKINT` is a program to interface SAP's backup tools (that is, `BRBACKUP/BRARCHIVE/BRRESTORE`) with third-party backup tools. SAP provides a specification as to what functionality the `BACKINT` program has to provide. The third-party vendor performs the implementation. SAP's tool can then work with the third-party tools by means of `BACKINT` using calls defined by SAP.

For more information, see Using External Backup Programs [Ext.].

We make the following recommendations for using `BRBACKUP`, `BRARCHIVE`, and `BRRESTORE`:

💡

Use `BRBACKUP`, `BRARCHIVE`, and `BRRESTORE`

SAPDBA has its own tape management system for the tapes used for database backups and archived redo log files. SAPDBA's restore function (that is, `BRRESTORE`) only works if SAPDBA has also been used to perform the database backup and archive the offline redo log files (that is, using `BRBACKUP` and `BRARCHIVE`).

💡

Use `BRRESTORE` if possible

SAPDBA uses `BRRESTORE` in recovery situations and you can use `BRRESTORE` if you wish to perform recovery without SAPDBA.

SAP recommends you use `BRRESTORE` since this makes the process of restoring files much easier.

💡

Use third-party tools if necessary to reduce time taken

If it takes too long to perform a backup or restore, you might need to use `BRBACKUP`, `BRARCHIVE`, and `BRRESTORE` together with third-party tools. The interface is described in the section above called `BACKINT`.

See also:

Database Administration (Oracle) with SAPDBA [Ext.]

SAP DBA

Documentation in SAPNet

Oracle documentation

SAPDBA: Informix

Use

SAPDBA for Informix is an integrated database administration tool developed by SAP for managing Informix databases for R/3 Systems under different host systems. It is particularly suitable for large databases.

> The information in this section is a summary and is not intended to give you exact instructions. For more information before you start using SAPDBA, see Getting Started with SAPDBA [Ext.].

You can use SAPDBA to monitor and tune your database, even if you do not have special knowledge of the database and its tools. SAP recommends you to familiarize yourself with SAPDBA since it can be used to reduce unnecessary database downtime as follows:

- Unplanned downtime by making fault diagnosis and database recovery easier
- Planned downtime by letting you monitor the database and perform preventive maintenance at a convenient time

Integration

SAPDBA is a standalone tool – that is, it works independently of the R/3 System – since many database administration tasks require the database to be in offline mode. Tools that depend on the R/3 System, such as the Computing Center Management System (CCMS) [Page 166], cannot operate when the database is offline. Many of the monitoring functions in SAPDBA are also available in the CCMS. SAPDBA runs with the UNIX and NT operating systems.

SAPDBA normally generates script files that then actually execute actions against the database. The results of the actions are recorded in log files for subsequent examination if problems arise. The script files are often restartable if an action needs to be repeated.

Features

SAPDBA filters information about the database, showing you only what you need and uses complex database statements to let you confidently manipulate the data with security and integrity. It offers the following features:

- Server Mode [Ext.]

  This offers you a safe way to change the database server mode. If you attempt to bring the server offline while the R/3 System is running, you get a warning message.

- Dbspaces [Ext.]

  SAPDBA offers you the following analysis and change facilities for dbspaces:

  – Analysis

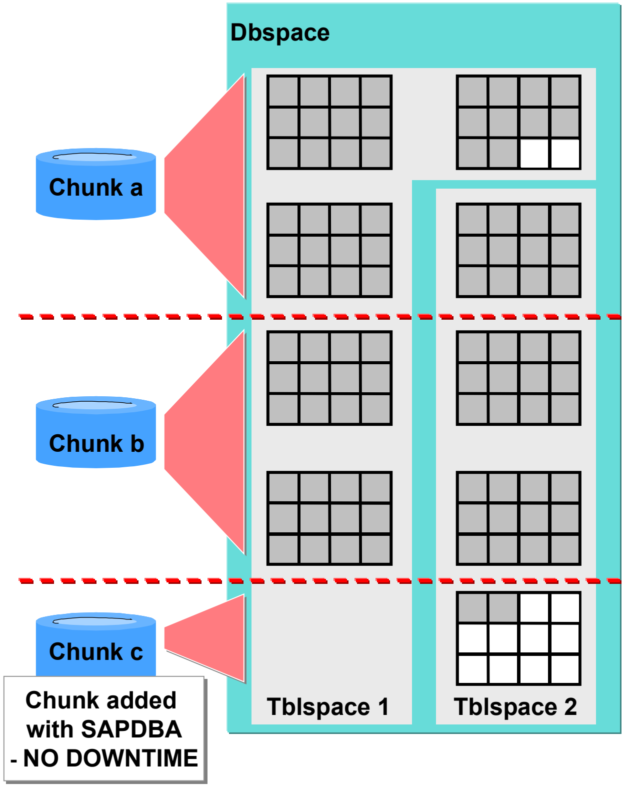
SAPDBA: Informix

The reporting options allow you to gather information about dbspaces, chunks and devices. You can easily see, for example, how full a dbspace is or how much free space is left in it. This helps you judge when to extend the dbspace.

– Change

The change options allow you to easily create, extend, and add dbspaces. Knowledge of the underlying raw device layout is not needed since SAPDBA gives you a menu of available free storage areas. The following diagram illustrates this feature:

SAPDBA with Informix: Adding a Chunk to Extend a Dbspace



Prevent dbspaces from filling up

If a dbspace fills up, the Informix server refuses to accept any more inserts to tables in the affected dbspace. You must extend the dbspace as soon as possible (that is, “add a chunk”).

- Monitor dbspaces

  To identify dbspaces that need extending, see Listing Dbspaces with SAPDBA [Ext.].

  To identify dbspaces that are restricting table growth and that therefore need to be extended soon, see Analyzing Tables for Critical Next Extent Size with SAPDBA [Ext.]. Tables appearing on this report could not be properly extended due to a lack of contiguous space in their dbspace. In this case, the Informix server allocates the largest available portion of space to extend the table with the result that the table can become scattered through the dbspace.

- Extend dbspaces if necessary (see graphic above)

  To extend a dbspace, see Adding a Chunk with SAPDBA [Ext.]. This procedure is fully automated and is much easier than extending the dbspace manually.

- Reorganization [Ext.]

  SAPDBA offers you the following analysis and change facilities for reorganizing database objects:

  – Analysis [Ext.]

    You can use the analysis options to analyze tables and indexes by a variety of parameters (fill level, size, number of extents, number of extents still available, and so on). This helps you monitor database objects to make sure they do not run out of storage space.

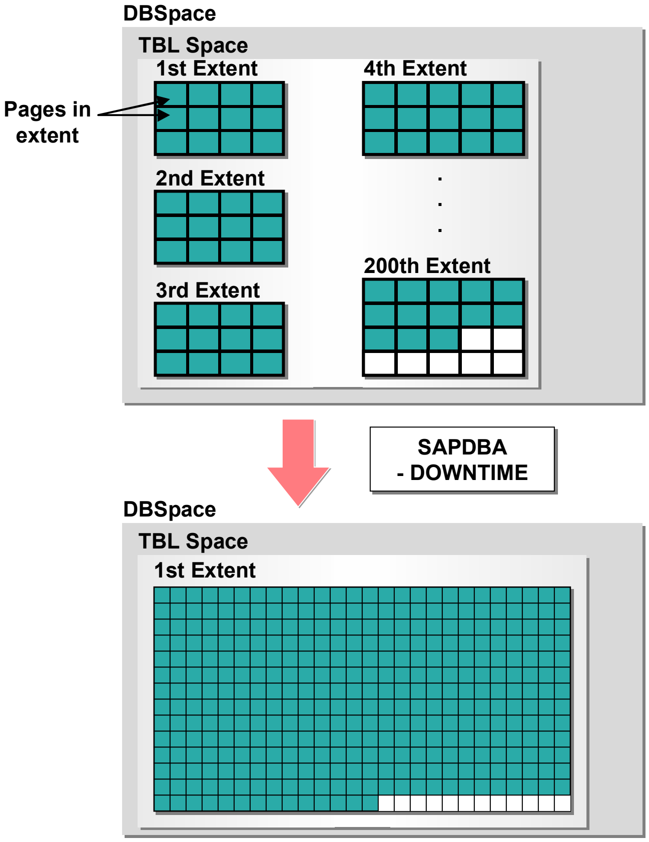
  – Change [Ext.]

    You can use the change options to reorganize a table or index with different storage parameters (for example, a larger extent size, in a different dbspace, and so on). Since the Informix server automatically doubles the extent size as a table expands, you should only perform a reorganization in exceptional circumstances, that is, when the number of extents is very high and the table is still expanding rapidly.

  With these facilities, you could, for example, find all tables with only a small number of extents still available and then reorganize a table before it runs out of extents. The following diagram illustrates how you can use SAPDBA to reorganize a table (in this example, to rebuild the table with a larger initial extent):

SAPDBA with Informix: Reorganize Table

SAPDBA: Informix



Prevent tables from filling up

Although the Informix server extends tables as necessary, there is a limit to how many extents each table can have. Refer to Managing Extents (Informix) [Page 72]. If a table completely fills up, the Informix server cannot accept any more inserts to the table and you have an error situation that might cause applications to fail.

- Monitor tables

  To find the number of extents still available for a table, see the report showing the number of extents left in Analyzing Tables by Fill Level, Size, and Extents with SAPDBA [Ext.]. If a table only has a few extents left, you must reorganize it.

- Reorganize a table

  To reorganize a table with different storage parameters, see Reorganizing a Single Table with SAPDBA [Ext.]. For example, if the table has become too big, you can rebuild it with a larger initial extent and larger next extents.

- Data consistency [Ext.]

  If database data is not physically consistent when you perform an archive or backup, you might not be able to use the archive or backup to restore the database. Therefore, check data consistency regularly.

You can plan a regular consistency check in the DBA Planning Calendar of the CCMS of the R/3 System. Refer to Checking Physical Consistency in the DBA Planning Calendar (Informix) [Ext.].

- Database System Checks [Ext.]

  These checks help you make sure that the database is running optimally, so reducing unnecessary downtime. For example, you can check the parameters in the `ONCONFIG` file, the chunk and disk layout, and so on.

  You can plan a regular system check in the DBA Planning Calendar of the CCMS of the R/3 System. Refer to Checking the DB System in the DBA Planning Calendar (Informix) [Ext.].

- Recovery Report [Ext.]

  When the report is correctly installed, this option enables you to list the recovery report(s) that have previously been automatically created. A menu of available reports is presented and you can choose the most up-to-date report. The report shows you the tapes you need to use and the sequence they need to be used for the restore. If the tapes on the most recent report are missing, damaged, or unavailable, you can look at previous reports until you find one that lists available tapes for the restore.

  The report is **only** available for `ON-Archive`, not `ON-Bar` or `ontape`.

  Install the recovery report (UNIX only)

  SAPDBA enables you to create a report that is extremely useful if your database needs to be restored. Make sure that the report is correctly installed. If so, the report is automatically created after archives and logical log backups.

  Use the recovery report (UNIX only) if you need to restore the database

  In the event of a database failure in which data is lost, you should use the recovery report to make restoring your data easier, so keeping downtime to a minimum.

See also:

BC R/3 Database Guide: Informix [Ext.] [Ext.]

Documentation in SAPNet, using the alias “dbainf”

Informix documentation

Database Manager (DBMGUI): SAP DB

Use

The Database Manager (DBMGUI) is the database administration tool designed specifically for the SAP DB database from version 7.2. This section describes the Database Manager (DBMGUI) from the high availability viewpoint and recommends how you can use it to increase the availability of your SAP DB database.

The information in this section is a summary and is not intended to give you exact instructions. Always consult the SAP DB documentation before performing tasks with the Database Manager (DBMGUI).

The information in this section is based on **SAP DB Version 7.2**.

Features

- Monitoring data and log storage

  You can use the main screen of the Database Manager (DBMGUI) to see how full the database is. The usage levels for both data and logs are graphically displayed. You should use this information to schedule manual log backups if necessary to avoid downtime. A value of 100% in either case stops the database from processing, so preventing R/3 applications from continuing.

- Server mode

  The *State* field on the main menu screen shows the current serverdb operational mode. To perform certain administrative tasks for high availability, you need to check which mode the serverdb is in.

- Background jobs

  The display area in the center of the main menu screen shows whether a backup, a verify devspaces, or an update statistics is active in the background.

- Condition of log segments

  In SAP DB Version 7.2, you can enable *AutoLog* to automatically back up full log segments. Use the *Details* view of the list of registered databases to check whether the function is disabled or enabled.

- Configuration of SAP DB database

  The *Configuration* option on the main menu screen enables you to adjust the SAP DB database as described below:

  – Database Parameters

    You can use *Instance* → *Configuration* → *Parameters* → *Edit* to display and modify the configuration of the database parameters.

  – Add Devspace

    You can use *Instance* → *Configuration* → *Data Devspace/Log Devspace* to expand the serverdb by the specified physical storage area.

  – Update system tables

    You can use *Instance* → *Configuration* → *Upgrade System Tables* to upgrade the system tables.

Activities

You can use the Database Manager (DBMGUI) to control and monitor the SAP DB database system and to execute backup and recovery procedures. All monitoring functions available in the Database Manager (DBMGUI) can be performed using the database monitor in the Computing

Center Management System (CCMS) [Page 166] of the R/3 System. You can schedule backups in operational mode WARM with the DBA Planning Calendar, part of CCMS.

Backup

The *Backup* option on the main Database Manager (DBMGUI) screen allows you to perform complete data backups (using *Instance* → *Backup* → *Complete*), incremental backups (that is, a backup of only the modifications made since the last data backup using *Instance* → *Backup* → *Incremental*), or backups of the log (using *Instance* → *Backup* → *Log* or switching on the automatic log backup via *Instance* → *Backup* → *AutoLog on/off*). The backup or restore operations apply to only the addressed server database (that is, to serverdb).

You can use *Instance* → *Backup* → *Log* to create a backup of the log (subsequently the log is released for reuse, independently of the database operational mode, COLD or WARM). With *Instance* → *Backup* → *AutoLog on/off*, you can have each log segment automatically saved as soon as it fills up. After the log backup successfully completes, the log segment is marked as free.

Automatic log backups are performed using pre-defined parameters whereas ad-hoc data or log backups are performed interactively. Incorrect entries or a timeout due to delayed input can cause the backup to abort, possibly leading to downtime.

Each backup is done to a backup medium that can be selected from the Media Manager. After you select the corresponding type of backup in the Database Manager (DBMGUI) menu, you can select or define the backup medium. The Media Manager supports parallel backups, allowing faster throughput in the case of backups.

Incremental backups are advisable in certain situations

Incremental backups (that is, using *Instance* → *Backup* → *Incremental*) are most effective when the database modifications focus on particular database objects. Incremental backups generally increase the data volume of the backup but speed up the recovery times because only database pages have to be restored. A recovery using log backup always re-executes commands and this is more complex and time-consuming.

Back up the log frequently in WARM mode

If the log fills up (that is, to 100%), the database automatically stops processing, leading to downtime for your R/3 applications. Therefore, back up the log regularly.

The best approach is to back up log segments automatically as soon as they are full. To do this, choose *Instance* → *Backup* → *AutoLog on/off* and set *AutoLog on*.

Accelerate backup and recovery using parallel backup devices

To accelerate the backup and recovery operations for large database systems (that is, operations *Instance* → *Backup* → *Complete, Instance* → *Backup* → *Incremental* or *Instance* → *Recovery* → *Database*), use several backup devices working simultaneously in parallel. The backup and recovery times then depend on the number of backup devices used and on the disk capacity of the largest data devspace.

See also:

SAP DB documentation

DB2CC Tools for DB2 UDB

Use

You can use the DB2 Control Center (DB2CC) tools to administer the DB2 database manager running on UNIX and Windows NT platforms. The R/3 System does not need to be running. The DB2CC tools are always available and are very helpful for database recovery activities.

Integration

DB2CC contains several database administration tools, including Alert Center, Command Center, Script Center, and Information Center. The Information Center features online documentation (you need a browser to take full advantage of this), which also deals with HACMP. For more information about HACMP, see Replicated Standby Database for DB2/UDB [Page 197]. SAP has added functionality to the Control Center that specifically enables you to manage DB2 Universal Databases (DB2 UDB) for the R/3 System.

SAP recommends that **only** experienced database administrators and Basis consultants use the DB2 command line interface (in any case, this normally provides no advantages over DB2CC).

The information in this documentation does **not** apply to DB2 for OS/400 or DB2 for OS/390.

Activities

Use the Computing Center Management System (CCMS) [Ext.] to monitor your database. Starting with R/3 Release 4.0A, you can also schedule DB2 backups in the CCMS using the DBA Planning Calendar (DB2/CS) [Ext.].

Computing Center Management System (CCMS)

Use

This section describes the Computing Center Management System (CCMS) [Ext.] from the high availability perspective. The CCMS lets you monitor, control, and configure your R/3 System. It lets you automate many administrative tasks, so reducing the chances of error and enabling in many cases unattended, 24-hour system management from within the R/3 System. Therefore, it is well suited to maintaining high availability of your system.

Features

The CCMS provides the following functions relevant to achieving high availability:

- Comprehensive monitoring and management

  This helps to avoid downtime, since a poorly managed system is more likely to fail. With the new monitoring architecture – available since SAP Release 4.0 – you can monitor components within and external to the R/3 System. Alerts are triggered when the values reported to the monitoring architecture conflict with predefined thresholds. Alerts can be handled automatically by the system or manually by the operator.

  All information relevant to troubleshooting a particular problem is available from a single, central point. This is the major advantage of the monitoring architecture. Having an overview of the R/3 System reduces errors – possibly leading to system downtime – that might otherwise arise from using multiple tools. You can also extend the monitor and alert infrastructures for special purposes by using the standard programming interface to incorporate external tools.

- Reconfiguration during 24-hour system operation

  You can configure instances and specify operation modes to suit the varying conditions and workloads in your system. For example, during the day you can assign the majority of work processes to dialog processing, whereas at night you can assign them to background processing. Switching between, for example, daytime and nighttime processing mode does **not** cause downtime of the R/3 System.

- Profile maintenance

  For tuning the configuration of R/3 servers, the CCMS also offers the profile maintenance tool, with which you can track and check the consistency of your changes to start-up and system profiles. Therefore, you can avoid downtime due to a poorly tuned system.

- Graphical control tools

  The CCMS has easy-to-use control tools with both graphical and list-oriented displays. You can use these tools for administrative tasks such as starting or stopping servers, viewing logs, and checking user and work process activity. Fewer errors are made using modern graphical tools, so avoiding unnecessary downtime.

- System and business process statistics

  The CCMS can collect detailed statistics for tracking system activity and application-related business process activity. These statistics are useful for assessing the current workload or for predicting the future workload, so avoiding downtime due to severe performance bottlenecks or lack of space in database objects.

- Database and archiving management

  Many database administration tasks can be performed from within the R/3 System using the CCMS. For example, you can use the DBA Planning Calendar to schedule and analyze the results of database administration activities. A database monitor is provided for in-depth analysis of the database management system (DBMS). These tools help you to better manage the DBMS, so reducing downtime.

  Data archiving [Page 39] ensures that the online part of the R/3 System is small and therefore easy to manage, while still allowing access to the entire data for reporting and other purposes. The Archive Development Kit (ADK) allows you to develop your own archive programs.

In the area of database monitoring and tuning, some of the functions of the CCMS are duplicated by SAPDBA (or the equivalent database administration tool if you do not have Oracle or Informix). It is not possible to give strict guidelines for when to use SAPDBA and when to use the CCMS. The characteristics of your installation and the nature of the problem or requirement determine the best tool to use. However, in general, use the CCMS to automate routine tasks where possible and SAPDBA or another tool for other tasks (for example, when the R/3 System is down).

- Workload management

  Reliable workload management means that you can successfully coordinate large amounts of system load without running into severe performance bottlenecks, which can in some cases amount to downtime. The CCMS provides the following features to help you with this:

– R/3 background processing

  This feature allows you to run programs asynchronously with step-by-step control. Various types of scheduling are available and you can schedule your own background jobs under program control using the easy-to-use application program interface (API). A variety of features are provided to make managing even large background jobs easy.

– Load Balancing

  The available workload is balanced by the CCMS in the following areas:

  - Logon load balancing distributes interactive users across available servers at logon time.
  - Output workload balancing in the spool system distributes job output across available spool servers and pooled output devices.
  - Background processing balancing distributes background jobs across available background servers.

Activities

SAP recommends the following when you use the CCMS to reduce downtime:

- Define operation modes for 24-hour operations

  You can use the CCMS to define different operation modes (for example, a night-time and a daytime mode). Using operation modes avoids the need for system downtime to reconfigure the system.

- Monitor your system for problems

  The CCMS provides you with the opportunity to monitor many different aspects of your system. Get to know these possibilities since, apart from the performance gains, you can also benefit by analyzing trends and spotting difficulties (such as system bottlenecks) before they turn into real problems. The advantage for system availability is that you can then schedule preventive maintenance and upgrades as planned downtime rather than being forced into unplanned downtime.

- Use alerts

Use the CCMS monitoring architecture to check the health of your system regularly. The following is a list of resources that can cause or increase downtime if a red alert appears:

- Swap space (an operating system parameter)
- File system (an operating system parameter)
- Enqueue table capacity (an R/3 System parameter)
- Freespace management, including extent allocation problems (a database system parameter)
- Elapse time since last backup (a database system parameter)

- Do not adjust alert thresholds

  Predefined thresholds are set for a specific system that you are monitoring. Only adjust the values for these thresholds if you are sure that the change is necessary and will bring the required result.

- Use SAPDBA together with the CCMS for database administration

  CCMS and SAPDBA have different strengths and weaknesses. In general, SAP recommends you to use the CCMS for routine database administration, and SAPDBA to perform any additional analyses or tasks not supported by CCMS (such as altering the database structure).

- Use the DBA Planning Calendar to schedule database activities

  Depending on which DBMS you have, various routine database management activities can be run regularly from the DBA Planning Calendar, such as backups, updating the cost-based optimizer (CBO) statistics, configuration and performance checks, and so on. Of course, you must still satisfy external requirements for the activity, such as tapes and tape devices for backups. You can check the job log produced by the CCMS to see if the activity completed successfully.

See also:

Documentation in SAPNet, using the alias “systemmanagement”

GoingLive and EarlyWatch

Purpose

This section describes the GoingLive and EarlyWatch services, which help to reduce system downtime. Both are specialized consulting services with the following advantages:

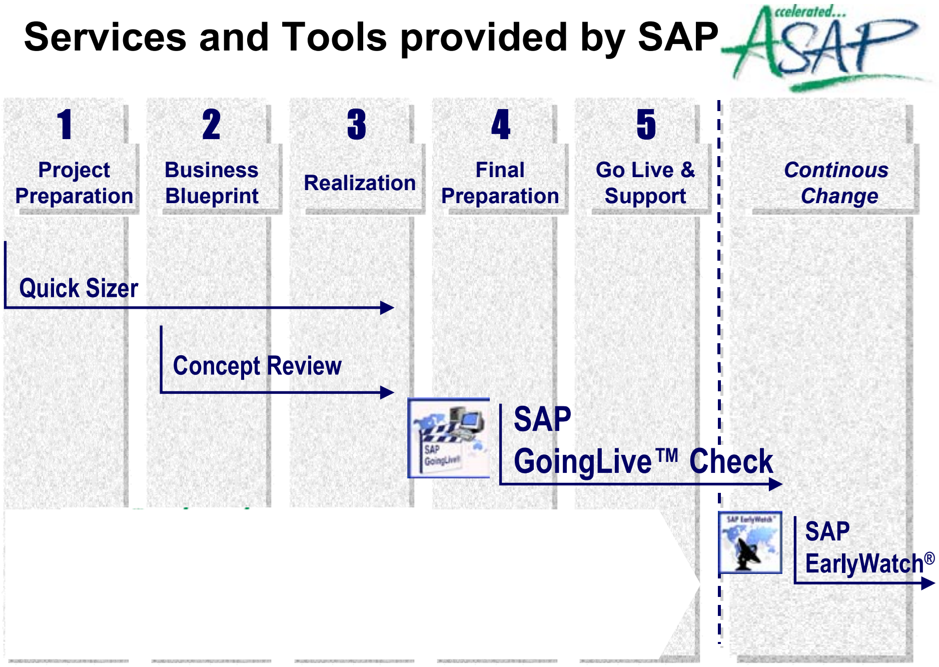
- Certified and trained SAP experts with in-depth knowledge of the R/3 System
- Best practices database with up-to-date information
- Service procedures used to maintain top service performance
- Analysis tools used to expose system weak points
- Empowering workshops to deliver expertise to your experts

GoingLive and EarlyWatch

Whereas GoingLive supports you during the implementation phase, EarlyWatch supports you doing ongoing live production. By using these services, you can make sure that your system is optimally tuned and therefore less likely to fail. The result is improved system availability.

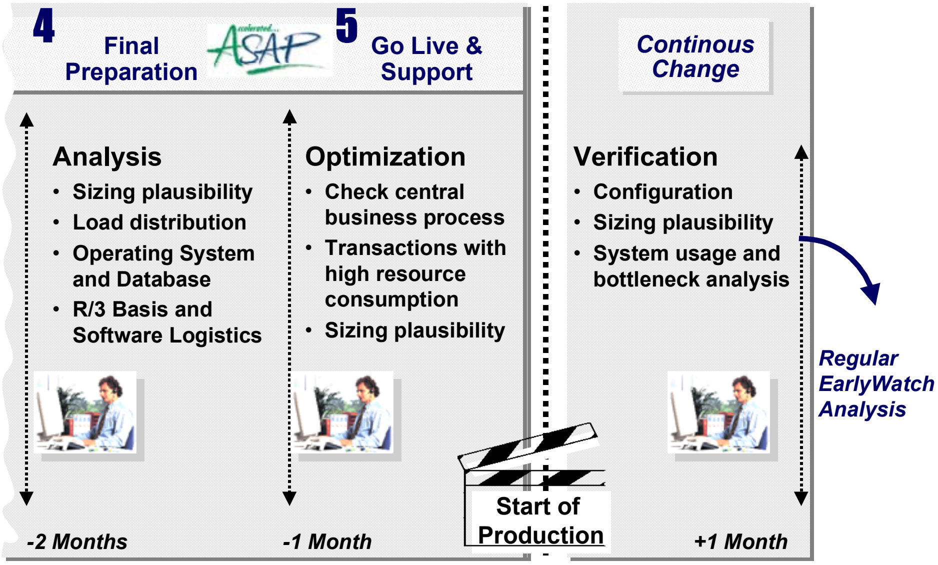
Process Flow

Both services contribute to improving the availability of your system because they stress prevention of problems before they become serious enough to cause downtime. As the following diagram shows, these services are complementary, with EarlyWatch taking over once your system has gone live:



The most critical phase in achieving high availability for your R/3 System is when you are going live, because it is more difficult to change a system once it is in production. The following diagram shows the different ways that GoingLive can help you during this critical phase and how EarlyWatch can help afterwards:

GoingLive™ Check Service Session



4 Final Preparation

5 Go Live & Support

Continous Change

Analysis

- Sizing plausibility
- Load distribution
- Operating System and Database
- R/3 Basis and Software Logistics

Optimization

- Check central business process
- Transactions with high resource consumption
- Sizing plausibility

Verification

- Configuration
- Sizing plausibility
- System usage and bottleneck analysis

Regular EarlyWatch Analysis

Start of Production

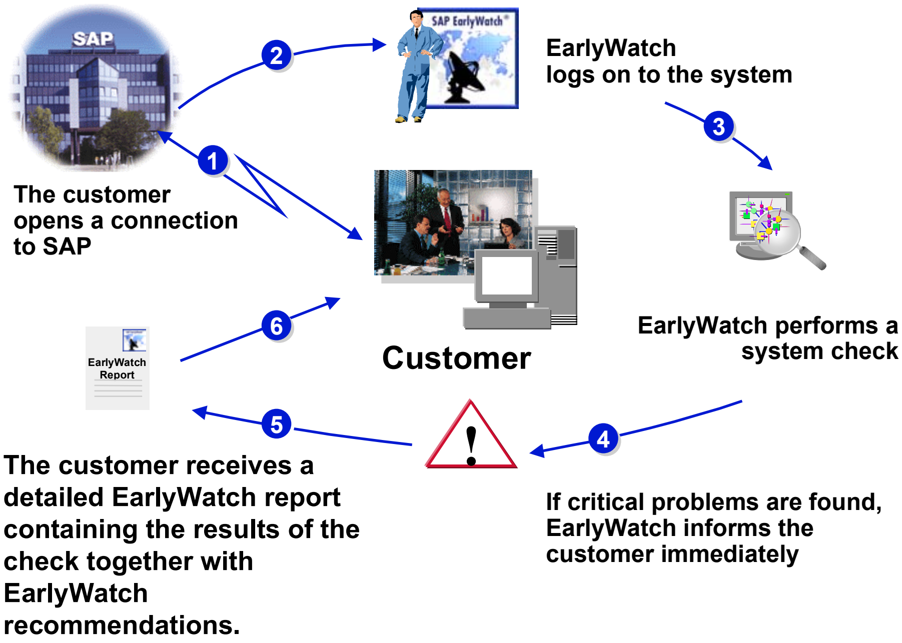
-2 Months

-1 Month

+1 Month

The EarlyWatch service works as follows:

How Does EarlyWatch Work?


SAP
SAP EarlyWatch
1
The customer opens a connection to SAP
2
EarlyWatch logs on to the system
3
EarlyWatch performs a system check
4
If critical problems are found, EarlyWatch informs the customer immediately
5
The customer receives a detailed EarlyWatch report containing the results of the check together with EarlyWatch recommendations.
6
EarlyWatch Report
Customer

EarlyWatch focuses on the following aspects:

- Server analysis
- Database analysis
- R/3 Configuration analysis
- R/3 Application analysis
- Workload analysis

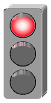
EarlyWatch Alert – a free part of your standard maintenance contract with SAP – is a preventive service designed to help you take rapid action before potential problems can lead to actual downtime. In addition to EarlyWatch Alert, you can also decide to have an EarlyWatch session for a more detailed analysis of your system.

Result

You receive a service report after an EarlyWatch session, broken down into the following sections:

The EarlyWatch Service Report

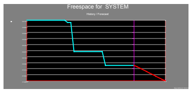
Chapter 1 Global EW Summary (red, yellow, green)



Chapter 2 Global Analysis

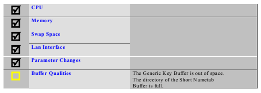
Chapter 3 Client/Server Analysis

Chapter 4 Application Server Analysis (Server 1 - N)



Chapter 5 Database Server Analysis

Chapter 6 Summary and Recommendations



☑	CPU	
☑	Memory	
☑	Swap Space	
☑	Lan Interface	
☑	Parameter Changes	
□	Buffer Qualities	The Generic Key Buffer is out of space. The directory of the Short Nametab Buffer is full.

Chapter 7 Appendix A: Checklists

Chapter 8 Appendix B: Glossary

To contact SAP about these services, see the following:

GoingLive and EarlyWatch Contacts

	GoingLive Check	SAP EarlyWatch®
Americas	**Telephone:** **800 677 7271** (from USA only) **(+1) 610 725 4545** **Fax:** **(+1) 610 725 4800**	**Telephone:** **800 677 7271** (from USA only) **(+1) 610 725 4545** **Fax:** **(+1) 610 725 4800**
Europe	**Telephone:** **(+49) 180 5 34 34 35** **Fax:** **(+49) 6227 34 4214**	**Telephone:** **(+49) 180 5 34 34 35** **Fax:** **(+49) 6227 34 4214**

See also:

Documentation in SAPNet

Special Products and Features

Purpose

This section describes non-standard products and features that you can use to improve the availability of your R/3 System. The items discussed in this section generally require more expertise and a higher level of commitment than items discussed elsewhere in the documentation. Switchover Software [Page 216] is described separately due to its technical complexity and central importance in high availability.

Implementation Considerations

For detailed technical guidance when implementing a specific product or feature, contact the appropriate source, such as your SAP consultant, your database supplier, and so on.

Integration

High availability for the R/3 System should be part of a system-wide strategy. Therefore, you should consider all components of the system, including the R/3 System itself, the database management system (DBMS), the network, the hardware and system software, and so on.

DB Reconnect

Use

DB reconnect refers to the automatic reconnect of an SAP work process to a database instance if the previous connection has been closed unexpectedly. Losing a database connection means partial loss of service on the SAP application server side. Connection problems are detected using database error codes.

Database error codes have been grouped together by SAP. The group that includes all errors related to database connection is called the RECONNECT group in this documentation.

There are the following types of DB reconnect:

- Reconnect to the same database instance [Page 175]
- Reconnect to one out of two database instances [Page 179]

  This only applies to an installation with Oracle Parallel Server (OPS) [Page 208] and Data Sharing for DB2 for OS/390 [Page 210].

The connection to the database service can fail due to various reasons, for example:

- The database was shut down.
- The database instance aborted.
- The node aborted.
- The network (TCP/IP) between application server and database server failed.

Activities

The reconnect to the same database instance is only successful if the error condition has been resolved, while the reconnect to a standby database instance is normally successful immediately (unless an error has occurred there as well).

In either case the time it takes to perform a reconnect depends on the type of failure. For the first two reasons in the above list, a request sent by the application server immediately returns one of the database errors from the reconnect group. For the last two reasons in the above list, a request sent using TCP/IP is "lost," because either the database host or the network did not respond. The time it takes to return an error to the application server depends on TCP/IP time-outs on the client side, which might take several minutes.

We are not able to provide a complete list of TCP/IP time-out parameters and how they are implemented for all hardware systems (on UNIX systems they are normally implemented as UNIX kernel parameters). The following example is for SUN.

Example

Relevant TCP/IP time-outs for SUN are as follows:

- For connect requests, parameter `tcp_ip_abort_cinterval`
- For retransmit requests, parameter `tcp_ip_abort_interval` (default is 8 minutes)

  This is a "time-out" parameter to stop retransmits of a package over an active connection if no response was received. This is usually decisive for failures in an R/3 environment for the following reason. Since most of the connections from application host to database host are active, transmitted packages that do not reach their destination are retransmitted until the time-out is reached. A RECONNECT group error is returned only after the time-out interval.
- For "keepalive," parameter `tcp_keepalive_interval` (default is 2 hours)

  This parameter specifies the period before the transmission of keepalive packages, which are sent over an idle connection to verify that the connection still exists (that is, that both partners are still functioning). Keepalive packages are sent for some time and, if none of them gets acknowledged, a RECONNECT group error is returned.

See also:

DB Reconnect Parameters [Page 180]

DB Reconnect to the Same Database Instance

Use

This section discusses how you can use DB reconnect [Page 174] to reconnect to the same database instance.

Features

The following methods can be used to reconnect to the same database instance:

- Reconnect **with** restart of the work process

  This is the method used if the R/3 internal reconnect mechanism is switched off using the profile parameter `rsdb/reco_trials`.

  A work process running with an error condition is restarted. This includes a rollback of the SAP logical unit of work (LUW) and a reconnect to the database. If the restart fails, the process terminates. You then have the following options to restart the process:

DB Reconnect to the Same Database Instance

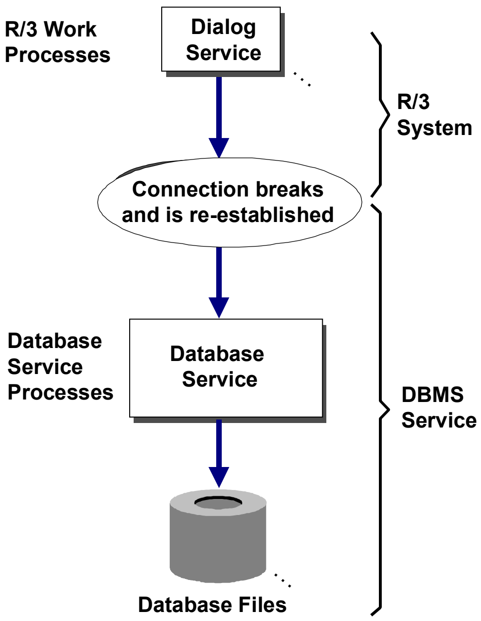
- If the application server is still running (that is, if the dispatcher, at least one dialog work process, and the SAP buffers are still available), you can use transaction `sm51` to perform a "restart after error."
- You can restart the application server. A restart of the application server means that all SAP buffers on the affected application server are lost.

This type of reconnect is not discussed further in this documentation.

- Reconnect **without** restart of the work process

The following diagram illustrates this situation (the work process shown is a dialog process but it might equally well be another kind of process):

DB Reconnect to Same Instance



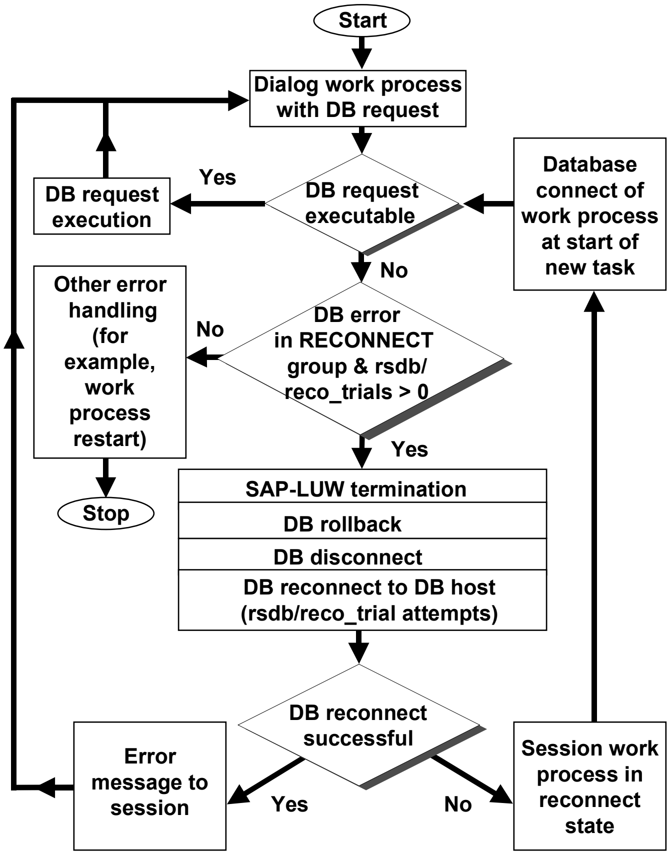
The reconnect is initiated if a request sent to the database by a work process returns one of the database errors included in the RECONNECT error group. The subsequent processing depends on the type of work process that sent the request.

The advantage of the reconnect without restart of the work process is that it does not require manual intervention. This means that you should never have to restart the whole application server. The application server buffers (that is, the table buffers) are never lost, as would happen during a full server restart. This type of reconnect is also useful if the database was shut down for maintenance or offline backups. Once the database is back up the work processes automatically reconnect.

For this reconnect feature to be active certain parameters have to be set in the application server profile. For more information, see DB Reconnect Parameters [Page 180].

The following diagram summarizes the reconnect process:

DB Reconnect: Schematic Flow


Start
Dialog work process with DB request
DB request executable
Yes
DB request execution
Database connect of work process at start of new task
No
DB error in RECONNECT group & rsdb/ reco_trials > 0
No
Other error handling (for example, work process restart)
Stop
Yes
SAP-LUW termination
DB rollback
DB disconnect
DB reconnect to DB host (rsdb/reco_trial attempts)
DB reconnect successful
Yes
No
Error message to session
Session work process in reconnect state

Activities

A work process with a connection problem is changed to “reconnect state,” the current SAP LUW is rolled back and the process tries to reconnect. The next steps depend on the type of work process:

- Dialog work process
  - Foreground request

    The scenario is that user activity generated a synchronous database request (that is, select, insert, modify, delete) that was not successful. The read is rolled back and the dialog process tries to reconnect as follows:

    - Successful reconnect

      The user receives an “ABAP run-time error” or an error message in the status line of the session screen. The user has to acknowledge the error message. The transaction (LUW) is rolled back and the session returns to the screen from which the aborted transaction was started. The user can now re-run the transaction.

    - Unsuccessful reconnect

      The user’s session screen disappears and a dialog box appears with the message that this session has lost its database connection. The dialog box disappears once it has been acknowledged.

The dialog process that handles the user session screens stays in “reconnect state”. Every time a request is passed to the dialog process requiring a database access, it tries to reconnect to the database (for example, if the user repeats the transaction that generated the database request in the first place).

From the dispatcher’s point of view, dialog work processes in reconnect state are no different from other dialog work processes. If a frontend issues a request, the dispatcher might route the request to a dialog work process in reconnect state. This request then causes the next reconnect attempt.

- Update request with asynchronous update

  The scenario is that changes have been posted to the database. Asynchronous updates are written by the dialog process to an update queue. An update process reads it from there and applies the changes to the actual database tables. The update queue is implemented as a cluster table (that is, `VBLOG`) in R/3 Release 2.x, and as three transparent tables (that is, `VBHDR`, `VBMOD` and `VBDATA`) in R/3 Release 3.x and 4.x. We use the term `VBLOG` in the reminder of this section for either implementation of the update queue.

  If the user transaction consists of a sequence of several screens, several data records will have been written to `VBLOG`. At commit time, a header record is written to `VBLOG` to complete the information for this change. The set of individual update entries and the single header entry forms one complete update record. The update process applies complete update records only. If a connection error occurred, update entries might have been written, but some update entries or the header entry are still missing. The dialog work process tries to write the missing entries but is not successful. The dialog work process then attempts to reconnect as follows:

  - Successful reconnect

    Processing is the same as described above for “Foreground Request – Successful Reconnect.” That is, the user has to re-run the transaction that was aborted and enter all the data again.

  - Unsuccessful reconnect

    Processing is the same as described above for “Foreground Request – Unsuccessful Reconnect.”

- Update work process

  - Read request

    The scenario is that an update process wants to read an update record out of the VBLOG, but the read fails due to a database error. The update work process then attempts to reconnect as follows:

    - Successful reconnect

      The update process tries to read the record again. If other update processes are still connected they might also try to read the record and apply it.

    - Unsuccessful reconnect

      The update process continuously tries to reconnect and apply records that have not been processed yet.

  - Change request

The scenario is that an update process has read an update record and wants to apply it to the actual database tables. The actual update cannot be executed due to a lost connection. The status of this update is “terminated” and the enqueue is released.

If another update process is available, that process is posted to execute the error handling of the terminated update. This process sends an express mail to the user with the message that the update did not complete successfully.

If no other update process is available, the error handling is put in the request queue of the “original” update process.

The next step is the same for a successful as for an unsuccessful reconnect. The update is not repeated automatically and the locks are released. The user has to execute the terminated update again using transaction `sm13`, if the application context allows to “redo” an update later. If the application context does not allow simply executing an update again, the user has to manually enter the data again to generate new entries in `VBLOG`.

– Error handling

The scenario is that an update process wants to do the error handling for a terminated update (that is, send an express mail to the user), but this cannot be done due to a lost database connection. The subsequent processing is independent of whether the reconnect is successful or unsuccessful.

No express mail is sent. The terminated update is shown in transaction `sm13`.

- Batch work process

  Independent of the outcome of the reconnect, the batch job is terminated. The user then has to restart the batch job, assuming that the application context allows for this.

- Spool work process

  The spool work process handles user requests to print data. The location of the data to be printed is configurable. It can reside in a table in the database or in external files at the operating system level. In both cases, access to the database is required to process the data.

  If a database request for a spool work process fails (that is, either the read of data to be printed or the write of spool administration data), then the spool request is aborted and the user has to start the spool request again. However, the data to be printed is not lost. The spool work process periodically checks for open spool requests.

DB Reconnect to an Available Database Instance

Use

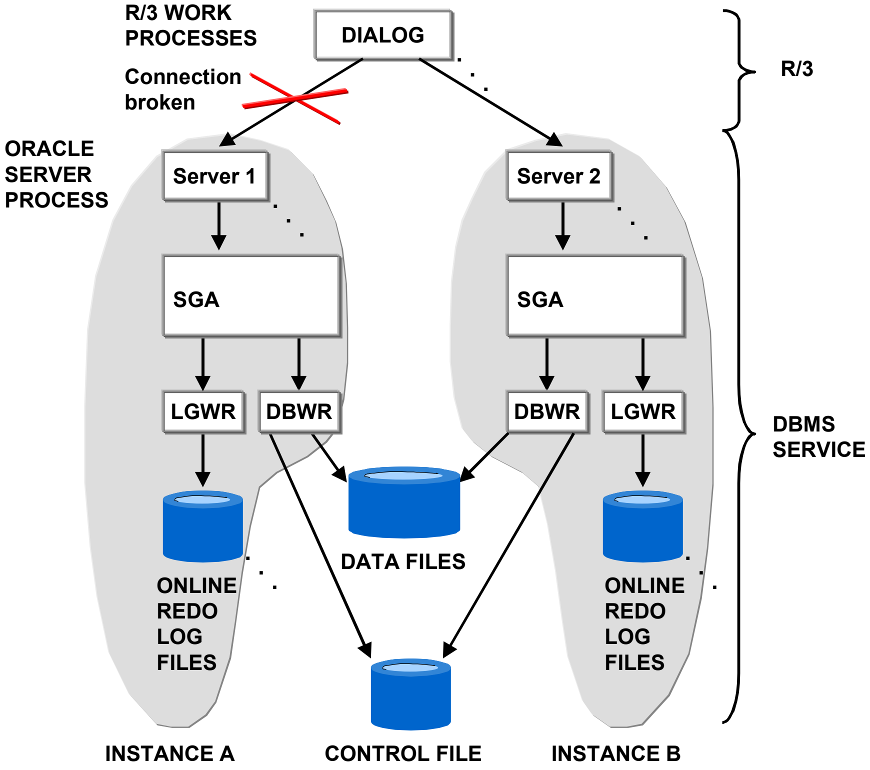
This section discusses how you can use DB reconnect [Page 174] to reconnect to the same database instance.

Features

This setup uses parallel database technology, where application hosts are connected to one database instance with a second database instance on another host available as a standby instance. The following diagram illustrates this situation:

DB Reconnect Parameters

DB Reconnect to Available Instance: Example Using Oracle



If a work process loses connection to the primary database instance, it invokes a test function to determine whether a connection to the primary database instance is still possible. If the test was successful the work process connects to the primary instance again. If the test was not successful, the work process tests the standby instance and, if successful, connects to it.

For more information about parameter settings, see “Parameters for Reconnect to Available Instance” in DB Reconnect Parameters [Page 180].

DB Reconnect Parameters

Definition

The DB reconnect parameters control the DB reconnect [Page 174] feature (without restarting the work process). For more information, see SAP Note 24806.

Structure

Standard Reconnect Parameters

- `rsdb/reco_trials`

  The default setting is 3 (with R/3 release 3.0D or later).

  The effects of this parameter are as follows:

  - `rsdb/reco_trials = 0`

    Reconnect without restart is disabled, a restart of the work process is executed.

  - `rsdb/reco_trials` = `n` (where `n` is greater than 0)

    Reconnect without restart of the work process is enabled.

    The number "n" refers to the number of times the reconnect is attempted before the session screen disappears and the dialog box (requesting the user to create a new session screen) appears.

- `rsdb/reco_sleep_time`

  This parameter describes the idle interval between each of the n reconnect attempts specified by `rsdb/reco_trials`. The default setting is 5 (the parameter can be set to a value of 0 or greater).

- `rsdb/reco_sync_all_server` – for R/3 Release 3.0C or later.

  Set this parameter to synchronize the reconnect across multiple application servers. It is relevant if you are using a parallel database system or switchover software for your database services. If an application server executes a DB reconnect, all other application servers in the system are informed and their work processes then execute DB reconnects. For more information, see Switchover Software [Page 216].

  To be sure of this functionality, you must set the parameter as below:

  ```
  rsdb/reco_sync_all_server = ON
  ```

To enable reconnect without restart of the work process, set `rsdb/reco_trials` to a value greater than 0. The parameter `rsdb/reco_sleep_time` is optional. That is, to use this type of reconnect during, for example, offline backups, you only have to set `rsdb/reco_trials`.

Parameters for Reconnect to Available Instance

For the reconnect to work, you must first set the standard parameters as described above. Then you must set parameters specific to the "reconnect to an available instance" as described here For more information, see SAP Note 24874. The following description is based on Oracle Parallel Server [Page 208], but there are similar parameters in Data Sharing for DB2 for OS/390 [Page 210].

- The following parameters are standard SAP parameters to identify the main database instance. They are **not** specific to any of the reconnect functions. These will have been set as follows before normal productive operations commence:

– `rsdb/oracle_sid`.	Default is `SAPSYSTEMNAME`.
– `rsdb/oracle_host`.	Default is `SAPDBHOST`.
– `dbs/ora/tnsname`.	Default is DB Host Net V2 server name.

- Parameters to identify the standby instance are as follows:

  - `rsdb/oracle_sid_standby`
  - `rsdb/oracle_host_standby`
  - `dbs/ora/tnsname_standby`

To use the reconnect feature, parameters `rsdb/oracle_sid_standby` and `rsdb/oracle_host_standby` have to be set in the SAP profile.

- Symmetric reconnect parameter – for R/3 release 3.0B or later – is as follows:

  ```
  rsdb/reco_symmetric = ON
  ```

  This feature enables automatic switch-back in cases where the primary database instance becomes available again and the standby database instance goes down. Therefore, a symmetric solution is available, enabling switching between the database instances.

- Parameters to control the test function are as follows:
  - `rsdb/reco_ping_cmd` used to activate a ping test to the database servers
  - `rsdb/reco_tcp_service` used to activate a tcp test to the database servers
  - `rsdb/reco_tcp_timeout` determines the timeout of the tcp test

- Optional parameter

  The parameter `rsdb/reco_sync_all_server` can optionally be set to enable reconnect synchronization across multiple application servers.

Disaster Recovery

Purpose

A disaster is a situation in which critical components in the R/3 environment become unavailable so that service cannot be resumed in a short period (less than a day as a general rule). A typical situation would be destruction of the hardware due to fire. The critical R/3 System components are the database and the R/3 application host instance that runs the enqueue and message services.

Process Flow

You consider the following steps to protect your installation:

- To protect R/3 application host running enqueue and message services

  This can be achieved by having a standby system available (at a remote site) that can be started up in the event of a disaster. If a standby system – which could also be another application server that has been reconfigured – is started to provide the critical R/3 services, all other application servers have to be restarted.

- To protect the database

  The entire database can be replicated but you have to use a method provided by the various database vendors. This approach is known as “Hot Site Backup” or “Standby Database [Page 186].” The products mentioned below follow the concept of “replication transparency”. This means that the functionality to achieve replication is built into the database service instead of having to be coded by the client applications.

Replicated Databases

Use

This section discusses ways of achieving high availability by replicating the data itself. We discuss database replication issues and the replicated database products available from various database management system (DBMS) vendors. Then we describe the possible uses of replicated databases in the R/3 System to provide high availability.

Replicated databases or replicated database servers?

Distinguish between replicated databases in which the **data** is replicated (discussed here) and replicated database servers [Page 208] (such as Oracle Parallel Server or DB2 data sharing with DB2 Parallel Sysplex) in which the **DBMS** is replicated.

This section describes the following products and features:

- Oracle Standby Database [Page 187]

  This offers asynchronous log-based replication of a database to one site.

- Symmetric replication from Oracle

  This offers asynchronous and synchronous statement-based replication of data to one or more sites.

- High-availability data replication (HDR) from Informix [Page 193]

  This offers asynchronous and synchronous, log-based replication of a database to one site.

- Continuous Data Replication (CDR) from Informix (not yet available)

  This offers log-based replication of data at the table level to one or more sites. It is planned to support synchronous and asynchronous replication.

- Microsoft SQL Server Standby Database [Page 206]

  This offers asynchronous, log-based replication of a database.

- Replicated Standby Database for DB2 UDB [Page 197]

  This offers asynchronous, log-based replication of a DB2 Universal Database.

- Replicated Standby Database for DB2 for OS/390 [Page 201]

  This offers synchronous and asynchronous replication of the DB2 for OS/390 database.

SAP does **not** specifically recommend any of the above products

SAP experience in this area is limited, so no recommendations are made concerning the products and their possible uses. The information in this section is intended as an overview only. Therefore, you should **not** use this information to make important decisions without taking further advice.

Features

Replicated Database Strategies

This section discusses a number of important high availability issues that you should consider when selecting a strategy for replicating your database.

- Transaction serializability

  This means that any concurrent transactions committed to the primary database can always be replicated in the secondary database (the replica) with the resulting physical database (that is, the part updated by the transactions) being identical to the primary physical database.

  In general, log-based replication schemes tend to guarantee transaction serializability whereas statement-based replication schemes tend not to. Oracle symmetric replication offers row-level and procedural-level replication (row-level replication guarantees transaction serializability while procedural level replication does not). If you use a product that does not guarantee serializability, you must either serialize dependent concurrent updates at the application level or be able to live with a replicated database that can potentially differ from the primary database.

- Transaction loss

  Transaction loss means that, if the primary database shuts down for some unexpected reason, transactions that have been committed to the primary database might not be propagated to the replica, resulting in replication inconsistency. The problem is due to the fact that the primary database usually keeps a queue of transactions that, if the database fails, can no longer be propagated (if the database becomes available later without damage, it might then be possible to replicate such transactions).

  Asynchronous replication schemes in general might suffer transaction loss in the case of database failure, while synchronous replication schemes by definition guarantee no transaction loss.

- Schema level and database level replication

  The following perform replication at the **schema or table** level:

  – Oracle symmetric replication

  – Informix CDR

  – Microsoft

  The following perform replication at the **database** level:

  – Oracle standby database

  – Informix HDR

  – Replicated Standby Database for DB2 UDB

  – Replicated Standby Database for DB2 for OS/390

  Schema level replication requires extra effort in that you must define the list of tables, either whole tables or subsets (horizontal or vertical), to be replicated.

- Blob handling

Oracle symmetric replication and SQL*Server either do not handle blobs (long fields) or have very strict limitations on how they are handled. However, Informix HDR and the Oracle standby database feature, and Replicated Standby Database for DB2 for OS/390 have no restrictions on blob handling.

Using Data Replication for the R/3 System

This section describes what existing data replication products can and cannot do for the R/3 System. The following are possible uses of data replication with the R/3 System:

- Maintain complete, hot standby database

  To maintain a complete, hot standby database means that, if the primary database fails, the R/3 System can switch to the standby database and continue to function without any disruption or loss of replication consistency. This guarantees that all transactions committed to the primary database are propagated to the replica and no committed transactions are lost.

  Currently only Informix HDR in synchronous mode and Geographically Dispersed Parallel Sysplex for DB2 for OS/390 can be used for this purpose. With asynchronous replication you risk the loss of committed transactions.

  Often it is sufficient to have a standby database (loss of some transactions is acceptable) rather than a hot standby. If you only require a standby database, you could consider any of the log-based replication schemes (Informix HDR, Oracle standby database, SQL*Server replication, Replicated Standby Database for DB2 UDB). Informix CDR and Oracle symmetric replication cannot be used to maintain a standby database, since they are designed to replicate only part of the data.

  Standby databases are commonly used as an alternative to recovery or as a disaster recovery site.

- Data replication for distributed databases

  Data replication can be used to maintain one or more databases at remote sites. In such a scenario, each remote site has its own R/3 instance running against the database. The main function of these remote sites is to read data propagated from the primary site. For example, a big, multi-site corporation might install such replicated databases at its remote manufacturing facilities to read product information without having to access the central database. Such applications must be able to tolerate certain delays that might occur when data is replicated.

  To be used as a distributed database, the replication product must allow at least read access to the replica. This rules out Oracle standby database. Informix CDR and Oracle symmetric replication should only be used to replicate parts of the data.

  In general, these remote sites should be set up as read-only sites and updates should be made against the primary database. Some replication products might not allow updates to the replicated data, while other products support multi-site updates, including propagation of changes to all participating sites. For ease of use, updates at the remote site should only be allowed so as to support remote read applications and must not be propagated back to the primary database.

  The use of this type of replication should be transparent to the R/3 System.

- Data replication for report jobs

  In an environment where most of the batch jobs are report jobs, it might be desirable to run these jobs at a replicated site to reduce the load at the primary site. The assumption here, of course, is that these jobs can tolerate data that is slightly less up-to-date than the data at the primary site.

This use is very similar to the previous item, “Data replication for distributed databases”.

- Data replication as a software alternative to disk mirroring

  Disk mirroring is usually done at the disk or hardware partition level. Using data replication, a more flexible solution can often be achieved because data can be replicated at the database level or even at the table level.

Standby Databases

Use

This section describes standby databases in general. The basic concept of a standby database is to set up a copy of the production database on a second hardware system, so greatly improving the overall reliability of the database service. The standby database is used if the production database fails.

Features

With standby database solutions, you need to consider the effects on the R/3 application hosts connected to the database, when the database fails and the database host changes.

If the two database hosts form a cluster, it might be possible to use a standard switchover solution. For more information about related issues (such as how to configure the database, application hosts inside and outside the cluster and the database reconnect feature), refer to “See also” at the end of this section.

Usually the two database hosts do not form a cluster (that is, switchover software cannot be used and use of the R/3 reconnect feature is not recommended). The effects on application hosts vary, depending on where they run:

- All application hosts run external to the two database hosts

  After failure, all application hosts have to be restarted using a different profile to connect to the standby database. It is sufficient to set the profile parameter `SAPDBHOST` to point to the new database host.

- R/3 CI (central instance, including enqueue and message service) runs on the same host as the production database

  If the database fails, both database and CI are restarted on the standby database host. At the very least, `SAPDBHOST` has to be different in the profile of the CI. You need to change several parameters in the profiles of all external application hosts and these have to be restarted.

If the standby database was set up using asynchronous transfer of transactions (that is, log file shipping), all users have to be made aware – after work resumes using the standby database – that transactions might be lost. Users also have to be told to check whether the last changes they made are still available. The following applies:

- If the last transaction of a user is committed to the standby database, all previous transactions for that user are definitely committed as well.

- If the last transaction for a user is missing, the user has to check back to find the last transaction that was successfully applied to the standby database. All work done after this transaction is lost and has to be repeated manually.

See also:

Switchover Software [Page 216]

Documentation in SAPNet

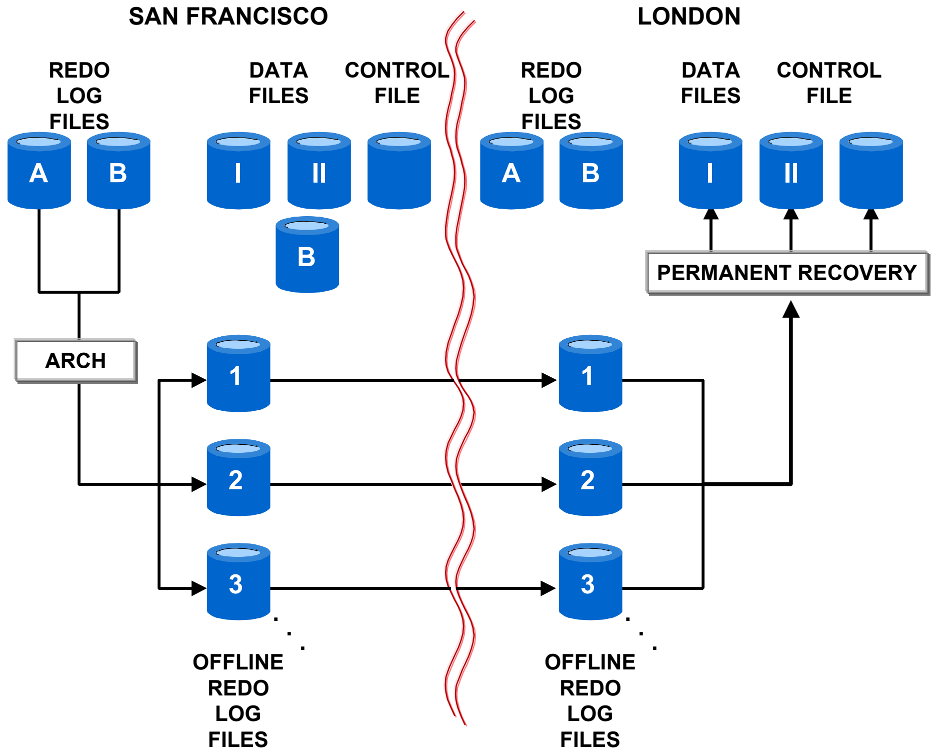
Oracle Standby Databases

Use

An Oracle standby database uses a copy of the production database on a second hardware system, so greatly improving the overall reliability of the database service. For more information about standby databases in general, see Standby Databases [Page 186].

The following diagram shows the setup for an Oracle standby database:

Oracle Standby Database



Features

- Setup overview

The production database is copied to a second location. The work done in the production database is recorded in redo log files. They are archived and shipped to the second location and applied there to keep the standby database up to date.

- Separation of standby system from production system

  The standby system is normally located at a remote site, since otherwise it too might be affected by whatever destroyed the production system.

- Standby is a copy of production database

  The standby database does not have to be an exact physical copy of the production database. Directory structures and file names might be different. The standby database might contain only parts of the production database. The standby database consists of copied data files, online redo log files and a standby control file created at the primary site and moved to the standby site.

  If BRBACKUP is used to make offline backups of the standby database, it has to be an exact copy. For more information, see “Oracle Standby Database with BRBACKUP and BRARCHIVE” below.

- Structural database changes

  Changes to the structure of the production database might affect the standby database. Certain changes are propagated to the standby database (that is, the control file of the standby database is updated). A discussion of potential problems with structural changes follows.

- Standby mode

  The standby database is mounted in standby mode. This means the database cannot be used in any way other than for recovery. Another consequence of mounting the database in standby mode is that it cannot be opened in the standard way. It has to be activated first and then opened. This prevents an accidental open of the database, which would invalidate the standby state of the database.

- Standby database runs in recovery mode

  Once the standby database has been mounted, it is put in recovery mode. The redo log files archived at the production site have to be shipped to the remote site and applied there using the database recovery mechanisms. This performs a “redo” of all work done in the production database in the standby database. The standby database always lags slightly behind, because the redo log file currently used by the production database cannot be shipped yet.

- What happens when the production database fails?

  If the production database becomes unavailable, the standby database has to be activated, shutdown, and then opened for normal use.

  When the production database fails, some committed transactions might be lost, because the current online redo log file that the production database was using at the time of the disaster might be inaccessible. The standby database can then only be recovered to the state reflected in the last archived redo log file.

  Before activating the standby database, always try to archive the current redo log in the production database, ship it to the standby site and apply it.

Perform a backup of the standby database

It is important to immediately perform a backup of the standby database once it is activated and opened for normal use. If no backup is performed and problems occur in the standby database, all work done since the activation is lost. This backup is also important to enable you to subsequently restore the database at the production site.

- What happens if the production database comes online again?

  If the production site becomes available again, SAP recommends not to use (that is, start) the database. The reason for this is that it is impossible to apply "new" transactions of the "production database" to the standby database. These transactions are lost when you revert to the original configuration (see next point below).

- Revert to the original configuration as soon as possible

  If the standby database is put into productive operation due to a disaster, it should then be considered the production system. Once the disaster situation is resolved at the production site, you have to decide how to switch back to the original configuration (if that is desired at all).

Problems with Oracle Standby Databases

The following are the main missing and problematic features with the Oracle Standby Database:

- Structural database changes

  This section discusses the following types of structural changes of the primary database only:

  – When a data file is added, the system uses information in the redo log to update the control file of the standby database, but the data file itself is not created at the operating system level. The file can be added in either of the following ways:

    - Copy a backup of the file from the production database to the standby database
    - Enter the following command (the size does not have to be specified because this information is already available in the control file):

      ```
      alter database create datafile
      ```

      If you forget to add a file and the recovery process finds redo data for that file in the redo logs, the recovery returns errors and aborts. You can now add the data file as described above and resume the recovery.

  – When a data file is dropped from the primary database, the standby control file is updated, that is, the data file is dropped from the standby database as well. Note that the data file is not deleted at the operating system level. You still have to do this manually.

  In general, structural changes should not cause major problems. Refer to the appropriate Oracle documentation for a detailed description of the effects of structural database changes.

- Copy of archived redo logs not automated

  Oracle does not provide anything to copy the archived redo logs from the primary database to the standby database. You have to manage this yourself. See "Oracle

Standby Database with BRBACKUP and BRARCHIVE” below for a possible solution for this.

- Recovery of standby database not automated

  Oracle does not provide a mechanism to automatically start the permanent recovery of the standby database. You have to initiate recovery with the command `recover standby database`. See “Oracle Standby Database with BRBACKUP and BRARCHIVE” below for a possible solution to this problem.

- I/O errors and disk failures

  These can cause datafiles to go offline and such files are not recovered. If this happens on the standby database, inconsistencies result and tablespaces might be lost. In this situation, you have to make a new copy of the production database to set up the standby database from scratch again.

- Data corruption during transfer

  Compression utilities used to electronically transfer files from one machine to another might cause data corruption. Make sure that the data files and the archived redo log files are transferred in such a way that no corruption or loss of files can occur.

- The effects on applications connected to the database

  See "Problems with Standby Databases" in Standby Databases [Page 186].

Oracle Standby Database with BRBACKUP and BRARCHIVE

BRBACKUP and BRARCHIVE support a standby database, as follows:

- BRBACKUP can be used to make offline backups of the standby database. BRBACKUP retrieves information about the database structure from the production database and backs up the standby database accordingly. This implies that the standby database is an exact copy of the production database. The advantage of this setup is that no backups have to be done at the production site, so reducing the load there.
- BRARCHIVE can be used to automate the process of copying archived redo log files to the standby database and recovering the standby database. BRARCHIVE also saves the redo log files at the standby site. A prerequisite is to be able to NFS mount (UNIX) or share (NT) the archive directory of the standby database from the production system.

SAP does not make any recommendations for use of the Oracle standby database feature.

Activities

Creating an Oracle Standby Database

💡

Before you start creating the standby database, you need to have installed the Oracle Software at the standby site. It is best if the software setup is identical between the two nodes. This includes making an Oracle parameter file `init<SID>.ora` available as well. Later we discuss how to use SAP tools (BRBACKUP, BRARCHIVE) to simplify the maintenance of a standby database setup. For the SAP tools to work, you have to install them at the standby site. You also have to make available the parameter files (that is, `init<SID>.dba` and `init<SID>.sap`) for the SAP tools. Both the Oracle and SAP parameter files are usually found in `$ORACLE_HOME/dbs`. Therefore, you can simplify setup by copying this directory and all its contents to the standby site.

1. Take a backup (online or offline) of the data files of the production database
2. Create a control file at the production site to be used at the standby site, by entering the following command:

   ```
   alter database create standby controlfile as <filename>
   ```

3. Archive the current online redo log of the production database, by entering the following command:

   ```
   alter system archive log current
   ```

4. Transfer the backed up data files, the control file and all archived redo log files to the standby site.

Maintaining an Oracle Standby Database

1. Startup the standby database without mounting it, by entering the following command:

   ```
   startup nomount
   ```

2. Mount the standby database in standby mode, by entering the following command:

   ```
   alter database mount standby database
   ```

3. Transfer archived redo log files from the production database site to the standby database site.
4. Put the standby database in recovery mode, by entering the following command:

   ```
   recover standby database
   ```

Switching Back to the Primary System After a Disaster has been Resolved

SAP recommends that you adopt one of the following procedures for switching back to the primary database after the failure has been resolved:

- **Standard** procedure for switchback to primary database

  ⚠

  Backup of standby database must be full offline (see step 2)

  A standard offline backup with BRBACKUP opens the database after the backup to update backup log information. Therefore, you must **not** do a standard offline backup with BRBACKUP here because the opening of the standby database after the backup means that the standby database and its backup are no longer identical. This in turn means it would not be possible to mount the database at the standby site in standby mode with a standby control file created at the production site, once the database has been restored there.

  A new backup type has been introduced in BRBACKUP to solve this problem. This backup type is documented here only and you should only use it in the case where a standby database is backed up offline to enable a restore at the production site. With this backup type the database is shut down and backed up offline, and is **not** restarted after the backup.

  Set the backup type in the BRBACKUP profile (`init<SID>.sap`) as follows:

```
backup_type = offline_stop
```

Perform the following steps with this procedure:

a. Stop production use of the standby database (that is, shut down all application services and close the database)

b. Take a complete backup of the standby database (data files, online redo log files, control files) and copy this to the production site. If you use BRBACKUP to create the backup, use a `backup_type` of `offline_stop` as described above.

c. At the production site, clear the directory for archived redo logs (usually `saparch`) of all files, including BRARCHIVE log files.

d. Start up the database at the production site.

e. Create a standby control file at the production site and move it to the standby site.

f. Mount the database at the standby site in standby mode.

g. Start the application services (profiles have to point to database host at the production site).

h. Resume normal production and standby operation of the two databases.

i. Resume the normal backup strategy.

The standard procedure outlined above is straightforward and easy to handle. On the other hand, the R/3 System is unavailable for a long period (that is, the time it takes to copy the whole database, including offline backup and restore)

- **Advanced** procedure for switchback to primary database

  Depending on how long the standby database was used, one or more backups are available. To repeat, you must back up the standby database immediately after activation (if the standby database was used for a longer period, you should have taken regular backups). You must have at least one (regular) backup of the standby database to use the advanced procedure described below

  Perform the following steps while the standby database is still open for normal use:

a. Restore a regular backup of the standby database to the production site (data files, backup control file) and all redo logs archived since the backup.

b. Mount the database at the production site and recover it, using the following commands:

```
startup mount
recover database until cancel using backup controlfile
```

c. Apply all available archived redo logs

d. Cancel recovery and shutdown the production database

e. Stop the standby database (that is, stop all hosts running R/3 application services and shut down the standby database too).

   The standby database is closed for the remaining steps.

f. Copy the current control file, all online redo log files and, if applicable, archived redo log files (that is, redo logs that have not been applied to the database at the production site yet, because they have just been archived) from the standby database to the production site.

g. Mount and recover the database at the production site using the following commands:

```
startup mount
recover database
```

Since the current control file is used, this performs a complete media recovery. The database at the production site is up-to-date at the end of the recovery.

h. Start up the database at the production site.

i. Create a standby control file at the production site and move it to the standby site.

j. Mount the database at the standby site in standby mode.

k. At the production site, clear the directory for archived redo logs (usually `saparch`) of all files, including BRARCHIVE log files.

l. Start the hosts running application services (profiles have to point to database host at production site).

m. Resume normal production and standby operation of the two databases.

n. Resume the normal backup strategy.

The advanced procedure outlined above has a short downtime (the time taken after shutdown of the standby database to copy the current control file, online redo logs and, maybe, archived redo logs and to recover to the current point in time). On the other hand, it is a more difficult procedure with greater risk of handling errors.

Both switchback procedures described above assume a backup of the standby database is restored to the production site. The backup must have been taken after the standby database was activated and opened for normal use. It is **not** possible to restore a backup taken before the activation and then recover beyond the activation of the standby database to the current point in time.

See also:

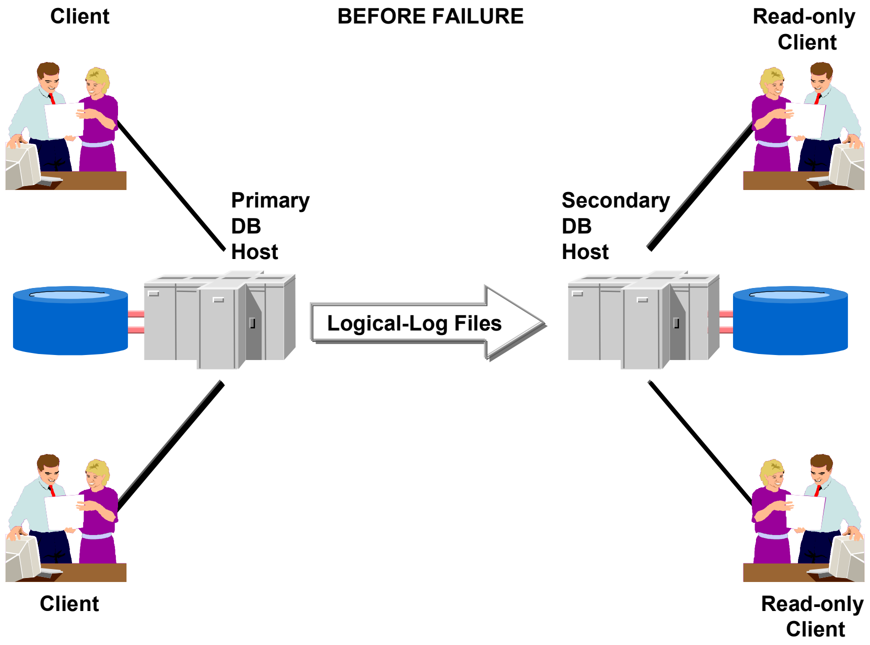
Database Administration(Oracle) with SAPDBA [Ext.]

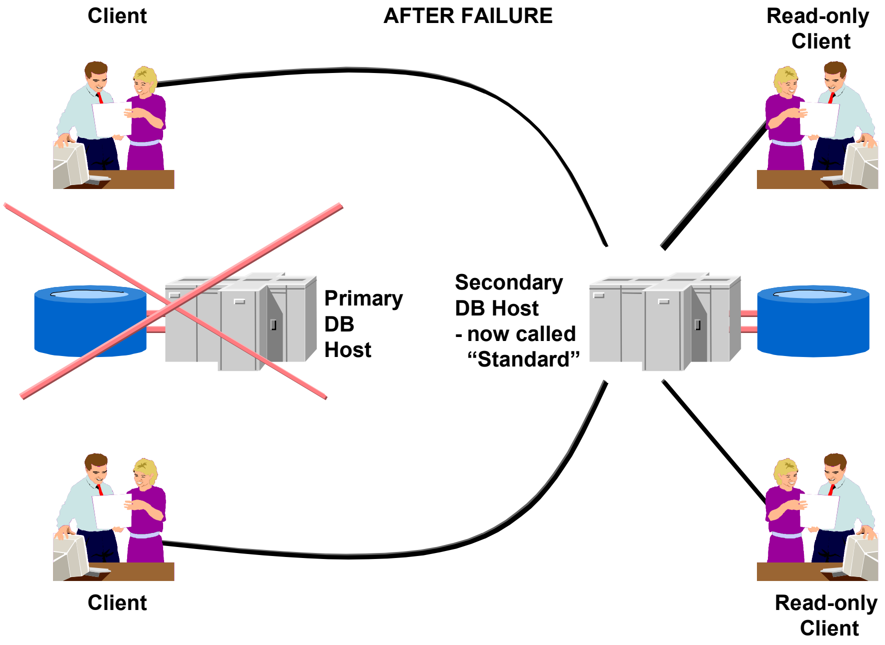
Informix High-Availability Data Replication

Use

With Informix High-Availability Data Replication (HDR), you set up a secondary database – an identical copy of the entire production database – on a second hardware system. This greatly improves the availability of the database service. The data in HDR is more available for client applications wishing to access it since, if the local copy of the data fails, the remote copy can be accessed. The following diagram shows how it works:

Informix High-Availability Data Replication (HDR)





Features

- Separation of secondary system from primary system

The secondary system is normally located at a remote site, since otherwise it too might be affected by whatever damages the primary system. The two servers must be connected by a TCP/IP network connection.

- Secondary is exact physical copy of primary database

  It is essential that the secondary database is an exact copy. Therefore, the physical layout of the secondary database must be identical to the primary database (that is, same disk configuration, same directory structures and filenames). If this is not the case, you can not successfully perform the restore on the secondary server (see below, “Setting up Data Replication”).

- Data-replication buffers store data to be replicated

  These buffers (the same size as the logical-log buffers) are used to store logical log data from transactions that are then sent to the secondary server. When used with the R/3 System, the updates are passed synchronously to the secondary server. This avoids uncommitted or partially committed transactions on the secondary server if the primary server fails.

- Secondary can operate as read-only server

  An added benefit of HDR is that the secondary database server can function as a read-only database server for certain clients, so improving performance by allowing distribution of the overall system workload. Since the secondary server operates in “logical-recovery” mode, it can not accept updates.

- Stress tests indicate no substantial performance decrease

  Data replication in an SAP environment does not normally have a significant impact on performance, as indicated by stress tests carried out to measure this.

  Use the ON-Bar data recovery tool with Informix HDR. With ON-Bar, you can perform parallel backup and restore on an Informix HDR system. For more information about ON-Bar, see ON-Bar for Data Recovery [Ext.].

Activities

Setting up Data Replication

To start data replication:

1. Do the following on the primary server:

   a. Perform a full level-0 archive (that is, including all dbspaces)

   b. Back up the logical log

   c. Change the database server mode to “primary”

2. Do the following on the secondary server:

   a. Perform a physical restore (using level-0 archive taken above)

   b. Set the database server mode to “secondary”

   c. Perform a logical restore (using logical log backup taken above)

Detection of Data Replication Failures

If data replication fails, this might be either due to a network problem or a failure of one of the servers. Either of the following conditions is interpreted as a data-replication failure by either of the database servers:

- Specified time-out period exceeded

  The confirmation from the other database server in a pair does not arrive within the specified time.

- Pinging fails

  The periodic reciprocal signaling (pinging) between the servers fails on four consecutive attempts. In this case, the sender assumes failure.

Failure Scenarios

Consider the following two principal scenarios:

- Failure of secondary server

  The primary server remains in online mode and no switchover is necessary.

- Failure of primary server

  The secondary database server can behave in any of the following three ways, depending on how you configure your system and what decisions you make:

  – No switchover

    The secondary server remains in logical-recovery mode, that is, no action is taken. This is acceptable if the primary server is likely to come online again soon and data-replication can be restored quickly.

  – Manual switchover

    The secondary server remains in logical-recovery mode, awaiting manual switchover. Contact the SAP-Informix Competence Center in Walldorf, Germany for further details since you need additional Informix-specific software and advice in this area.

    💡

    If your network is not entirely stable, you must use manual switchover.

    ⚠

    You need to identify the problem and restart data replication with two servers as soon as possible, since the surviving server is running in standard mode (that is, without data replication on another server) until data replication is fully back in action. Refer to “After Switchover”.

Result

After a successful switchover there are the following possibilities to restore the data replication pair:

- Both database servers revert to their original type

When the original primary database server comes back online, the data replication connection is automatically established. The secondary server, which has meanwhile been acting as a standard server, shuts down gracefully and then switches back to being a secondary database server. The logical logs from the period when the primary server was out of action are then transferred from the secondary server onto the primary

- Both database servers change their original type

  After a switchover to the secondary server, which has meanwhile been acting as a standard server, the secondary server switches type directly to become the primary server, without shutting down first. The primary server switches to become a secondary server and can then take part in data replication again.

See also:

Informix documentation.

Contact the SAP-Informix Competence Center in Walldorf, Germany for more information.

Replicated Standby Database for DB2 UDB

Use

After you have restored an offline backup of the production database on the standby system, you must install an additional DB2 “user exit” to trigger the transfer of the inactive log files from the production system to the standby system and initiate a rollforward to the end of the log as soon as the log file has been transferred. DB2 uses Coordinated Universal Time (CUT) to roll forward.

> Contact the IBM Competence Center in Walldorf, Germany for further information about the DB2 replicated standby database. DB2 Universal Database (DB2 UDB) now fully supports HACMP on AIX and Microsoft Cluster Server (MSCS) on Windows NT. The IBM Competence Center can also give you more information about this.

MIMIX Standby Database for DB2/400

Use

The DB2 database on the AS/400 hardware platform can be protected against failure by using the MIMIX high availability solution from Lakeview. MIMIX has been specially adapted for the R/3 System. MIMIX protects database and application hosts – and attached clients using application services – from the consequences of a failure in the database service.

During normal production, R/3 data in both SQL tables and Integrated File System (IFS) files is replicated synchronously from the primary to the standby machine. If the primary machine fails, MIMIX switches in the standby and adjusts the R/3 network so that application services recognize the new location of the database service on the standby machine. Therefore, client applications can normally continue with minimal interruption and data loss.

The information here is a short introduction. Contact the IBM Competence Center in Walldorf, Germany or Lakeview for further information about MIMIX.

Integration

MIMIX is fully integrated with the R/3 System. Note that MIMIX only protects the machine with the database service. If you locate the R/3 central instance on a different machine, you introduce a second point of failure in the network, which makes high availability more difficult to achieve.

To minimize points of failure, SAP advises you to locate the central instance on the same host machine as the database and critical IFS files.

Prerequisites

- Define the standby machine as the hub and the primary machine as the satellite (that is, for the OptiConnect configuration).
- Make sure that journaling is active for database tables.
- Avoid using remote IFS files. For example, use database tables instead for spool output. This is because IFS requires entire files to be shipped rather than individual records.

Features

The MIMIX solution consists of the following components:

- MIMIX/400

  Provides asynchronous replication of changes to data in R/3 tables. Changes are transmitted from the source (normally the primary) to the target machine (normally the standby), where they are stored in log spaces before being applied to the database tables.

- MIMIX/Object

  Provides asynchronous replication of object-level changes. For the R/3 System, this means changes to IFS files and the creation or deletion of SQL tables. You can also do `config objects` in SQL to replicate these objects.

- MIMIX/Switch

  Monitors the database machine and, on failure, controls the switch to the standby, including necessary network adjustment. MIMIX/Switch contains customized features for the R/3 System.

Activities

You need to perform the following tasks:

1. Before starting normal production, synchronize the database and standby machines. This involves copying all IFS files (both critical and non-critical) and all SQL tables from the database to the standby machine, with the R/3 System down.
2. Configure a standby central instance.

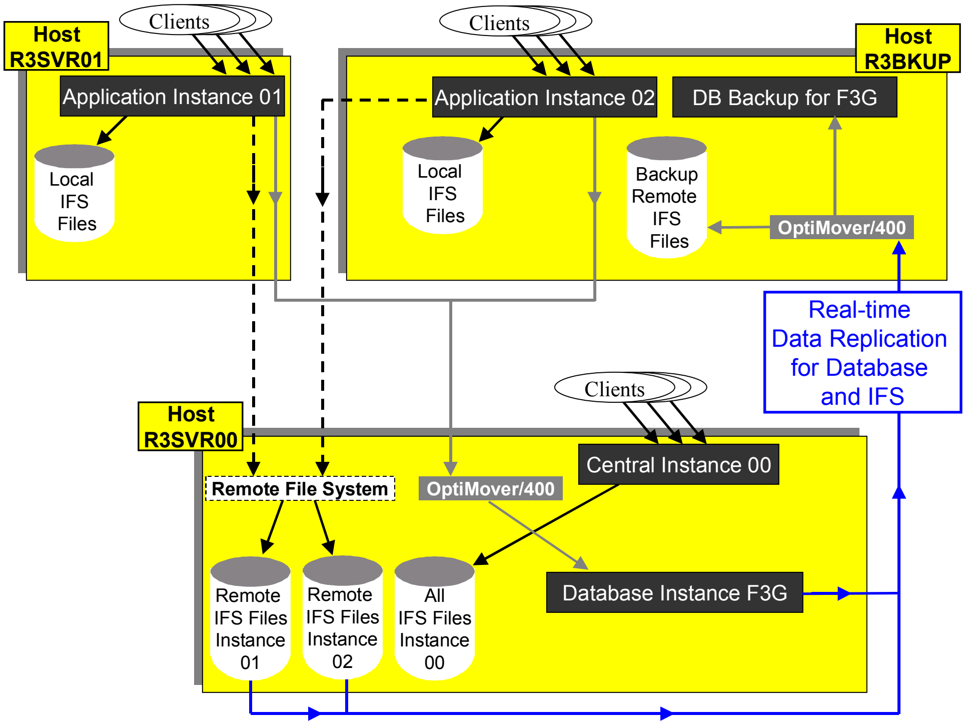
3. When the R/3 System is up, start data replication.

   MIMIX automatically copies database changes (that is, changes to critical IFS files and to all SQL tables) to the standby machine.

   MIMIX automatically handles failures, and prompts for operator action if required. Do not make profile changes to the R/3 System during the failure period. During the failure period, data generated on the standby machine is stored for future replication, once the failed primary machine is back in service.

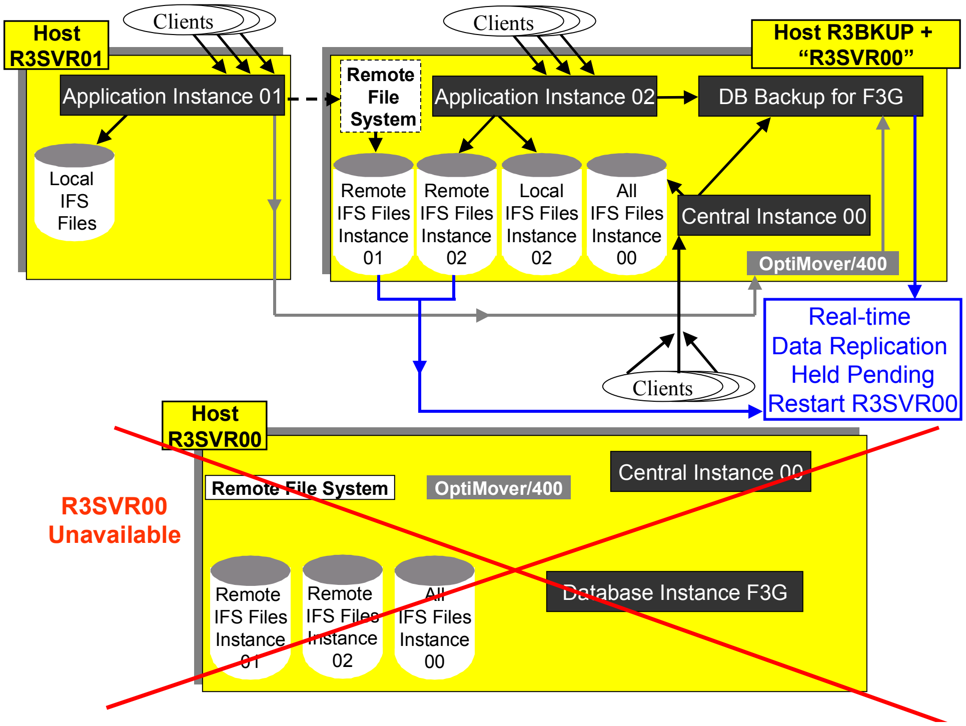
The following diagrams show the configurations before and following failure:

Before Failure



Following Failure

OMS/400 Standby Database for DB2/400



See also:

OMS/400 Standby Database for DB2/400 [Page 200]

IBM documentation: *SAP R/3 Implementation for AS/400 (document number SC24-4672-01)*, Appendix E "High Availability Solutions from ISVs"

OMS/400 Standby Database for DB2/400

Use

The DB2 database on the AS/400 hardware platform can be protected against failure by using the OMS/400 high availability solution from Vision Solutions. OMS/400 automatically replicates changes from the primary to the standby host machine. As changes are made on the primary machine, the OMS/400 sending process sends the journal receiver transactions to the standby machine, where the transactions are placed in a user space. Apply jobs on the secondary machine then process the user space entries to update the database on the standby machine. The data replication process is referred to as mirroring.

The information here is a short introduction. Contact the IBM Competence Center in Walldorf, Germany or Vision solutions for further information about OMS/400.

Integration

OMS/400 can be configured to run with the R/3 System.

Features

The following features are relevant to the OMS/400 configuration for the R/3 System:

- Synchronization check of mirrored objects, automatically or on demand
- Flexible journal management facilities
- Full data integrity
- Support for commitment control and null value
- No user intervention for monitoring and maintenance

See also:

MIMIX Standby Database for DB2/400 [Page 197]

IBM documentation: *SAP R/3 Implementation for AS/400 (document number SC24-4672-01)*, Appendix E "High Availability Solutions from ISVs"

Vision Solutions documentation

Replicated Standby Database for DB2 for OS/390

Use

You can protect your R/3 data stored on DB2 for OS/390 against failure by setting up a replicated standby database with IBM solutions such as:

- Geographically Dispersed Parallel Sysplex (GDPS), also called "Geoplex"
- DB2 Tracker Site

The site where the R/3 database server is initially started is known as the "primary site" and the site where the database is replicated as the "standby site". In IBM documentation these sites are called:

- "Primary" and "tracker" (for DB2 Tracker Site)
- "Production" and "recovery" (for Geographically Dispersed Parallel Sysplex).

Integration

You can integrate all the replicated standby databases discussed in this section with the R/3 System.

Features

DB2 for OS/390 can be synchronously and asynchronously replicated to protect against database failure, as follows:

- Synchronous database replication provides a consistent copy of the data on the standby site. For R/3 applications, this means that the database server on the standby site contains all updates until a specific point in time. Since the data on the standby site is consistent, the application can be emergency restarted in the standby location without having to perform time-consuming data replication. However, application performance can be affected by the replication.

Geographically Dispersed Parallel Sysplex uses synchronous database replication. For more information, see Synchronous Replication of DB2 for OS/390 [Page 202].

- Asynchronous database replication has no performance impact on most applications, but updates that are not copied to the standby system before the failure are lost. Therefore, asynchronous recovery means that the database is less up-to-date than synchronous recovery.

  DB2 Tracker Site uses asynchronous database replication. For more information, see Asynchronous Replication of DB2 for OS/390 [Page 205].

See also:

For more information see the following IBM documentation:

- *DB2 for OS/390 Version 6 Administration Guide* (SC26-8957-00)
- *PLANNING FOR IBM REMOTE COPY* (SG24-2595-00)
- *RAMAC Virtual Array: Implementing Peer-to-Peer Remote Copy* (SG24-5338-00)
- *Geographically Dispersed Parallel Sysplex: the S/390 Multi-site Application Availability Solution* (GF22-5063-00)

Synchronous Replication of DB2 for OS/390

Use

You can use synchronous replication of DB2 database data if your site:

- Requires the standby system to always be fully up-to-date with the primary system
- Can accept some performance impact on write operations at the primary location

You can use Geographically Dispersed Parallel Sysplex (GDPS) to copy data synchronously to the standby site.

Features

Geographically Dispersed Parallel Sysplex (GDPS), also called "Geoplex", is a Parallel Sysplex cluster spread across two sites, with all critical data mirrored between the sites using Peer-to-Peer Remote Copy (PPRC). It offers a controlled site switch for both planned and unplanned site outages, with no data loss, maintaining full data integrity in the R/3 System.

PPRC is a high availability solution that provides storage-based disaster recovery and workload migration with the capability to copy data in real time to a remote location.

A PPRC data copy to the standby system is simultaneously synchronized with the I/O operation on the primary system. This means that PPRC does not consider the primary system write operation complete until it has received a signal from the standby system that the operation is complete. This signal is sent from the disk system of the standby system after the data is saved in the cache. The response delay increases the primary system's response time.

Geoplex consists of a Parallel Sysplex cluster spread across two sites with one or more OS/390 systems at each site. The maximum distance between the two sites is 40 kilometers.

Geoplex supports the following configuration options:

- Single site workload

  This configuration is intended for those systems that have critical workload on the primary site and expendable work on the standby site.

- Multiple site workload

  This configuration is intended for those systems that have critical and expendable workload on both sites.

In this section only the single site workload configuration is described. For more information about the multiple site workload see the IBM documentation *Geographically Dispersed Parallel Sysplex: the S/390 Multi-site Application Availability Solution*.

Geoplex consists of the following systems:

- Primary systems

  The primary systems execute mission-critical work.

- Standby systems

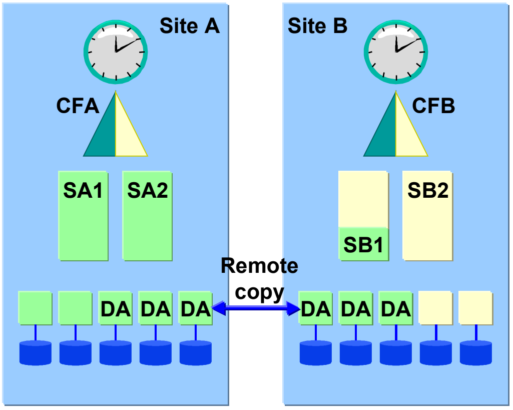
  The standby systems do not run any mission-critical work. They can run work that is considered expendable. If one or more of the primary systems is unavailable, expendable work on the standby systems is slowed down or stopped and processing resources are provided to take over the mission-critical work.

- Controlling system

  The controlling system coordinates switchover.

Geoplex uses Parallel Sysplex cluster facilities to communicate between systems. For example, each Geoplex system joins the Geoplex Parallel Sysplex cluster group. Each system monitors the Parallel Sysplex cluster, coupling facilities, and storage subsystems and maintains Geoplex status. The following graphic illustrates a possible configuration and is used for the description in the section "Activities" below:

Geoplex with Parallel Sysplex



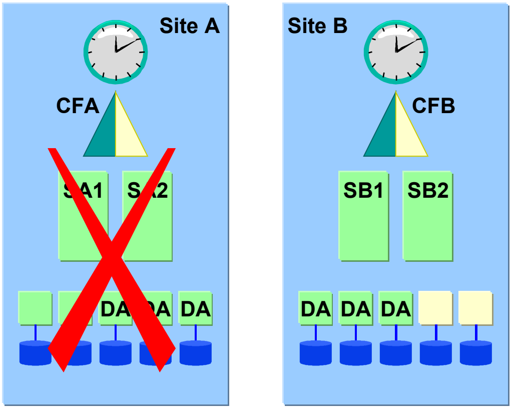
Activities

You configure Geoplex so that:

- The mission-critical R/3 System is running on the primary systems SA1 and SA2, located in the primary site A. In this example DB2 data sharing is used but you can also run the R/3 database server without DB2 data sharing in the Geoplex environment.
- Two standby system on site B provide processing resources when one or more of the primary systems are unavailable or site A is unavailable.
- The controlling system SB1 monitors the Geoplex configuration and coordinates Geoplex processing.
- Database DA on primary site A is mirrored to the standby copy on site B.

The following graphic shows the situation after site A fails or is shut down.

Geoplex Following Failure or Shutdown at Site A



After shutdown or failure at site A:

1. Geoplex freezes the standby copy of data on site B to guarantee data consistency. PPRC has stopped working because site A is not operational.
2. Geoplex initiates a site takeover so that:
   - Standby copy of the data is now considered the primary copy.
   - Active standby system SB1 expands.
   - Inactive standby system SB2 is started to acquire processing resources needed to run R/3.
   - R/3 database server is started on the systems SB1 and SB2 on site B.
3. The R/3 System administrator starts the application servers on the standby site.

When site A resumes operation:

- OS/390 operator can initiate the switch from site B to site A.
- Geoplex restores the original configuration.
- The R/3 System administrator starts the application servers on the primary site.

After shutdown or failure at site B:

- Remote copy processing is suspended.
- One of the primary systems in site A assumes the role of the controlling system.
- The R/3 System remains available.

When site B resumes operation:

- OS/390 operator can initiate the switch from site A to site B for the controlling system.
- Geoplex restores the original configuration.

Asynchronous Replication of DB2 for OS/390

Use

Asynchronous replication of data is designed for those sites that:

- Need to maintain the highest levels of performance on their primary system
- Need to support extended distances between primary and standby system
- Can accept a gap of a few seconds between writes on the primary system and the subsequent write updates on the standby system

R/3 on DB2 for OS/390 supports asynchronous data replication with DB2 tracker.

Features

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's data. No independent work can be run in the DB2 subsystem on the standby site. For example, you cannot update the catalog and directory or the data at the standby site. The standby site is also called the "tracker site."

To copy the data between primary site and standby site, Extended Remote Copy (XRC) is used. XRC is a remote asynchronous copy facility with minimal performance impact on most applications during normal operation. XRC automatically sends copies of updated data to the standby system. To maintain high performance at the primary location, XRC lets the I/O operation on the primary database server signal completion before receiving confirmation of the write on the standby system's database server. Since this secondary copy is normally only seconds behind the primary write, there is usually little or no data loss if a system failure occurs when data is "in transit" between the two locations.

You can operate a read-only database server on the standby site. The standby site disallows update commands like `INSERT` or `DROP` but allows read-only `SELECT` statements.

Activities

- To set up the standby site:
  1. Create a mirror image of your primary DB2 subsystem or data sharing group.
  2. Set the subsystem parameter `TRKSITE` to `YES` on the standby site.
  3. Send full offline image copies of all DB2 data of the primary site to the standby site.

- To establish a recover cycle at the standby site:

  When the standby site has full image copies of all data at the primary site, the BSDSs and the archive logs are copied from the primary site to the standby site. A `LOGONLY` recovery runs periodically on the standby site to keep the shadow data up-to-date.

  If the tablespace or partition is reorganized, loaded, or repaired with the `LOG(NO)` option, send a full image copy of any objects to the standby site.

- If a disaster occurs at the primary site such that the primary database server fails:

  The standby site becomes the takeover site. Since the standby site has been shadowing the activity on the primary site, DB2 recovery does not have to use image copies, so the time to switch over is usually minimal.

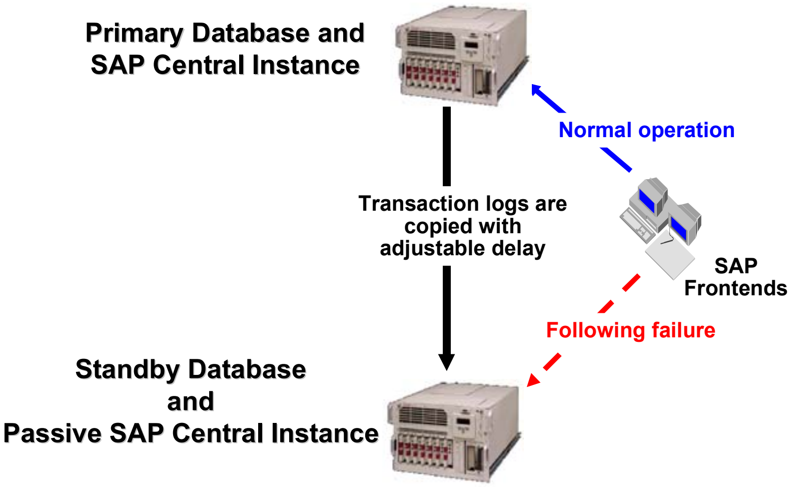
  When the primary site resumes operation, you can then switch back to the primary system.

Microsoft SQL Server Standby Database

Use

The Microsoft SQL Server can be protected against failure by setting up a standby database. The standby database can be brought online in the event of primary database failure. The standby contains an up-to-date copy of the primary database and runs in standby mode. The copy is set up by an initial restore of the primary database to the standby database followed by periodic shipping of backed-up transaction logs from the primary to the standby database, known as “log shipping.”

The main advantage of a standby database is to allow the system to be quickly brought back into production following complete failure of the primary database:



Integration

In a system where the central instance and database are on the same machine, a standby database alone is insufficient to ensure continued system operation after failure. Since a failure also affects the central instance, a backup version of it should be installed on the standby server. During normal operation it is inactive, but can quickly be brought online when required.

For more information on how you can combine Microsoft SQL Server Standby Database with the Microsoft Cluster Server on Windows NT [Page 220], see Comprehensive Microsoft SQL Server High Availability Solution [Page 214].

Features

- The standby database can be used for read-only purposes by non-critical applications such as reporting. This means that it must always be left in a logically consistent state after log shipping. Therefore, an undo file must be specified on the standby database, to allow rollbacks of uncommitted transactions contained in the transaction log from the primary database.
- More frequent log shipping means:
  - Higher processing workload
  - Reduced time to bring the standby database online after a failure on the primary database
  - More up-to-date standby database, so there is less data loss following failure in which the current transaction log on the primary database is destroyed
- A single standby database can act as backup for several production databases, as it is unlikely that all production databases fail at the same time (especially if they are on geographically separate sites).
- Log shipping can be implemented using buffering. This means that the transaction logs are backed up regularly (such as every 10 minutes) then held for a longer period (such as two hours) so that troubleshooting and decision support queries can be performed. Then the logs are shipped and applied to the standby database.

Activities

1. Initially, you back up the primary database and restore it to the standby database.
2. Periodically, the system automatically performs log shipping.
3. If the primary database fails, you:
   a. Determine the cause of the problem
   b. In view of your diagnosis, decide which transaction logs from the time of failure should be applied to the standby database
   c. If required and if possible, back up the current transaction log from the primary database
   d. Ship transaction logs to the standby database as required
   e. Switch users to the standby database, which is now the production database

See also:

The alias “HA” in SAPNet, then look in the media center

Replicated Database Servers

Use

Replicated database servers help to ensure continuous operation of the database management system (DBMS) by duplicating its functionality. This leads to an increase in the availability of database services by replicating the underlying components (such as database background processes, memory buffer areas, and so on).

Replicated database servers or replicated databases?

Distinguish between replicated database servers in which the **DBMS** is replicated (discussed here) and replicated databases [Page 183] in which the **data** itself is replicated.

The section describes the following replicated database servers:

- Oracle Parallel Server (OPS) [Page 208]
- Data Sharing for DB2 for OS/390 [Page 210]

Oracle Parallel Server

Use

Oracle Parallel Server (OPS) is an example of a replicated database server [Page 208]. This section describes the OPS architecture for background information only.

SAP does **not** currently encourage the use of OPS, neither as a high availability solution nor as an option to boost performance. This is due to the administrative overhead of installing and maintaining OPS and the limited scalability in an online transaction processing (OLTP) environment.

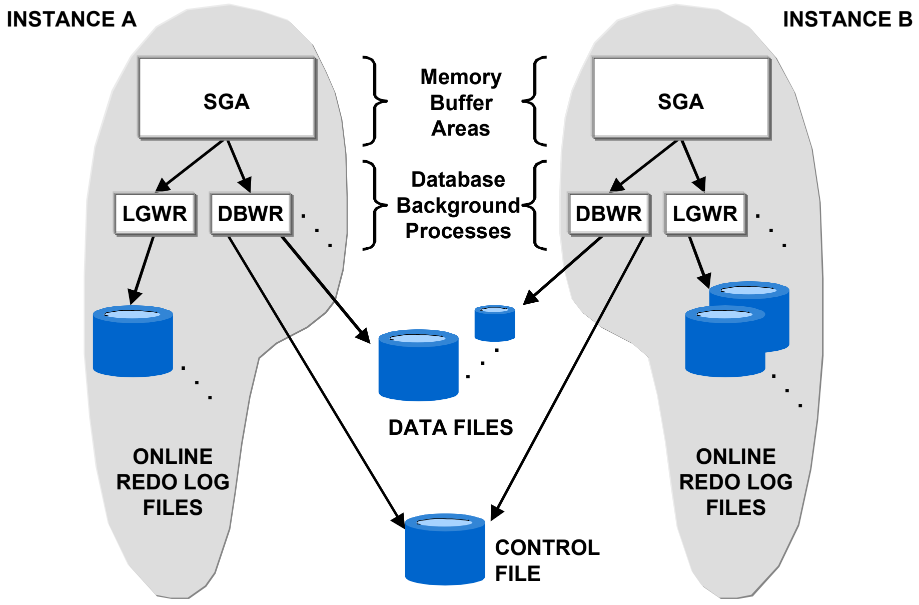
An open database is comprised of several main components, that is, data files, online redo log files (transaction log), control file/s, SGA (buffer area in memory), and a set of database background processes (DBWR, LGWR, and so on). The SGA and background processes are referred to as “the database instance”. Each instance has a private set of online redo log files. In a normal installation you only have one database instance per database (the term "database" here refers to the data files and control files).

With OPS it is possible to have multiple database instances running on top of the same database. This architecture is particularly beneficial in cluster hardware environments since it enables the DBMS to make better use of the available CPUs. See Cluster Technology [Page 130] for more information about clusters.

Features

The following diagram illustrates a typical OPS setup:

Overview of Oracle Parallel Server (OPS)



SGA = System Global Area
DBWR = Database writer
LGWR = Log writer

This setup provides additional protection against database instance failure as follows:

- If one instance fails, a surviving instance performs the recovery for the failed instance. Transactions open at the time of the instance failure are rolled back and committed transactions are applied. The surviving database instance continues to provide the database service.
- Application servers connected to the failed database instance can reconnect to a surviving database instance and continue.

Activities

1. You consider using OPS for extra high availability of the DBMS.

   OPS for high availability use in an R/3 environment implies that you have two database instances configured (that is, in a cluster with two nodes) and all application server work processes connect to one of these instances. The second database instance is idle and acts as a standby in case of failure. If the production database instance fails (for example due to process errors, memory faults, or CPU error), the application server work processes reconnect to the standby instance and continue processing. For more information, see DB Reconnect [Page 174].

   A drawback of OPS is that it requires a raw device installation, in other words, the data files, redo log files and control files have to be created as UNIX raw devices. This makes the administration (for example, extension of a tablespace, backup, recovery) more difficult.

2. You consider using OPS principally to boost performance on your system, using multiple database instances concurrently.

3. In both the above cases, you must make sure that you have SAP approval before going into production with OPS. SAP supports OPS **only** for approved installations. The approval process is as follows:

    a. You run several tests with your hardware partner using the R/3 System, OPS, and operating system release that you intend to use.

    b. SAP evaluates your project.

    c. SAP and Oracle provide consulting for OPS installation and maintenance.

    Contact SAP to find out more about this release procedure.

Data Sharing for DB2 for OS/390

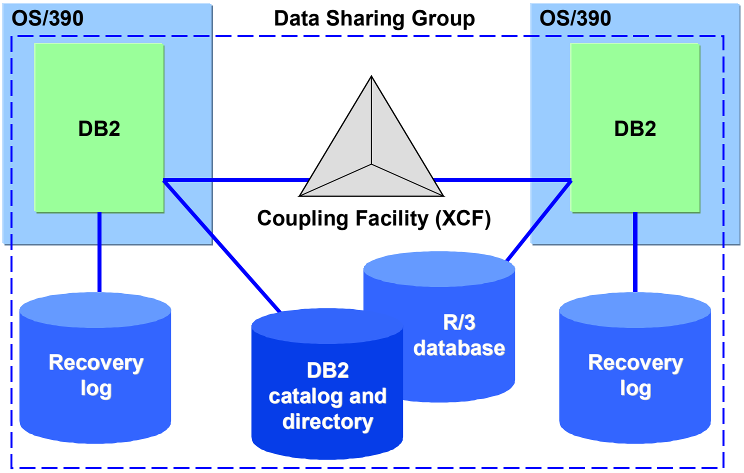
Use

You can use DB2 Data Sharing and R/3 Sysplex Failover Support as a high availability solution to protect the database of your R/3 System against failure.

Features

To increase availability and scalability, you can set up DB2 for OS/390 as a parallel database, known as DB2 data sharing. With this setup, multiple DB2 subsystems share the same R/3 database. Each subsystem is called a data sharing member, and the set of subsystems is referred to as a data sharing group. All members of the group use the same shared DB2 catalog and directory. This setup is shown in the following graphic:

DB2 Data Sharing



R/3 Sysplex Failover Support is the implementation of DB reconnect [Page 174] for DB2 for OS/390. It allows each application server to recognize two database servers, a primary database server that is used normally, and a standby database server that is used in the event that the application server can no longer work with the primary database server. This can be caused by failure in any of the following:

- ICLI server
- DB2 subsystem
- OS/390 system
- Entire S/390 hardware
- Network

We recommend you to have separate ICLI server instances for primary and standby connections. This means that all primary connections to a DB2 member are made to one ICLI server instance and all standby connections to another ICLI server instance, which is connected to the same DB2 subsystem, as shown in the graphic below.

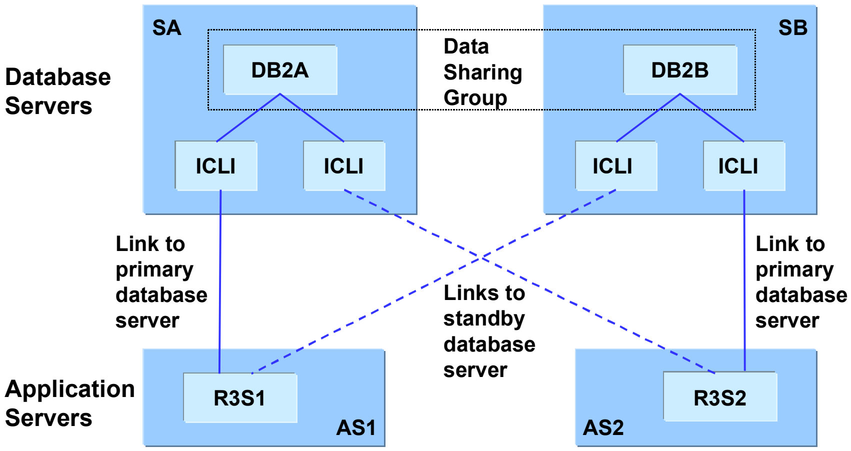
The reason for this is that there is currently no other way of making sure that the switched application servers switch back to their primary connections, other than stopping and restarting the ICLI server instance for the standby connections.

Sysplex Failover Support can be used in the following three scenarios.

Scenario 1: Failover into Surviving DB2 Member

Each DB2 data sharing member runs on a separate OS/390 system. It is used as the primary database server for one application server and as standby for another application server. There should be at least two ICLI servers for each data sharing member: one for the primary connections and the second for the standby connections, as described above.

If DB2A fails, the R/3 work processes of AS1 reconnect to DB2B. Then DB2B has to handle the workload of the application servers AS2 and AS1. Using this option requires that the data sharing member DB2B has enough free DB2 capacity and OS/390 capacity to handle the additional workload of the failed subsystem DB2A.

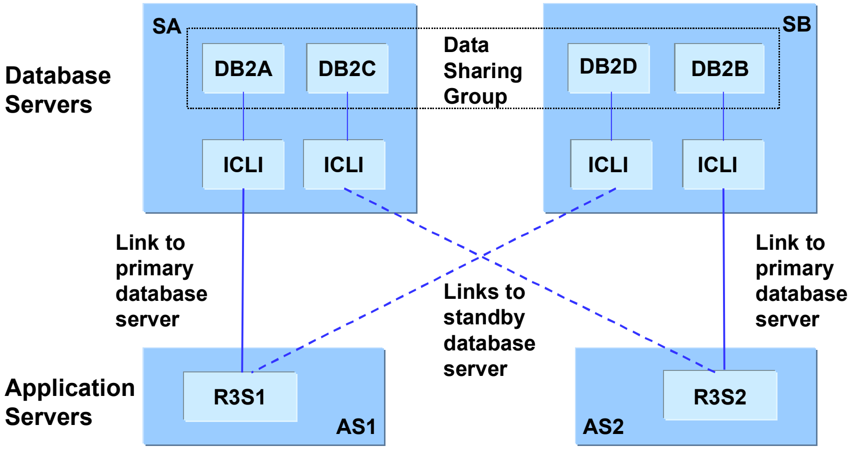


Scenario 2: Failover to Hot Standby DB2 in Same LPAR as Surviving DB2(s)

This failover scenario should be implemented if a data sharing member cannot handle the additional workload of a failed over application server because not enough DB2 capacity, such as virtual memory, is available. In this scenario each DB2 subsystem is used either as primary database server or standby database server but not as both. The standby DB2 subsystem is running on an OS/390 system where primary DB2 subsystems are running as well.

If DB2A fails, the R/3 work processes of AS1 reconnect to DB2D. Then DB2D only has to handle the workload of application server AS1. Using this option requires that the OS/390 system SB has enough capacity, such as CPU and memory, to handle the workload of both data sharing members.

This scenario requires slightly more memory and CPU than Scenario 1 because of the **additional DB2 subsystem**. For a sizing of the additional memory and CPU requirements, contact your IBM ERP Competence Center or your SAP/IBM support group.

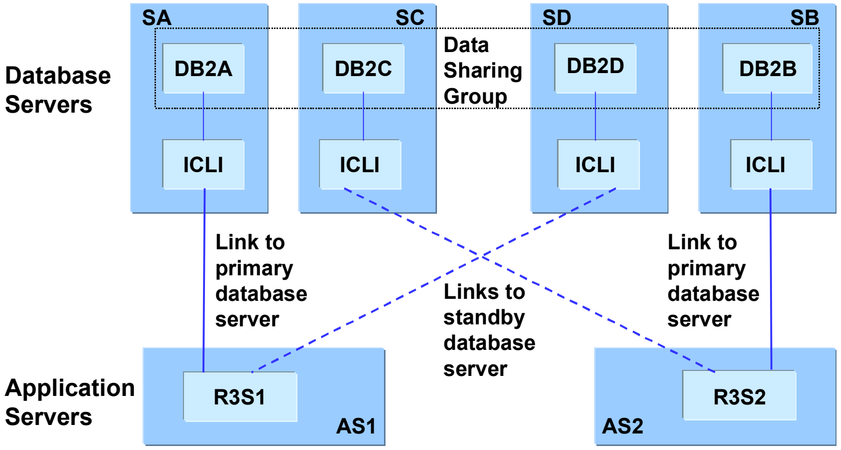


Scenario 3: Failover to Hot Standby DB2 in Hot Standby LPAR

This failover scenario should be implemented if problems could occur because not enough OS/390 capacity (such as CPU and memory) is available. Each data sharing member is on a separate OS/390 system and is used either as primary database server or standby database server but not as both.

If DB2A fails, the R/3 work processes of AS1 reconnect to DB2D on SD. DB2D only has to handle the workload of application server AS1.

This scenario requires more memory and CPU than Scenario 2 because of the **additional OS/390 system**. For a sizing of the additional memory and CPU requirements, contact your IBM ERP Competence Center or your SAP/IBM support group.



Integration

You can fully integrate DB2 Data Sharing and R/3 Sysplex Failover Support with your R/3 System.

Activities

The way in which an R/3 System using Data Sharing for DB2 for OS/390 recovers, differs according to the type of failure.

Failure of ICLI Servers

1. The application servers that were connected to the failed ICLI server instance automatically reconnect to their standby database servers and continue working. R/3 transactions might be rolled back, depending on the context of the R/3 user. If multiple ICLI server instances were running with a DB2 subsystem, and only one of them failed, this results in only a minor shift of workload from one database server to another.
2. After the failed ICLI server instance is restarted (automatically or manually), you can switch the failed over application servers back to their primary database server.

Failure of DB2 Subsystem

1. The application servers that were connected to the failed database server automatically reconnect to their standby database servers and continue working. R/3 transactions might be rolled back, depending on the context of the R/3 user.
2. The ICLI server instances that were connected to this DB2 subsystem clean up their connections to the application servers and to DB2 and prepare to receive new connections.
3. The failed DB2 subsystem is automatically restarted and performs restart processing by using the recovery log. Non-committed transactions are rolled back and locks are released. After that, the DB2 subsystem is again ready for work.
4. You can now switch the failed over application servers back to their primary database server.

Failure of Entire OS/390 System or Entire S/390 Hardware

1. The application servers that were connected to the failed database server automatically reconnect to their standby database servers and continue working. R/3 transactions might be rolled back, depending on the context of the R/3 user.
2. The DB2 subsystem that went down because of the failure and the relevant ICLI server instances are automatically restarted on another OS/390 system. The DB2 subsystem performs restart processing by using the recovery log. Non-committed transactions are rolled back and locks are released.
3. Once restart processing of the recovered DB2 subsystem has finished, the database server is available for work again, now running on another OS/390 system.
4. You can now switch the failed over application servers back to their primary database server.

Network Failure

The best protection against network failures is duplication of network adapters as described in Server Network in a DB2 for OS/390 Environment [Page 119]. If this is not possible, the application servers still attempt to fail over. Depending on where the network failure was, they might even get through to their standby database server.

Comprehensive Microsoft SQL Server High Availability Solution

Use

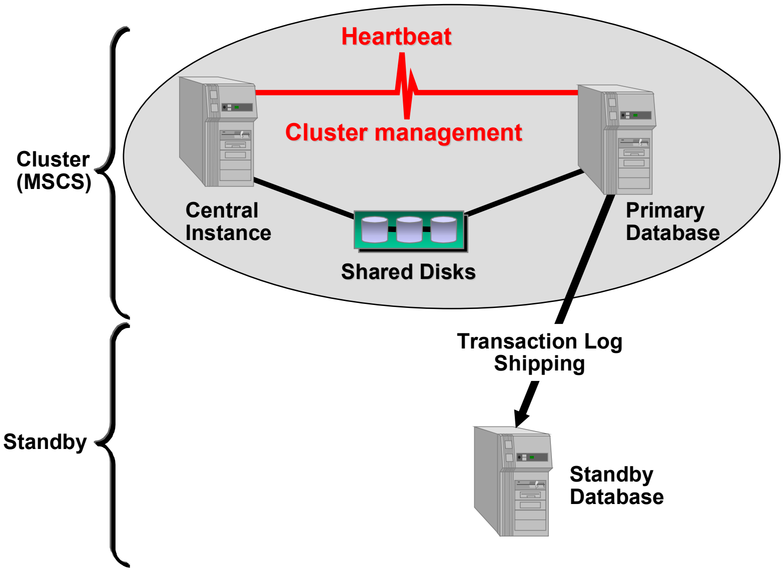
You can combine the following products from Microsoft to achieve a comprehensive high availability solution:

- Microsoft SQL Server Standby Database [Page 206]
- Microsoft Cluster Server (MSCS) on Windows NT [Page 220]

For more information, use the alias “HA” in SAPNet and look in the media center.

Integration

Comprehensive Microsoft High Availability Solution with Cluster and Standby



Features

The following table compares the standby database with MSCS:

	Strengths	Weaknesses
Standby Database	• Physically remote units • Separate and distinct SQL server and database instances • Separate disks and other key system resources • Good disaster recovery solution • Standby node can be used for read-only access	• Requires manual intervention following failure • Not automatic • Slow recovery following failure
MSCS	• Rapid failover • Automatic failover • Good solution for “frozen” operating system or application	• Shared disk is single point of failure (SPOF) • Not normally geographically separate • Transaction rollback to point-in-time not provided • Spare resources kept in “waiting” status

Switchover Software

Purpose

This section describes switchover software, an advanced technology that has been developed to improve systems availability for the hardware and system software that support the R/3 System in environments where host machines are grouped in clusters. As the name switchover implies, services can be automatically switched from a failed host to a standby host in the event of failure, allowing continuation of R/3 System operation. This section focuses on the switchover technology that is available for the R/3 services to provide resilience in the event of host machine failure.

Implementation Considerations

Switchover software is inherently complex. Therefore, for detailed technical guidance when implementing a specific product or feature, be sure to contact the appropriate source, such as your SAP consultant, the SAP Competence Center, and so on.

Integration

High availability for the R/3 System should be part of a system-wide strategy. Therefore, you should also consider all components of the system, including the R/3 System itself, the database management system (DBMS), the network, and so on.

Features

General aspects of switchover software in relation to R/3 Systems are discussed, mostly in terms relevant to both UNIX and NT operating systems. For a more detailed technical discussion of the use of switchover products for the R/3 System, see the following:

- *R/3 in Switchover Environments*, relevant to UNIX (see below for how to find this documentation).
- *R/3 Installation on Windows NT* (choose the version for your database)

See also:

R/3 in Switchover Environments (in SAPNet)

Switchover Products for the R/3 System

Use

This section summarizes the available switchover products for the R/3 System, looking at general features, product range, architecture, and functionality.

Switchover products protect system services by switching them over to standby resources in case a critical resource fails. These products address the single points of failure in hardware that can not be protected by standard technology (such as hot pluggable RAID, UPS, backup power supply, and so on). If you use switchover products in conjunction with the standard technologies discussed elsewhere in this documentation, you can substantially improve the availability of your R/3 System by comprehensively covering its single points of failure.

Switchover products offer a certain level of automation in monitoring the health of system components as well as in the detection of and reaction to component failures. Switchover software clearly cannot guarantee “zero downtime.” However, switchover products can limit the impact of host machine failures to your R/3 System and restrict its unplanned downtime to tolerable levels.

Switchover products allow the definition of highly available cluster systems, which are defined as a number of loosely coupled hosts with shared disks. For more information, see Cluster Technology [Page 130]. In general, switchover products are capable of monitoring and controlling different system resources such as host machines, network adapters so on. In the event of failure, the service offered by the resource is automatically taken over by a standby resource.

This section focuses on how to use switchover products to protect the R/3 System against failures of host machines (such as power supply failure, CPU failure, board failure), which are of key importance to the availability of your R/3 System.

Switchover products can be:

- Part of the operating system
- Closely attached to the operating system, but not actually part of it

Example

For the NT operating system, Microsoft offers the Microsoft Cluster Server (MSCS), an example of a switchover product that is part of the operating system. SAP supports this product, known as "R/3 Cluster on Windows NT," as the standard switchover solution for NT. For more information, see *R/3 Installation on Windows NT* (choose the version for your database).

Examples of the second type of switchover product – that is, closely attached to the operating system but not actually part of it – include the following, available for UNIX, NT and IBM platforms (in alphabetical order):

- COMPAQ On-Line Recovery Server
- DEC TruCluster Available Server
- HP MC/ServiceGuard
- IBM AS/400 Cluster (from Version 4.R4)
- IBM DB2 Tracker, DB2 Geoplex, DPropR for Replication
- IBM DB2/390 Sysplex Failover
- IBM HACMP for AIX
- NCR Lifekeeper Fault Resilient Systems
- SEQUENT Dynix ptx/CLUSTERS
- SIEMENS Reliant Monitor Software (RMS)
- SUN Cluster

This is not a complete list. It does not intend to express the significance of the products mentioned, nor their tested compliance with the R/3 System. If you have further questions on specific products please contact your hardware vendor for details, since this discussion is general (individual products are not described).

Activities

SAP is currently working with suppliers of switchover software to set up a procedure for testing their products for compliance with the R/3 System. SAP provides both detailed technical guidelines and a list of tested switchover products for R/3 to its customers. Refer to the documentation *R/3 in Switchover Environments* in SAPNet.

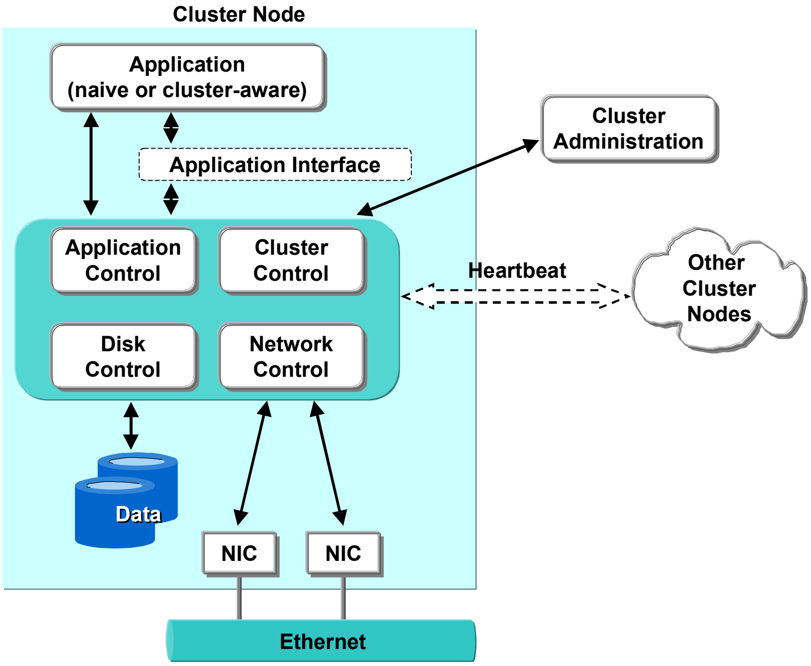
At the center of switchover products are one or more software components, usually called the “cluster manager.” The cluster manager establishes a “heartbeat” between the cluster nodes, which is used diagnostically to decide whether a network link or a node has gone down. The cluster manager might also monitor other local resources and take appropriate actions in the event of failure. However, this section focuses on host machine failures and corresponding switchover solutions.

In the R/3 System, all data is stored on a central database. Disk sharing is therefore necessary to make switchover consistent and transparent for the end user. The normal implementation for this is a twin-tailed SCSI bus, where each connected node has its own SCSI interface (so using an additional SCSI-ID). Some products also allow the usage of a proprietary disk sub-system. Shared disk access is controlled by an additional component of the switchover software. Access is usually exclusive to one of the nodes, and accordingly has to be switched with any R/3 service switchover.

Apart from the surveillance of network links by a heartbeat, the network adapters are also controlled. Switchover products enable configuration of an additional network adapter as a “standby” for the primary adapter. If the primary one fails, the standby adapter – which does not have an IP address – takes over the IP address of the primary. For the purpose of increasing the redundancy in the communication link between the cluster nodes, you can connect the two network cards in each node to separate physical network hardware.

The diagram below shows in simplified form the basic logical components of a switchover product, showing a single cluster node and the connections to and from it. Note that the diagram does not imply a single cohesive process. For example, disk control is usually part of the operating system (for example, a logical volume manager) and part of network control might be firmware in a physical device or might also be located in the operating system (for example, programming of MAC address).

Switchover Product Components



Apart from the physical configuration of the switchover environment, the most difficult part of the set-up is the definition of the actions that are necessary to properly switch an application over to the standby resource.

Some switchover products pre-define events (for example, `node_down`, `network_down`) as well as rules and actions for the cluster manager to react to these events. The actions are normally defined as a sequence of commands in a simple shell script. These actions include some “generic” system commands (activate shared disks, check and mount filesystems) and application-specific commands (start database, move some files).

With R/3 Cluster on Windows NT, SAP delivers a pre-configured solution. Therefore, you do not need to define switchover events, rules, or actions.

All switchover products allow management of the cluster with system commands or with more user-friendly interfaces. There are basic commands for cluster management to perform the following tasks:

- Configure the cluster and its nodes
- Start and stop the cluster
- Add nodes to the cluster
- Switch applications manually to another node (for node maintenance)
- Query the status of the cluster

Manual switchover is useful to make cluster nodes available for maintenance (that is, to allow planned downtime).

This documentation and the current R/3 solutions focus on node failures. For more information about the use of switchover products for the R/3 System, see the SAP documentation *R/3 in Switchover Environments* (see below).

See also:

R/3 in Switchover Environments (in SAPNet)

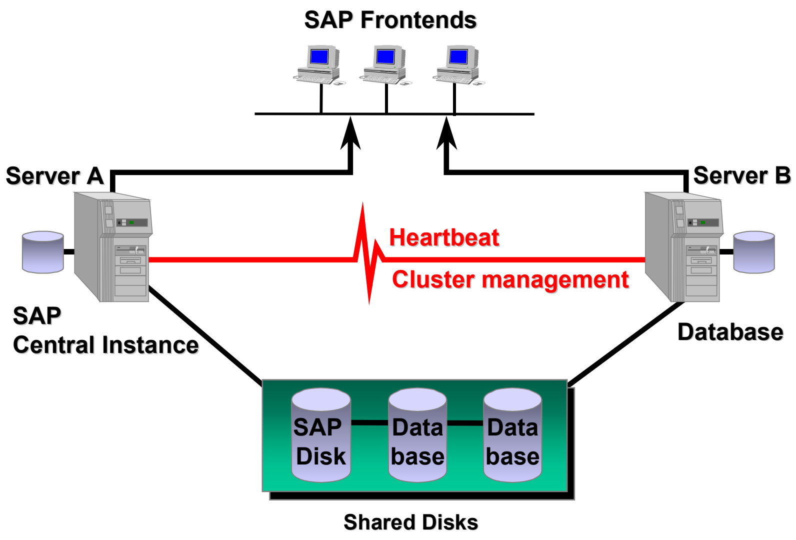
Microsoft Cluster Server on Windows NT

Use

The Microsoft Cluster Server (MSCS) configuration is the standard switchover solution for R/3 running on Windows NT. In this configuration, the R/3 System is installed on two nodes of a cluster. Under normal operation, the R/3 central instance runs on one node and the database on the other node of the cluster. If one of the nodes fails, the affected central or database instance is automatically moved to the other node, so preventing downtime. The failover mechanism is enabled by the MSCS software and a cluster-aware version of the database management system (DBMS).

The aim of the MSCS configuration is to increase the overall availability of the R/3 System by replicating single points of failure – that is, the database and the central instance. The cluster is normally fully transparent to clients accessing MSCS, although there might be a delay in the event of failure due to failover.

MSCS Configuration



Integration

The capability to fail over resources in the cluster is enabled by the MSCS software in combination with a cluster-aware version of the database management system (DBMS) and an appropriately configured R/3 System. To set up R/3 on a MSCS configuration it is necessary to perform a new installation of the system following special instructions provided for MSCS.

It is adequate for most systems to run the database and central instance in a single cluster. However, for large systems, it is better to set up a cluster for the database and a separate cluster for the central instance. The disadvantage of separate clusters is increased complexity of administration.

For more information on how you can combine MSCS with the Microsoft SQL Server Standby Database [Page 206], see Comprehensive Microsoft SQL Server High Availability Solution [Page 214].

Prerequisites

To use R/3 Cluster on Windows NT, you must meet the following prerequisites:

- Enterprise Edition (EE) of SQL Server for Windows NT
- Hardware certified by R/3 and approved by Microsoft for MSCS
- Cluster-aware DBMS
- R/3 data stored on RAID1 and RAID5 disks

For more information, see the installation documentation for the R/3 System.

SAP recommends you **not** to use “Active-Active” configuration of MSCS, in which two SQL Servers coexist on the same cluster.

Features

With R/3 Cluster on Windows NT, SAP offers a standard high availability solution for the R/3 System on all hardware platforms supported for NT. To achieve this, SAP uses the cluster capabilities of MSCS.

Performance during normal installation is comparable to a distributed installation but can drop significantly after failover.

High availability in MSCS operates on the following levels:

- Hardware replication

  Since the server is physically replicated, all internal machine components are replicated. This includes motherboard, memory, Disk controller, power supply, NIC, bus, and so on. If any of these fail, the system can fail over and still continue functioning.

- Automatic failover

  Failover occurs rapidly without manual intervention, so no time is lost manually working out the cause of the problem and deciding how to fix it.

Activities

During operation of R/3 Cluster on Windows NT, MSCS monitors the system for failure and switches over services in the event of a failure. Both monitoring and switchover occurs automatically, that is, without operator intervention. However, the system administrator is still responsible for solving the problem that led to failure (for example, replacing faulty hardware components).

See also:

The alias “HA” in SAPNet, then look in the media center

R/3 Installation on Windows NT (choose the version for your database)

Cluster Server Concepts and Architecture (see the Microsoft web site)

Interfacing Switchover Technologies with R/3

This section looks at how you can make the R/3 System compatible with switchover technologies. This means solving the central problem of network addressing. See Switchover Addressing Example [Page 225] for an example of the principles discussed here. For more information about R/3 networks, see Network System Key Issues [Page 108].

The discussion in this section is important for an understanding of how the R/3 System is implemented with switchover software. For a more detailed discussion, see *R/3 in Switchover Environments* (for how to find this paper, see the reference at the end of this section).

⚠

You must follow the guidelines in this documentation and in the paper *R/3 in Switchover Environments* to guarantee a problem-free integration of the R/3 System with switchover software. Suppliers of switchover software can test their products using the test routines described in *R/3 in Switchover Environments*. SAP and its partners have tested in this way the standard SAP high availability solution for the Microsoft Windows NT platform, R/3 Cluster on Windows NT.

Basics of TCP/IP Addressing

Transparent network addressing is the key to successfully running and maintaining the R/3 System in a switchover environment. Although there is no explicit R/3 software interface for switchover products, you should think of R/3 address configuration as the critical point where both products are interconnected.

Service and Host Addressing in R/3

R/3 inter-process communication across the network (in some cases also locally within a host machine) is based on TCP/IP. Two pairs of IP address/port number are necessary to uniquely identify each communication channel on a network. The port number identifies a given service (or process) on a specified host. The IP address is used to specify the host machine (or at least one network adapter of this host). It is the task of the operating system to make sure that addressing is properly handled below the IP level. Addresses must be converted to the protocol that is run by the physically underlying hardware.

Host Names

IP addressing is made user friendly by providing logical host names to address machines on the network. Logical host names are mapped to IP addresses through an address database, which is either local (`/etc/hosts` file) or network-wide (distributed name services like DNS, NIS, and so on).

There is also the concept of a local hostname for a given host machine, as returned by system calls or operating system commands (for example `hostname, uname`). This local scheme uses a namespace that is different from the networking one. Local hostnames and logical hostnames on the network are often set up to match, but, in fact, do not necessarily have to do so.

Transparent Addressing and Switchover Technologies for R/3

Switchover software is able to handle the migration of R/3 services to different host machines after a hardware failure has occurred. These changes in the actual, physical location of a given

R/3 service within a cluster-based client/server environment must be made as transparent as possible to the attached clients (for instance dialog users). This is one major request that switchover software has to serve. The straightforward way to achieve this kind of transparency is to keep the address, where a service can be reached, constant under all circumstances.

The R/3 System can be configured to support two different major techniques (formerly called “modes”) to achieve this functionality:

- Identity takeover
- Virtual IP address takeover

The advantage of both implementations to R/3 is that all address information held in R/3 profiles or in the R/3 database can be left unchanged. Only MAC addresses and hardware identification numbers need to be changed.

Identity Takeover

The standby host takes over the complete “identity” of a failing node, by taking over the local hostname and an IP address. Note that it must be the IP address that is used by the clients, to keep addressing transparent.

The implementation method for enabling identity takeover varies with different software vendors. Some identity takeover schemes take over the root partition of the failing node and boot from this device (standby reboot). The main advantage of an identity takeover is that it is considered to be a “safe” switchover because it guarantees consistency of all addresses involved. There are also solutions that do not require rebooting. If the local hostname of the standby node must be changed during these types of switchover, you must make sure that no other software is currently using this information.

Virtual IP Address Takeover

Switchover products – especially on UNIX platforms – mostly use the concept of virtual or relocatable IP addresses. This in effect inserts something like another logical layer within the IP protocol. The virtual IP address is able to migrate between different host machines on the cluster, while clients (on the same host, on the cluster network or on the external network) transparently access a constant virtual address.

External clients are made independent of the physical location (host machine) of a particular service and can reattach to it using the same logical address as before. It is the task of the operating system and the switchover software to properly maintain the mapping of IP address to the underlying physical hardware, for example updating ARP translations in the case of Ethernet.

Virtual IP addresses provide the advantage that clients do not have to be restarted if an automatic reconnect is offered by the application. R/3 fully supports virtual IP address takeover and also includes automatic reconnect features.

Specific support from the operating system is needed to implement virtual IP addressing. Several possibilities exist, for instance:

- A second network adapter (“Network Interface Card”, or NIC) is configured to hold the virtual address
- Stacks of IP addresses on one adapter

Address Parameters in Switchover Environments

The different types of IP addresses and hostnames that can be found in switchover environments are shown in the table below. This table also provides a definition of the terminology used by SAP:

Types of Addresses Used with TCP/IP in an R/3 Switchover Environment

Address/Name	Explanation
Stationary IP address	Standard IP address configured in a stationary way on a NIC
Virtual IP address	IP address that can be moved between the cluster nodes
Stationary hostname	Name that maps to the stationary IP (via the network address database)
Virtual hostname	name that maps to the virtual IP address (via the network address database)
Local hostname	Host name returned by system call or operating system command
AppServer name	Name used internally by R/3 to address a specific application instance (for example, `<...hostname>_<SID>_<NR>`)

Note the following points concerning the address definitions:

- Usually local hostname and stationary hostname are configured to be equal, but do not necessarily need to be.
- Either the local hostname or the virtual hostname can be part of the AppServer name, depending on the specified configuration.
- For information about where and how to use these addresses in R/3 profiles, see the SAP documentation *R/3 in Switchover Environments* (for information on how to find this documentation, refer to “See also” at the end of this section).
- For an example of the principles discussed in this section, see Switchover Addressing Example [Page 225].

Set-Up of Network and Addressing

The real task of network and address configuration is a demanding issue. One complication is that several networks are used to split the heavy load of the server network, from the lighter traffic on the access network used for frontend communication. See the diagram in “Switchover Scenarios 1 and 2: A Typical Set-up” in Principal R/3 Switchover Scenarios [Page 235] for an example using separate server and access networks. Several topologies and set-up options are possible, but all of them require careful configuration of the network, SAP profile parameters and network routing. For more information, see the documentation *R/3 in Switchover Environments* (see below)*.* For an example, see Switchover Addressing Example [Page 225].

See also:

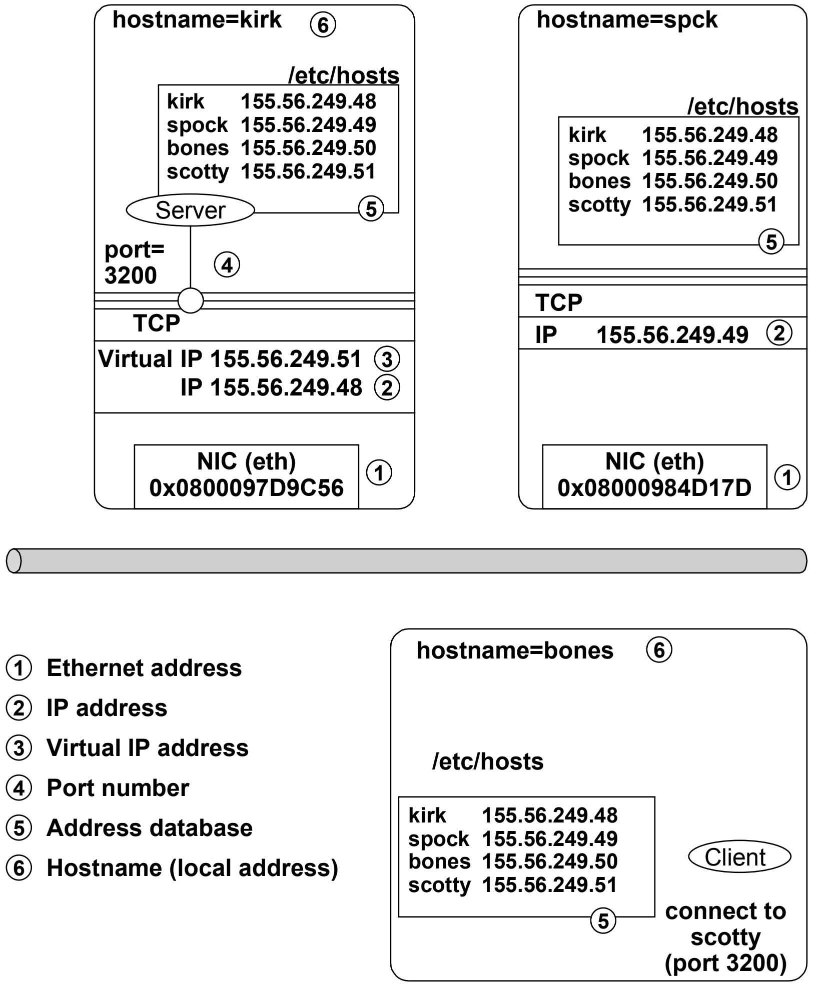
R/3 in Switchover Environments (in SAPNet)

Switchover Addressing Example

The example in this section of addressing in switchover environments is simplified and assumes a single-network cluster. The example illustrates the basic principle of how virtual addressing works in a switchover cluster.

Host `kirk` uses the virtual IP address `155.56.249.51` and is addressed by clients running on host `bones` as `scotty`. If host `kirk` fails, host `spock` would take over the service and assume the virtual IP address `155.56.249.51`. Host `bones` could still use the address `scotty` although the actual service would have migrated in the meantime (the switchover might require the client to initiate a reconnect). When activating the virtual IP address on a standby node, the switchover product guarantees address consistency on lower network layers by refreshing the local address resolve protocol (ARP) caches on the network.

Example of TCP/IP Addressing in a Switchover Environment



Protecting R/3 SPOFs by Switchover Software

To maximize the availability of any system, it is necessary to protect its single points of failure (SPOFs). The following table gives an overview of the critical services and SPOFs in a standard R/3 System:

R/3 Critical Services and Single Points of Failure (SPOFs)

R/3 Component or Service	Number of Redundant Units	System-Wide SPOF?
Database service (DB)	Exactly 1 per R/3 System	Yes
Enqueue service	Exactly 1 per R/3 System	Yes
Message service	Exactly 1 per R/3 System	Yes
Dialog service	Configurable: 1... n per APP	No
Update service	Configurable: 0... n per APP	No
Batch service	Configurable: 0... n per APP	No
Spool service	Configurable: 0... 1 per APP	No
Gateway service	Exactly 1 per APP	No
SAPCOMM	Configurable: 0 or 1 per APP	No
SAPROUTER (WAN access)	Configurable: 0... n per R/3 System	No
NFS service	Exactly 1 per R/3 System	Yes

The term “APP” refers to “application host”. The term “NFS” refers to “network file system”. For further information, see “Architectural Rules for Mapping” in Mapping of R/3 System Services [Page 14] and R/3 System Failures [Page 27].

This table shows that the failure of a particular host machine might mean that an entire R/3 System is unavailable. You can approach this problem in a twofold manner:

- Critical centralized R/3 services (that is, DB, enqueue, message and NFS)

  A failure of the host machine supporting any one of these services leads to R/3 System unavailability and such services can not be distributed for redundancy in a standard R/3 System. Therefore, to minimize unplanned downtime you should protect all these services either with a switchover solution or with the extended enqueue service [Page 232].

- Remaining R/3 services

  The other services – which are not system-wide SPOFs – can be set up redundantly in the R/3 System, that is, distributed and mapped onto several application hosts. Refer to “Mapping R/3 System Services For High Availability: General Recommendations” in Mapping of R/3 System Services [Page 14]. This in effect provides these services (on a system-wide level) with sufficient failure resilience since these services form virtual clusters. With appropriate distributed configuration to make sure of redundancy, you do need to further protect these services with switchover software.

Distribution of R/3 Switchable Services on Switchover Clusters

To run the R/3 System in a switchover environment, SAP recommends that you map the R/3 services constituting system-wide SPOFs into just two switchover packages:

- CI – central R/3 instance (containing both enqueue and message services)
- DB – database service

Note that splitting CI and DB is a general recommendation for running R/3 in a multi-host environment with heavy database workload. It is an option inherent in the three-tiered architectural layout of R/3 and absolutely independent of any switchover considerations, with the advantage that the dedicated DB host is freed from the CI workload. For more information, see Mapping of R/3 System Services [Page 14].

Mapping the enqueue and message services onto a single host minimizes the overhead of inter-process communication because the enqueue service is always accessed via the message service in a standard R/3 System. Note that the database and NFS (network file system) services are separate packages of their own in any case already.

The R/3 System requires the functionality of NFS to provide its own service properly because several R/3 directories need to be put on NFS, so necessitating a third switchable package, in addition to those for CI and DB. However, note that from the R/3 perspective, NFS functionality is in some respect part of the operating system. It therefore needs to be handled by the switchover product independently of the R/3 System. In most cases, either the database node or the central instance node acts as the host for the NFS service (a particular set-up might be forced by DB requirements).

Several choices for distributing these three switchover units exist, depending on how many host machines are available in a switchover cluster. The following table gives examples of the different ways to distribute these service units across a set of two or three cluster hosts in a switchover environment.

Distribution of Critical R/3 Services in a Switchover Cluster

Cluster Node 1	Cluster Node 2	Cluster Node 3	Usage
DB / CI / NFS	Idle	Not used	Central system with idle standby host
DB	CI / NFS	Not used	Mutual takeover – NFS on CI node
DB / NFS	CI	Not used	Mutual takeover – NFS on DB node
DB	CI	NFS	Fully distributed – dedicated host for NFS service

Therefore, SAP makes the following basic recommendations for installing the R/3 System in a cluster environment:

Install DB, CI, and NFS on cluster nodes

Since it is essential to protect the SPOFs in an R/3 System, most installations install the database service (DB), the central instance (CI) and the network file system (NFS) on cluster nodes. The CI at least contains dialog work processes, the enqueue work process and the message service. Most installations leave the default CI with all further services configured as well (that is dialog, update, batch, spool, and gateway).

CI should also have NFS

The central instance (CI) should also be the location for the network file system (NFS), since all the NFS mounted filesystems contain R/3 instance files.

Impact of Failure and Switchover of R/3 Key Services

Using switchover software for protecting the system-wide components increases the availability of the R/3 System in general. However, the functionality that can be offered by the R/3 System directly after switchover of a particular component depends on the type of unit or service, CI or DB, that has been switched. Accordingly two major types of functionality can be distinguished, depending on the type of switchover, as described below.

CI Switchover – Losing Enqueue Locks

This section refers to the standard enqueue service. For more information on advanced solutions, see Protection of the Enqueue Service [Page 229].

If the machine hosting the enqueue service (that is, the machine with the CI) fails, all SAP locks for transactions that have not yet been committed are lost. This is because R/3 locking information is maintained exclusively in the memory of the host machine supporting the enqueue service (predominantly for performance reasons). This statement is valid for all current releases of the R/3 System.

The R/3 System guarantees that no user can perform a transaction as long as the host supporting the enqueue service (that is, the CI in the definition of switchable units given above) is not available. This implies that database consistency is guaranteed in the event of enqueue (or CI) failure. To guarantee database consistency after switchover or restart, all open transactions in the system must be aborted and rolled back before the enqueue service (that is, on the CI) is restarted.

For releases prior to R/3 3.0E it is absolutely necessary that the switchover scripts (or the administrator) make sure that all R/3 application hosts are stopped and restarted. This action must be taken to properly reset all enqueue locks on the entire R/3 System. This point is absolutely critical to R/3 database consistency and you must make sure that you follow it. CI failure and switchover impacts the R/3 System as follows:

- R/3 locks for transactions that have not yet been committed are lost on a system-wide level.
- All attached R/3 instances must be stopped and restarted after CI restart. Memory buffers inside application hosts are scratched, so overall system performance is less than optimal until the buffers return to acceptable hit ratios.
- All users of the R/3 System lose their sessions and contexts, so all users must log on again manually.
- All user input for all transactions that have not been finished by a `COMMIT WORK` (ABAP command) must be re-entered.

Starting with R/3 release 3.0E the system is capable of handling the important task of resetting open transactions automatically (within certain limitations). Enqueue locks are reset on a system-wide level when the enqueue work process is restarted on the CI. Failure of the CI followed by switchover impacts the R/3 System as follows:

- R/3 locks for transactions that have not yet been committed are lost on a system-wide level.
- All user input for all transactions that have not been finished by a `COMMIT WORK` (ABAP command) needs to be re-entered.

Starting with R/3 Release 4.5A, the resetting of open transactions is much improved and fully automated. See "Enqueue Service Failure" in R/3 System Failures [Page 27].

DB Switchover – DB Reconnect

This is the second major type of switchover functionality. In the event of DB host failure or DB switchover, the network connection of R/3 work processes to the database service is lost. The DB reconnect functionality makes sure that the work processes of the CI and all APP instances are automatically reconnected as soon as the DB instance gets restarted and the DB service becomes available again. If configured correctly, all users remain connected to the system and all memory buffers are preserved.

To the end-user the temporary unavailability of the DB service is almost fully transparent, apart from the waiting time for the DB service to be switched over and become operational again. Note however, that the exact functionality offered depends on the type of access service (dialog, batch, update) concerned. For more information about this, see DB Reconnect [Page 174] or SAP Note 24806.

Note that DB reconnect functionality is offered in R/3 releases from 3.0A onwards. It has been tested only for a restricted set of database systems.

Protection of the Enqueue Service

Use

The enqueue service (that is, the R/3 lock manager) is a system-critical component of the R/3 system. The enqueue service consists of the enqueue server process and enqueue table. The enqueue table is held in the main memory of the host. This means that, in the event of an enqueue service failure, locking information is lost. Therefore, user transactions and background processes have to be aborted before the R/3 System can resume operations.

Since in a standard R/3 System there can be at most only one enqueue work process for each R/3 system, enqueue service failure normally causes the entire system to stop functioning until it can be restarted or, in a clustered environment, moved to another server.

To address the enqueue server's vulnerability you can take the following approaches:

- Standard R/3 System with no additional features

  Fix the cause of the failure and restart the enqueue service. Dialog users must re-enter all interrupted transactions and you must resubmit all interrupted batch jobs after the enqueue server is restarted.

- Standard enqueue service with switchover software and cluster solution

  This solution uses switchover software to transfer the enqueue service automatically from the failed to the surviving server in the cluster. Therefore, recovery time is significantly reduced, because no time is lost fixing the failed server. However, there is a short service interruption for the switchover.

  Although this approach is an improvement on the standard setup, it does **not** eliminate the enqueue server as a single point of failure (SPOF). For more information, see "CI Switchover – Losing Enqueue Locks" in Protecting R/3 SPOFs by Switchover Software [Page 226].

- Extended enqueue service with replication

These solutions further protect the enqueue service as follows:

- Enqueue service with replicated lock table [Page 230]

  This approach uses the rapid cluster communication in a switchover environment to replicate the lock information from the primary to the secondary host. In the event of failure, the replicated lock information is used to rebuild the lock table on the secondary host. The rebuilt lock table is then used by the restarted enqueue server.

- Enqueue service with replicated enqueue server and lock table [Page 232]

  This approach is the most advanced, as it eliminates the entire enqueue service as a single point of failure. It consists of a fault-tolerant unit, in which two mirrored enqueue services replicate one another. If one enqueue service fails, the unit repairs itself transparently and continues functioning without interruption of the enqueue service.

  With this solution you do **not** have to stop user transactions or background processes. Therefore, a failure in one of the enqueue service hosts does not affect either of these.

  However, user transactions and background processes are still affected if they are running on the application host that fails. These must be restarted. However, this is **not** related to the failure of the enqueue server. This is necessary when any application server fails.

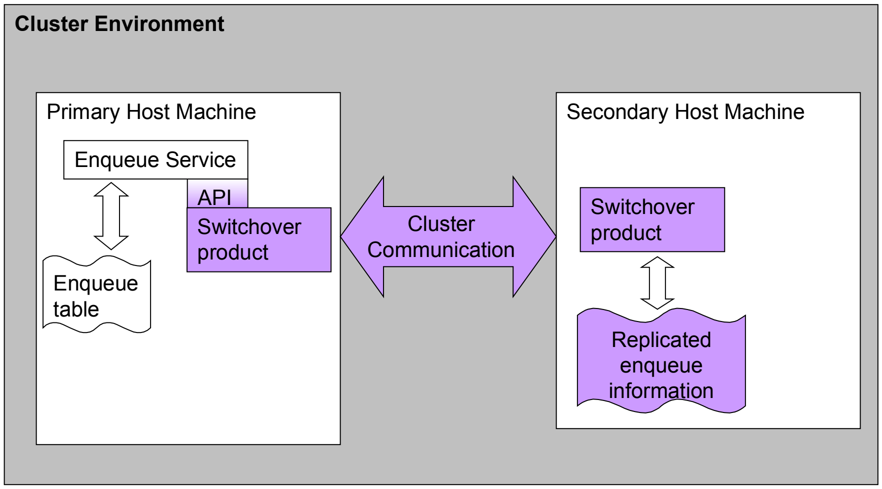
Extended Enqueue Service with Lock Table Replication

Use

Starting with R/3 Release 4.5A, you can use switchover software [Page 216] to replicate the enqueue locking information to the secondary machine, so allowing a fully transparent restart on the secondary machine in the event of enqueue service failure on the primary machine. Transactions that are already running do not need to be restarted and new transactions can also be locked as required. The result is significantly improved availability due to the automated recovery following enqueue service failure.

Compare this solution, which replicates only the enqueue service **locking information,** to Extended Enqueue Service with Full Replication [Page 232], which replicates the **entire** enqueue service. For more information about the enqueue service as a single point of failure (SPOF), see Protection of the Enqueue Service [Page 229].

Using Switchover Software to Replicate the Enqueue Service State



For more information on the new features of the enqueue service for installations **without** switchover, see "Enqueue Service Failures" in R/3 System Failures [Page 27].

Integration

An application program interface (API) driven by the R/3 System feeds information to the switchover product, which then replicates the locking information to the secondary host machine. The API is identical for all switchover products, but the implementation on the switchover side varies from product to product and is the responsibility of the relevant SAP hardware partner.

As in all switchover setups, it is the switchover software that replicates the required information and performs the necessary actions in the event of failure.

Prerequisites

- An approved switchover product for the R/3 System. See Switchover Products for the R/3 System [Page 216]. The switchover product must be able to communicate with the enqueue service API in the R/3 System.
- SAP has explicitly released this feature for your combination of operating system and database management system.
- R/3 System Release 4.5A or higher

If your R/3 System is older than Release 4.5A, you need to have updated it with the functionality from 4.5A. See R/3 Downward-Compatible Kernel [Page 44]. This is only possible with R/3 Release 3.1 and 4.0.

If in any doubt about whether you meet the prerequisites, contact SAP and your SAP hardware partner.

Features

- Improved availability for the R/3 System, because the enqueue table is no longer a single point of failure
- Simpler handling of error situations in the R/3 System

- Simpler integration of the R/3 System with switchover products (that is, the scripts required to handle failure are less complex than before)

See also:

R/3 in Switchover Environments (in SAPNet)

Extended Enqueue Service with Full Replication

Use

You can use the extended enqueue server and lock table replication to protect the enqueue service, which is otherwise a single point of failure (SPOF) in the R/3 System. This eliminates unplanned downtime caused by the failure of the application server running the enqueue service. For more information about the enqueue service as a SPOF, see Protection of the Enqueue Service [Page 229].

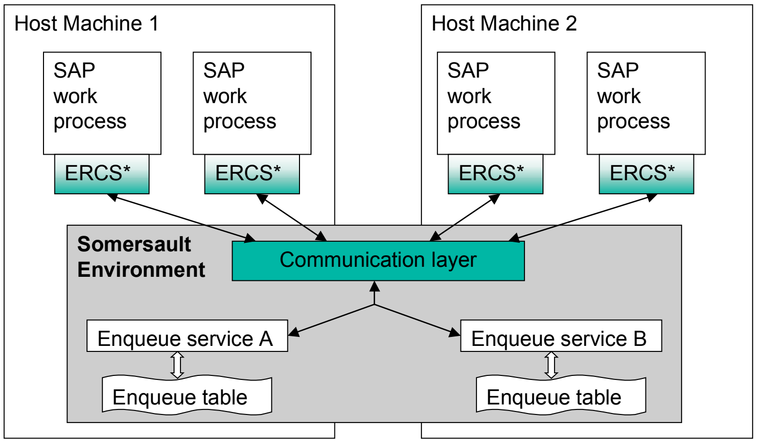
The extended enqueue service with full replication consists of an enqueue remote communication stub and the integration with a platform-specific fault-tolerant solution. SAP and HP offer the first complete solution of this kind for release 4.6. It is based on Somersault, the HP "process mirroring" technology, and is available on HP and Windows NT operating system (a Linux solution is planned).

Somersault provides fault tolerance through replication. It eliminates the enqueue service as a SPOF by using "recovery units," in which replicated processes mirror one another while executing in parallel. An independent witness unit monitors operations and ensures uninterrupted operation when a process fails. When a Somersault unit repairs itself after a process failure, no state is lost and no in-flight requests and responses are lost.

Compare this solution, which replicates the **entire** enqueue service, to Extended Enqueue Service with Lock Table Replication [Page 230], which only replicates the enqueue service **state**.

Integration

Somersault is integrated with the R/3 System as follows:



* Enqueue Remote Communication Stub

Somersault constitutes the middleware for transparent replication of the enqueue server and for communication of individual SAP work processes with the replicated enqueue server. The integration of the fault-tolerant middleware is based on an extension of the SAP work processes with the "enqueue remote communication stub." This enables the integration of HP Somersault with the R/3 System. The enqueue server requires no adaptation.

Prerequisites

Somersault is supported on HP-UX 11.0 and Windows NT 4.0 for R/3 Release 4.6.

For HP-UX 11.0 you can use Somersault with restricted Service Guard Configurations as follows:

- The R/3 database can exist on a Service Guard Cluster.
- The enqueue service is protected by Somersault, and all members of the machine set for the primary and secondary must reside outside the Service Guard cluster.
- The witness can reside on a Service Guard cluster.

For Windows NT 4.0 and later, you can use Somersault with a restricted set of MSCS (Microsoft Cluster Service) configurations as follows:

- The R/3 database can exist on an MSCS cluster.
- The enqueue server is protected by Somersault; all members of the machine set for the primary and the secondary must reside outside the MSCS cluster.
- The witness can reside on an MSCS cluster.

Support for Linux is planned.

Features

Somersault provides the following benefits:

- Eliminates enqueue as SPOF
- Minimizes data to be re-entered and batch jobs to be restarted following failure

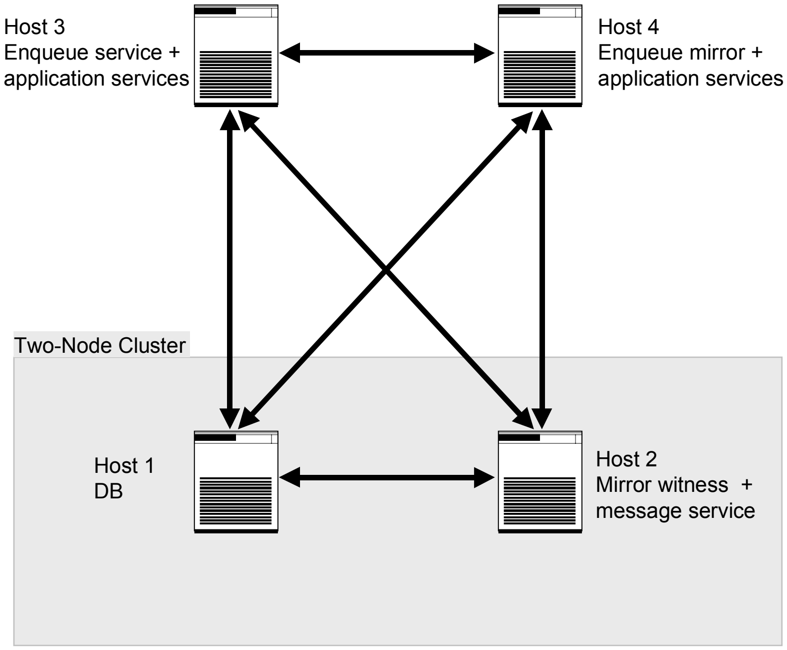
Extended Enqueue Service with Full Replication

- Is transparent to users and administrators

Activities

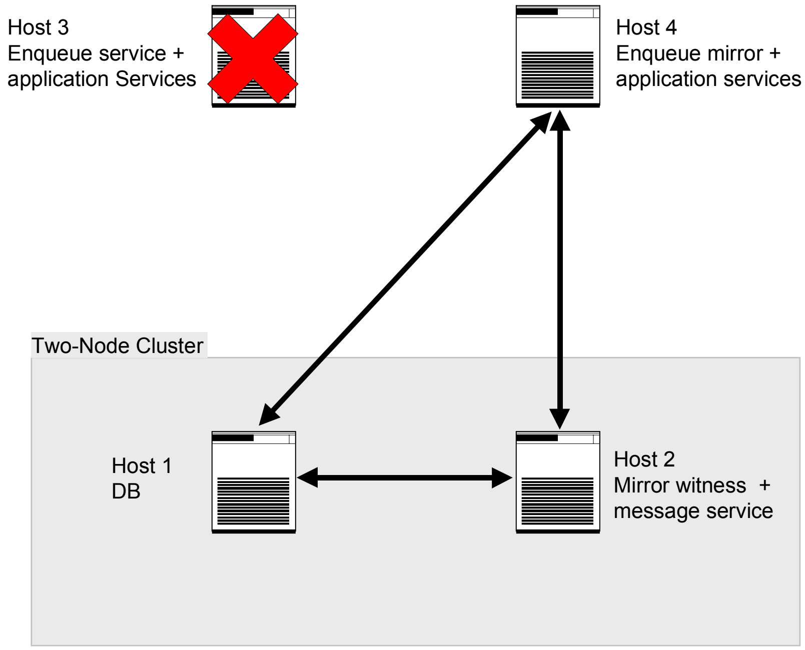
The following diagrams illustrate a typical configuration before and after failure of the enqueue server host:

Extended Enqueue Service Configuration Before Failure



The mirror witness monitors servers 3 and 4, which support the enqueue service and its mirror. Note that the enqueue server and its mirror are **outside** the cluster. The same operating system must be used for all servers.

Extended Enqueue Service Configuration After Failure



After failure, the enqueue mirror continues to provide the enqueue service to the SAP work processes.

Users on server 3 lose their sessions and are automatically routed to server 4 when they sign on again. They have to re-enter uncommitted transactions. You also have to restart batch jobs. However, this is **not** related to the failure of the enqueue server. This is necessary when **any** application server fails.

Contact SAP for more information on extended enqueue service configuration and HP for more information on Somersault.

Principal R/3 Switchover Scenarios

Since there is a well-defined, restricted set of switchable R/3 units, there is also a restricted set of principal switchover scenarios that are reasonable for the R/3 environment. For more information about switchable R/3 units, see “Distribution of R/3 Switchable Services on Switchover Clusters” in Protecting R/3 SPOFs by Switchover Software [Page 226] (SPOF refers to “single point of failure”).

This section focuses on the switchover of the DB (database) and CI (central instance) units of R/3 only, because NFS (network file service) can be considered to be part of the operating system. Therefore, the assumption is made that NFS switchover is handled transparently (that is, internally by the switchover product).

Any particular work process-based service (that is, dialog, batch, update, spool) that you want to centralize on one dedicated host should be configured as part of the CI on the switchover cluster (similar to the recommendation for the enqueue and message services given in Protecting R/3

SPOFs by Switchover Software [Page 226]). This avoids introducing another system-wide SPOF. However, you should generally aim to distribute services crucial to your business onto several host machines for greater redundancy: see “Mapping R/3 System Services For High Availability: General Recommendations” in Mapping of R/3 System Services [Page 14]. If you do this, you do not need to include hosts running non-critical R/3 services (that is, dialog instances) in the switchover scheme.

Switchover of R/3 units must not occur between different operating systems in heterogeneous system environments (that is, mixed NT and UNIX systems). For reasons of consistency, all cluster-external application hosts should be running the same operating system as the CI.

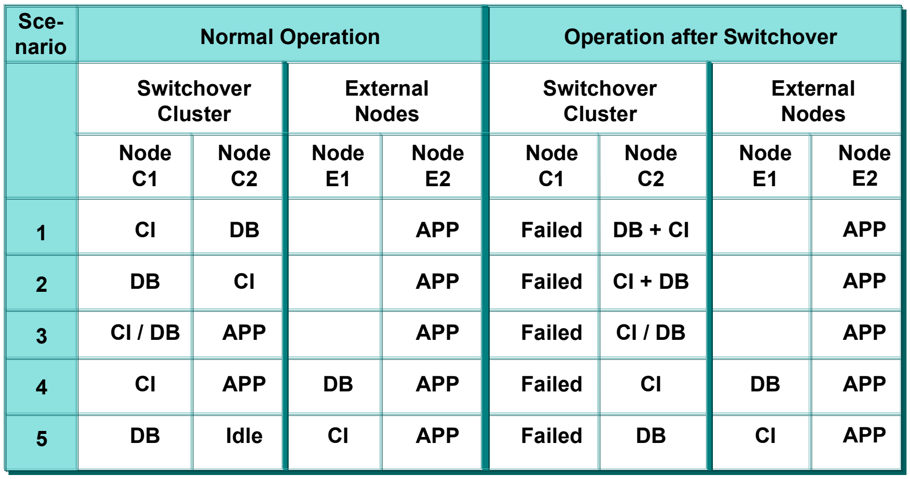
Switchover Scenario Definitions

A minimal hardware configuration for an R/3 switchover environment consists of:

- A two-node switchover cluster
- One or two additional non-clustered, “external” host machines (number of external hosts depends on the particular scenario).

Based on this set of hardware resources, the table below classifies the scenarios described below. These scenarios have been designed mainly to test switchover products for compliance with R/3 functionality. They can also be used as the basic building blocks out of which a customer switchover cluster can be constructed.

R/3 Switchover Scenarios



Scenario	Normal Operation				Operation after Switchover			
	Switchover Cluster		External Nodes		Switchover Cluster		External Nodes	
	Node C1	Node C2	Node E1	Node E2	Node C1	Node C2	Node E1	Node E2
1	CI	DB		APP	Failed	DB + CI		APP
2	DB	CI		APP	Failed	CI + DB		APP
3	CI / DB	APP		APP	Failed	CI / DB		APP
4	CI	APP	DB	APP	Failed	CI	DB	APP
5	DB	Idle	CI	APP	Failed	DB	CI	APP

Abbreviations: CI = Central instance
DB = Database service
APP = Application service
(also known as “dialog instance”)

The external node (that is, external to the switchover cluster) is connected to the cluster machines through the server network. The external host is not normally a member of the switchover cluster (although it can be, if required). Of course, all machines are connected to the access network and any frontend can connect to any instance on any host on the whole system.

The host machine E2 is included for test purposes in these scenarios only. It serves to prove the capability of handling connections from an external R/3 instance in switchover situations. You can omit the E2 node from your setup, if only two (scenarios 1, 2, and 3) or three (scenarios 4 and 5) host machines are available.

Scenarios 1, 2, and 3

For scenarios 1 and 2, the switchover cluster consists of two machines, with CI and DB services split between them. If node 1 fails, the failing service — either DB or CI — is switched to join the complementary service (either CI or DB respectively) on the second cluster node. See diagram and further description below in “Switchover Scenarios 1 and 2: A Typical Setup”. Note that scenarios 1 and 2 correspond to the setup used by R/3 Cluster on Windows NT [Page 220].

Scenario 3 offers a third option for switchover: CI and DB run jointly on one host machine. The second cluster node runs an optional application service (APP) that is shut down before CI/DB switchover (this node can be run as an idle standby if so desired). If the CI/DB node C1 fails the entire CI/DB (R/3 central system) is switched over to node C2 (the optional APP on node C2, if present, is shut down beforehand).

Scenarios 4 and 5

These scenarios are included to demonstrate switchover solutions for mixed operating system environments. Such environments consist of multiple hosts running several different operating systems — for instance UNIX and NT — on the host machines used to provide the CI, DB or APP (frontend machines in the presentation layer do not matter in this context).

In scenario 4 there is a standby host available to the CI. This standby host can run another R/3 application service or run in idle mode for normal operation. If the CI host fails, the CI is switched to its standby and connects itself again to the database system, which is running on a node external to the cluster (again, the application service on the standby is shutdown before the switchover is performed).

In scenario 5 there is an idle standby host available to the DB (this host can also run an APP if needed). If the DB host fails, the DB is switched to its standby. Both the CI and the APP on the external node are able to reconnect to the database system as soon as it becomes operational again.

As an example of employing scenario 5 in a mixed operating system environment, consider the following. Assume the DB can only run on UNIX, while the CI is put on an NT machine. The database system is installed on a node that is part of a UNIX-based workstation cluster (node C1, node C2). The other UNIX machine in the cluster is idle and configured to take over the DB, if node C1 fails. The CI is running independently on node E1 under operating system NT. Another application service is attached on node E2, which should be running NT as well.

Note that scenarios 4 and 5 can also be used jointly on a 3- or 4-node cluster, assuming a homogeneous operating system environment. In a 4-node cluster, both the CI and the DB are supported by dedicated standby host machines, which can take over a critical service if the initial host machine fails. In a 3-node cluster, there is a node for CI, DB, and standby (the standby might be running an APP). The standby (after shutting down the optional APP) can take over either the CI or DB service depending on which of the two hosts has failed.

Switchover Scenarios 1 and 2: A Typical Setup

The example below shows a typical setup of mutual CI and DB switchover on a two-node cluster (corresponding to scenarios 1 and 2 discussed above). If the DB host fails, the CI host takes over the DB service and runs a common CI+DB service thereafter. Alternatively, the DB host takes over the CI in the event of CI host failure. Both switchover options can be used at any time.

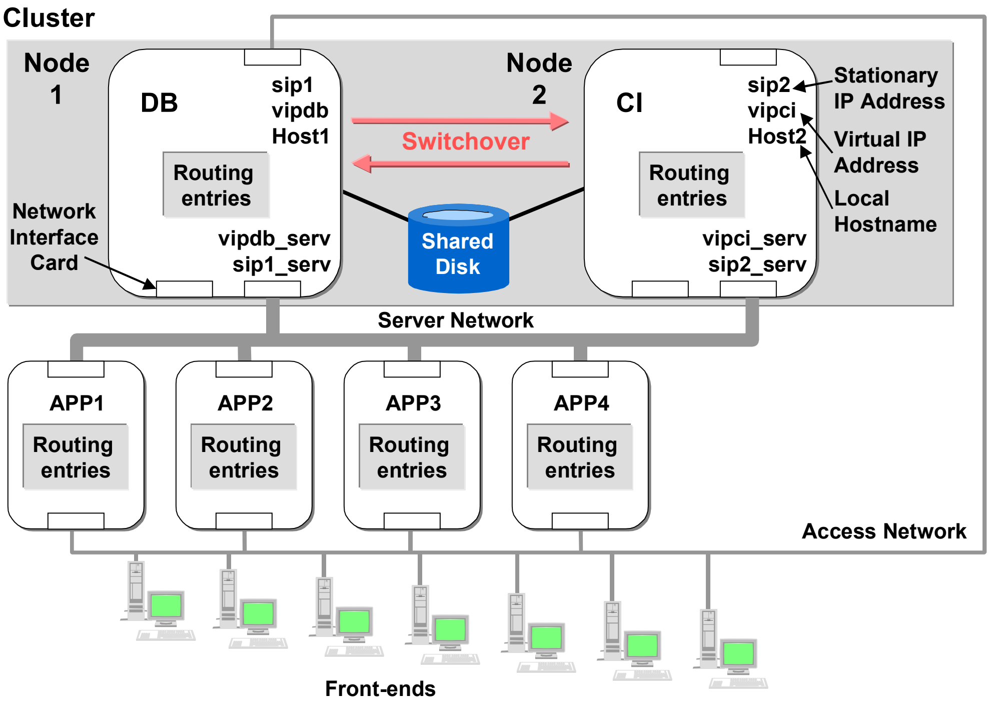
Two separate networks for frontend and server connections are used: the access and server networks. On both LANs, connections are handled via virtual IP addresses (`vipci` and `vipdb` on the frontend side, `vipci_serv` and `vipdb_serv` on the server side). Both addresses are switched over to the other cluster host together with their application component (CI or DB). Frontend machines and cluster-external application hosts keep accessing constant virtual IP addresses in any case. Constant routing entries in all hosts are required to make sure that the network traffic is properly split between frontend and server LAN.

For reasons of redundancy, two network interface cards (NICs) are installed on the server network. These avoid a SPOF in the hardware necessary to access the server network. The task of automatically substituting the standby NIC for the defective one can be handled by the switchover software, but this is done independently of any R/3 System switchover.

Such failures occur independently of any switchover scheme but can also be handled internally by switchover software.

For a detailed technical discussion of how to correctly configure network addressing in switchover clusters, see the SAP documentation *R/3 in Switchover Environments* (for information about how to find this, refer to “See also” at the end of this section).

Switchover Scenarios 1 and 2: A Typical Setup



See also:

R/3 in Switchover Environments (in SAPNet)

R/3 Configuration in Switchover Environments

This section gives example configurations of R/3 in conjunction with switchover products, the mapping of R/3 services onto several cluster nodes, and the effect failures have on the system. For more general information about R/3 configurations and mapping, see Mapping of R/3 System Services [Page 14].

R/3 Installation Issues

Currently, the R/3 installation tool (R3SETUP) offers the following options for the installation of R/3 components:

- Database only (DB)
- Central instance only (CI)
- Central instance and database (CI/DB)
- Dialog instance (APP, application services).

Although R3INST does not directly support R/3 installation in switchover clusters, it can be used to install all pre-packaged switchable R/3 units of the scenarios defined in Principal R/3 Switchover Scenarios [Page 235].

If you choose a central instance, this installs an R/3 instance with dialog, update, enqueue, batch, message, and spool service (a gateway process is implicitly started by the dispatcher on every R/3 instance).

The mapping of R/3 services can be changed very easily by changing some parameters in the instance profiles after installation (for example, to configure additional batch work processes on a dialog instance). The services configured on the R/3 instance (that is, dialog, update, enqueue, batch, message, gateway and spool services) are initially reflected in some filenames (profiles) and paths. However, the most important definition can be found in the instance profile (work process types and numbers) and the start profile (message service startup).

Example Configurations of R/3 in Switchover Environments

Important questions to consider when setting up a switchover cluster environment for R/3 include:

- Are there additional instances (that is, outside the cluster)?
- Should the central instance be available for normal online users?
- Should the update service be centralized (preferably on the CI) or distributed?
- Is another R/3 System running on a cluster node?

Using the approach outlined above, examples are given below of the most common R/3 configurations for switchover cluster environments. In SAP’s experience, these sample configurations cover most of the high availability R/3 installations.

R/3 Configuration in Switchover Environments

The scenarios described below are valid for R/3 Releases 4.0A onwards. For more information about R/3 configuration, see the SAP documentation *R/3 in Switchover Environments*. This also describes the behavior of the system prior to Release 4.0A. See the end of this section for details of how to access this documentation.

Central System

The central system is the simplest R/3 configuration. All R/3 services (except the SAPGUI) are installed on one host. If a failure occurs, switchover is made to an idle standby host (that is, to host 2 in the diagram below). The two nodes should have the same hardware configuration to guarantee the same system performance after a switchover**.**

Central System

R/3 Host Machines

		1	2	3	4	5	6
	Cluster Member?	Yes	Yes	No	No	No	No
R/3 Services	Dialog	7					
	Update	3					
	Enqueue	1					
	Batch	2					
	Message	1					
	Spool	1					
	DBMS	DB					

Configuration and Switchover Steps

In the event of a node failure, all R/3 services fail, so you simply need to restart the database service and the central instance on the standby host.

The effects of failure are described below:

- Failure of host 1
  – End-User: All users lose their sessions and must reconnect manually.
  – Transactions: All running transactions are aborted (rolled back).
  – Async Update: Async updates are either committed or stay in state “init”. If the parameter `rdisp/vb_start` is set, they are processed when the update service is available again.
  – Batch: All running batch jobs are aborted and must be restarted manually.
  – Print jobs: Print jobs being processed are aborted and must be released again.
- Failure of host 2

  The high availability protection for host 1 is lost. Host 2 should be repaired as soon as possible.

Central System with Additional Dialog Instances (Outside Cluster)

Additional instances can be installed as dialog instances using the R/3 installation tool (R3SETUP). You can change the number and types of work processes by editing the profiles.

Central System with Additional Dialog Instances (Outside Cluster)

		R/3 Host Machines					
		1	2	3	4	5	6
	Cluster Member?	Yes	Yes	No	No	No	No
R/3 Services	Dialog	7		7	7	10	10
	Update	3					
	Enqueue	1		3	3		
	Batch	2					
	Message	1					
	Spool	1				1	1
	DBMS	DB					

Configuration and Switchover Steps

The database service and the central instance are restarted on the standby host. The effects of failure are described below:

- Failure of host 1
  - End-User: All users on host 1 lose their sessions and must reconnect manually.
  - Transactions: All running transactions on all hosts are aborted (that is, rolled back), because the enqueue service loses all R/3 locks.
  - Async Update: Async updates are either committed or stay in state “init”. If the parameter `rdisp/vb_start` is set, they are processed when the update service is available again.
  - Batch: All running batch jobs on the failing node are aborted and must be restarted manually. Batch jobs on other R/3 instances fail when accessing a service on the node that failed (for example, database request).
  - Print jobs: Print jobs being processed on the failing node are aborted and must be released again. Print jobs processed on other nodes also fail, if print requests are stored in the database.
- Failure of host 2

  The high availability protection for host 1 is lost. Host 2 should be repaired as soon as possible.

Central System with R/3 Instances (Active Standby)

It makes sense to use the standby node for dialog processing during normal operation and you can achieve this by simply installing a dialog instance on the second host. The dialog instance on the standby node should use a different number from the central instance to avoid filesystem conflicts.

Central System with R/3 Instances (Active Standby)

R/3 Configuration in Switchover Environments

		R/3 Host Machines					
		1	2	3	4	5	6
Cluster Member?		Yes	Yes	No	No	No	No
R/3 Services	Dialog	7	7	7	7	10	10
	Update	3					
	Enqueue	1					
	Batch	2	3	3	3		
	Message	1					
	Spool	1				1	1
	DBMS	DB					

Configuration and Switchover Steps, if Host 1 Fails

The database service and the central instance are restarted on the standby host. The effects of failure are described below:

- Failure of host 1
  - End-User: All users on host 1 lose their sessions and must reconnect manually.
  - Transactions: All running transactions on all hosts are aborted (that is, rolled back), because the enqueue service loses all R/3 locks.
  - Async Update: Async updates are either committed or stay in state “init”. If the parameter `rdisp/vb_start` is set, they are processed when the update service comes up again.
  - Batch: All running batch jobs on the failing node are aborted and must be restarted manually. Batch jobs on other R/3 instances fail when accessing a service on the node that failed (for example, database request).
  - Print jobs: Print jobs being processed on the failing node are aborted and must be released again. Print jobs processed on other nodes also fail, if print requests are stored in the database.
- Failure of host 2

  Users logged on to the R/3 instance on host 2 are logged off and their current transactions are aborted. The users then have to log on to another R/3 instance. This can be simplified by using the group logon procedure (saplogon).

Central System with Distributed Update

If you have several dialog instances installed, you can install dedicated update services on each R/3 instance (the following example does not have update work processes configured on the central instance, because online users do not use it).

Central System with Distributed Update

R/3 Host Machines

		1	2	3	4	5	6
	Cluster Member?	Yes	Yes	No	No	No	No
R/3 Services	Dialog	2		6	6	7	7
	Update			2	2	3	3
	Enqueue	1					
	Batch	2		3	3		
	Message	1					
	Spool	1				1	1
	DBMS	DB					

Configuration and Switchover Steps

The database service and the central instance are restarted on the standby host. The effects of failure are described below:

- Failure of host 1
  - End-User: All users on host 1 lose their sessions and must reconnect manually.
  - Transactions: All running transactions on all hosts are aborted (that is, rolled back), because the enqueue service loses all R/3 locks.
  - Async Update: Async updates are either committed or aborted if accessing services located on the host that failed. If the parameter `rdisp/vb_start` is set, update records in state “init” are processed when the update service comes up again.
  - Batch: All running batch jobs on the failing node are aborted and must be restarted manually. Batch jobs on other R/3 instances fail when accessing a service on the node that failed (for example, database request).
  - Print jobs: Print jobs being processed on the failing node are aborted and must be released again. Print jobs processed on other nodes are processed until the R/3 instance is stopped.
- Failure of host 2

  The high availability protection for host 1 is lost. Host 2 has to be repaired as soon as possible.

Distributed System (Mutual Takeover)

For larger systems, the database load created by all R/3 instances might be just enough to be handled by the database host. In this case, installing a central instance on the database host would reduce performance. An option to avoid this is to install the central instance on the second cluster node. In the event of failure, each node takes over the application running on the other. After switchover, the performance might well be degraded, so you need to reconstruct the normal environment as soon as possible.

Distributed System (Mutual Takeover)

R/3 Configuration in Switchover Environments

R/3 Host Machines

		1	2	3	4	5	6
	Cluster Member?	Yes	Yes	No	No	No	No
R/3 Services	Dialog		7				
	Update		3				
	Enqueue		1				
	Batch		2				
	Message		1				
	Spool		1				
	DBMS	DB					

Configuration and Switchover Steps

If a virtual IP address is used for the database, the R/3 instance can reconnect (does not need to be restarted) to the database if host 1 fails. If host 2 fails, the R/3 instance is restarted on host 1.

The effects of failure are described below:

- Failure of host 1 with database reconnect
  - End-User: User gets an ABAP short dump or a sapdext message (that is, in the bottom line of the SAPGUI), but stays logged on.
  - Transactions: All running transactions are aborted (rolled back) when accessing the database during the switchover.
  - Async Update: Async updates currently processed are aborted when accessing the database during the switchover. The user is notified by an express mail, if parameter `rdisp/vbmail` is set.
  - Batch: All running batch jobs are aborted when accessing the database during the switchover and must be restarted manually.
  - Print jobs: Print jobs being processed are aborted when accessing the database during the switchover and must be released again.
- Failure of host 2
  - End-User: All users lose their sessions and must reconnect manually.
  - Transactions: All running transactions are aborted (rolled back).
  - Async Update: Async updates are either committed or stay in state “init”. If the parameter `rdisp/vb_start` is set, they are processed when the update service is again available.
  - Batch: All running batch jobs are aborted and must be restarted manually.
  - Print jobs: Print jobs being processed are aborted and must be released again.

Distributed System (Mutual Takeover) with Distributed Update

For a distributed system (database and central instance on different nodes) you can also distribute the update service. In the following example, the central instance is also used as dialog instance and therefore has its own update work processes.

Distributed System (Mutual Takeover) with Distributed Update

R/3 Host Machines

		1	2	3	4	5	6
	Cluster Member?	Yes	Yes	No	No	No	No
R/3 Services	Dialog		5	7	7	10	10
	Update		2	3	3	4	4
	Enqueue		1				
	Batch		2	2	2		
	Message		1				
	Spool		1			1	1
	DBMS	DB					

Configuration and Switchover Steps

If a virtual IP address is used for the database, the R/3 instances can reconnect (they do not need to be restarted) to the database if host 1 fails. If host 2 fails, the central instance is restarted on host 1.

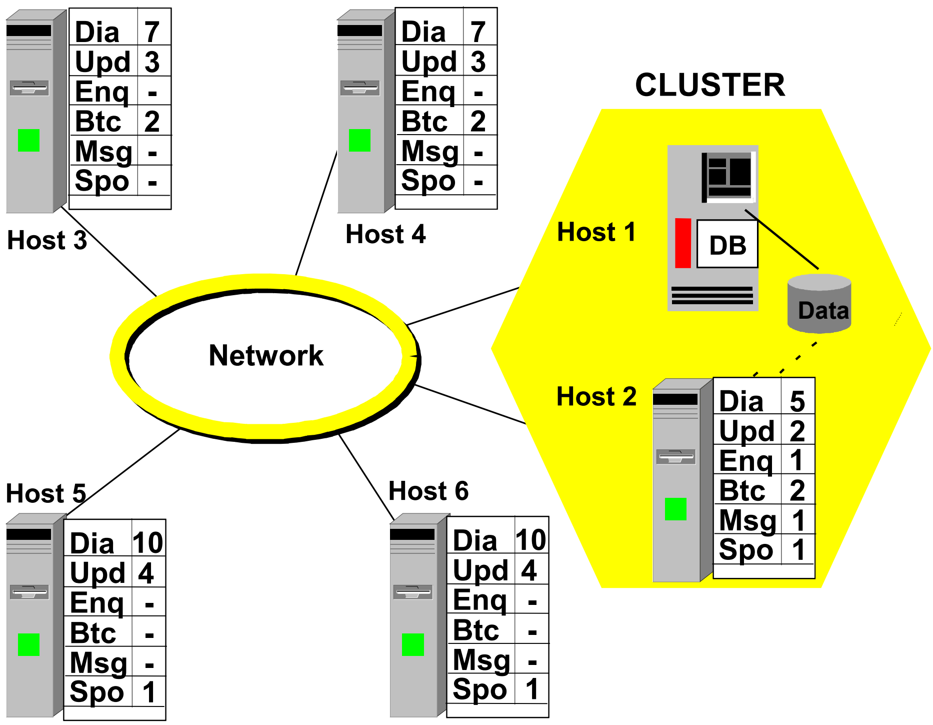
The effects of failure are described below:

- Failure of host 1 with database reconnect
  - End-User: User gets an ABAP short dump or a sapdext message (that is, on the SAPGUI bottom line), but stays logged on.
  - Transactions: All running transactions are aborted (rolled back) when accessing the database during the switchover.
  - Async Update: Async updates currently processed are aborted when accessing the database during the switchover. The user is notified by an express mail, if parameter `rdisp/vbmail` is set.
  - Batch: All running batch jobs are aborted when accessing the database during the switchover and must be restarted manually.
  - Print jobs: Print jobs being processed are aborted when accessing the database during the switchover and must be released again.
- Failure of host 2
  - End-User: All users logged on to the central instance on host 2 are logged off immediately. Other users are not logged off.
  - Transactions: All running transactions on all hosts are aborted (that is, rolled back), because the enqueue service loses all R/3 locks.
  - Async Update: Async updates processed on host 2 are either committed or stay in state “init”. If the parameter `rdisp/vb_start` is set, they are processed when the update service is again available. Other updates might fail when trying to access a service on host 2.
  - Batch: All batch jobs running on host 2 are aborted and must be restarted manually. Other batch jobs can be aborted when trying to access a service on host 2.
  - Print jobs: Print jobs being processed on host 2 are aborted and must be released again. All other print jobs are processed until the R/3 instances are stopped.

R/3 Configuration in Switchover Environments

The following picture represents this scenario as a graphic (compare with the tabular representation above):

Distributed System (Mutual Takeover) with Distributed Update



Distributed System (Mutual Takeover – Three Nodes)

If the performance degradation caused by the mutual takeover in a two-node cluster is too large, a three-node cluster could be used. In this scenario, there is a third node in the cluster that can take over either of the two services installed on the other nodes. If all three nodes have the same capacity, this helps avoid performance problems.

Distributed System (Mutual Takeover – Three Nodes)

		R/3 Host Machines					
		1	2	3	4	5	6
	Cluster Member?	Yes	Yes	Yes	No	No	No
R/3 Services	Dialog		5		7	10	10
	Update		2		3	4	4
	Enqueue		1				
	Batch		2		2		
	Message		1				
	Spool		1			1	1
	DBMS	DB					

Configuration and Switchover Steps

If a virtual IP address is used for the database, the R/3 instance can reconnect (does not need to be restarted) to the database if host 1 fails. If host 2 fails, the R/3 instance is restarted on host 3.

The effects of failure are described below:

- Failure of host 1 with database reconnect
  - End-User: User gets an ABAP short dump or a sapdext message (that is, on the SAPGUI bottom line), but stays logged on.
  - Transactions: All running transactions are aborted (rolled back) when accessing the database during the switchover.
  - Async Update: Async updates currently processed are aborted when accessing the database during the switchover. The user is notified by an express mail, if parameter `rdisp/vbmail` is set.
  - Batch: All running batch jobs are aborted when accessing the database during the switchover and must be restarted manually.
  - Print jobs: Print jobs being processed are aborted when accessing the database during the switchover and must be released again.
- Failure of host 2
  - End-User: All users logged on to the central instance on host 2 are logged off immediately. Other users are not logged off.
  - Transactions: All transactions running on all hosts are aborted (that is, rolled back), because the enqueue service loses all R/3 locks.
  - Async Update: Async updates processed on host 2 are either committed or stay in state “init”. If the parameter `rdisp/vb_start` is set, they are processed when the update service is available again. Other updates might fail when trying to access a service on host 2.
  - Batch: All batch jobs running on host 2 are aborted and must be restarted manually. Other batch jobs can be aborted when trying to access a service on host 2.
  - Print jobs: Print jobs being processed on host 2 are aborted and must be released again. All other print jobs are processed until the R/3 instances are stopped.

See also:

R/3 in Switchover Environments (in SAPNet)

High Availability Procedures at Your Site

Purpose

This section helps you adapt the information given in the rest of the documentation to the R/3 System setup at your site. The first section gives you some clues to how you can adapt the procedures at your site to cope with the special demands of high availability. Then there is a set of questions to help you assess your high availability requirements (this does not claim to be an exhaustive list of questions).

Finally, two checklists are provided. The general checklist covers a wide range of high availability issues while the single point of failure (SPOF) checklist covers how you can improve the overall availability of your system by looking at its particularly vulnerable aspects.

Process Flow

1. You think about high availability as early as possible in the installation of your R/3 System. It is cheaper and easier to build in high availability from the start than to add it in afterwards.
2. You regularly review your setup and procedures in the light of changing circumstances, particularly when the system configuration changes.

Procedures for Managing Your System

Purpose

To maintain a high level of availability for your R/3 System, you must have formalized procedures in place covering the full range of operational issues. The more complex the system, the more important this becomes. Without such procedures, the management of your data center becomes very difficult to coordinate, resulting in more chance of errors and therefore reduced system availability.

Process Flow

It is not the aim of this documentation to advise you exactly how to define management procedures at your site. However, SAP draws your attention to the following important areas, for which you should have defined adequate procedures:

Procedures for Managing Your System

System area involved	Cross-reference in this documentation
System configuration	GoingLive [Page 169] R/3 System Key Issues [Page 10]
Change management	–
System upgrade (R/3)	R/3 Upgrade [Page 34]
System failure (R/3)	R/3 System Failures [Page 27] Switchover Software [Page 216]
Archiving (R/3)	R/3 Level Data Archiving [Page 39]
Cluster administration	Cluster Technology [Page 130]
System monitoring and tuning	EarlyWatch [Page 169] Computing Center Management System (CCMS) [Page 166]
Database administration (backup, upgrades, monitoring/tuning)	Database Key Issues [Page 45] SAPDBA: Oracle [Page 155] SAPDBA: Informix [Page 159] Database Manager (DBMGUI): SAP DB [Page 163] DB2CC Tools for DB2 UDB [Page 166]

Database failure	Switchover Software [Page 216] Replicated Database Servers [Page 208] Replicated Databases [Page 183] Disaster Recovery [Page 182]
Database - R/3 connection failure	DB Reconnect [Page 174]
Disaster recovery	Disaster Recovery [Page 182]
Network	Network System Key Issues [Page 108]
Fault reporting with follow-up	–
Switchover	Switchover Software [Page 216]
System load assessment for mission-critical applications	–

Test your procedures

Pay special attention to procedures for managing failure scenarios, for example, with switchover software. If your system is growing rapidly, you should be prepared to re-test procedures that are sensitive to complexity and volume, for example, database recovery.

Important Questions About Your Setup

Purpose

This section lists important questions that you need to ask about your setup if you want to achieve high availability. Many of the questions are picked up in more detail in General Checklist [Page 250].

Process Flow

1. You ask yourself how much uptime your system needs:
   - What are the normal hours of operation (that is, the period when the system is needed):
     - During a day (for example, 12 or 24 hours)?
     - During a week (for example, 5, 6, or 7 days)?
   - Do the hours of operation include only the time online users access the system or is the time when batch jobs or other interfaces (for example, EDI) need access counted too?
   - What are the end users' expectations regarding uptime during normal hours of operation, for example, can they tolerate interruptions of a couple of minutes or hours?
2. You ask yourself how much downtime your system can tolerate:
   - What is the maximum downtime before you see a minor business effect?
   - What is the maximum downtime before you see a severe business impact (that is, loss of business)?
   - What is the maximum downtime before your business is at risk?

- Do you have special or unusual availability requirements (that is, critical periods of an application that should not be interrupted at all or where an interruption can be tolerated for less than a couple of minutes only)?

3. You ask yourself about your installation:
   - Is the R/3 System installed on existing hardware or is the entire hardware new?
   - Was the system planned with high availability in mind?
   - Have installation options been evaluated for application servers and database server?
   - Is special disk technology in place?
   - Are redundant network components in place?
   - What is the expected data volume and was database setup carefully planned?
   - What is the expected transaction load?
4. You ask yourself about your resources:
   - Internal
     - Is there a budget available to implement high availability features?
     - What is the availability of qualified personnel to operate the system?
     - Is the training of qualified personnel planned?
   - External
     - Are support contracts in place?
     - Is access to your system for remote support and maintenance possible?
5. You ask yourself about the environment or other factors:

   Are environmental or other failures likely, for example, unstable power supply in the area of your business?

General Checklist

Purpose

This checklist covers questions from Important Questions About Your Setup [Page 249], with cross-references to other sections in this documentation. A major problem is to define reasonable thresholds to "trigger" specific recommendations because there are so many complex and inter-related dependencies to consider. Therefore, many of the formulations are general and need to be adapted to your particular installation.

Risk analysis with checklists

Before you start to build high availability into the systems at your site, SAP strongly advises you to try and quantify which of the items listed in this general checklist and in the single point of failure (SPOF) checklist [Page 255] are relevant for you. Having ranked the vulnerable aspects of your system according to the costs of failure (for example, whether you need to call in an external engineer, order a replacement part, and so on), you are then in the best position to start improving availability.

Process Flow

1. You consider how much system uptime your business requires:
   - 5 days a week / 12 hours a day
     - Sufficient offline time to do system maintenance and offline database backups during operational days
     - Database size might require online database backups during operational days or partial offline backups
     - R/3 Upgrade [Page 34], system maintenance, offline database backups can be done during non-operational days (for example, at weekends)

     For information on database backups, see Backup with Oracle [Page 60], Archive and Backup with Informix [Page 69], Backup with SAP DB [Page 83], Archive and Backup with DB2 UDB [Page 91], Backup with DB2 for OS/390 [Page 97], or Backup with DB2/400 [Page 107].
   - 5 days a week / 24 hours a day
     - Online database backups are required during operational days
     - R/3 Upgrade [Page 34] and general system maintenance have to be done during non-operational days
     - Offline database backup has to be done during off-days
   - 7 days a week / 12 hours a day
     - Sufficient non-operational time available during each day to do system maintenance and offline database backups
     - Database size might require online database backups
     - Scheduling R/3 Upgrade [Page 34] might become an issue
   - 7 days a week / 24 hours a day
     - Special time slots have to be defined to do upgrades, both R/3 Upgrade [Page 34] and DBMS software upgrades. For more information about DBMS upgrades, see Upgrade with Oracle [Page 67], Upgrade with Informix [Page 81], Upgrade with SAP DB [Page 89], Upgrade with DB2 UDB [Page 95], or Upgrade with DB2 for OS/390 [Page 105].
     - Database backups have to be online. Refer to Backup with Oracle [Page 60], Archive and Backup with Informix [Page 69], Backup with SAP DB [Page 83], Archive and Backup with DB2 UDB [Page 91], Backup with DB2 for OS/390 [Page 97], or Backup with DB2/400 [Page 107] [Page 91].
     - Redundant hardware components are worth considering (see Disk Technology [Page 132], Switchover Software [Page 216], and Uninterruptible Power Supply (UPS) [Page 148].
     - Disaster Recovery [Page 182] is also worth considering
2. You consider how much system downtime your business can tolerate until there is only a **minor** business effect:
   - Several hours or one business day

    - Probably no special measures necessary for protection against hardware or operating system failure
    - Standard recovery procedures for databases are most likely sufficient

  – Less than above

    - Redundant hardware components might become necessary, such as Disk Technology [Page 132], Network System Key Issues [Page 108], Switchover Software [Page 216], and Uninterruptible Power Supply (UPS) [Page 148].
    - Database backup frequency, restore and recovery times have to be evaluated.

      If restore takes too long, backup devices need to be replaced with faster ones or more devices added. Alternatively, you might need to increase the backup frequency. See Recovery with Oracle [Page 66], Recovery with Informix [Page 80], Recovery with SAP DB [Page 87], Recovery with DB2 UDB [Page 95], Recovery with DB2 for OS/390 [Page 103] or Recovery with DB2/400 [Page 107].

      If data volume is simply too large to finish restore/recovery in an acceptable period, you need to evaluate your Disk Technology [Page 132] and employ, for example, mirrored disks to avoid restore and recovery altogether (except in the event of multiple simultaneous failure).

3. You consider how much system downtime your business can tolerate until there is a **major** business effect (for example, loss of business):

   If your business can only tolerate a few hours downtime:

   – You must make sure that a restore/recovery can be completed in the time available. Refer to Recovery with Oracle [Page 66], Recovery with Informix [Page 80], Recovery with SAP DB [Page 87], Recovery with DB2 UDB [Page 95], Recovery with DB2 for OS/390 [Page 103], or Backup with DB2/400 [Page 107] [Page 91].

     If this cannot be guaranteed, your disk technology has to be looked at (see next point).

   – Redundancy for hardware components becomes important, so evaluate the use of:

     - Special Disk Technology [Page 132] (for example, consider disk mirroring, RAID, or LVM). Disks are generally the most vulnerable of all hardware components so it makes sense to start with them.
     - Redundant network components. Refer to Network System Key Issues [Page 108].
     - Cluster CPUs with switchover solutions to protect the database server and/or the central application server. Refer to Switchover Software [Page 216].
     - Uninterruptible Power Supply (UPS) [Page 148] is cheap and worth considering.

   – If your database is Oracle or DB2 for OS/390, an alternative to switchover software is to use Replicated Database Servers [Page 208] together with the R/3 DB Reconnect [Page 174] feature.

   – More than one node should be available as application server. Also, at least two nodes should be prepared to act as central application server after a manual reconfiguration. This means that, if the node where the central application server is running becomes unavailable, another node should be prepared to start the central application server. Refer to Mapping of R/3 System Services [Page 14].

4. You consider how much system downtime your business can tolerate until it faces collapse:

   – Evaluate your system for single points of failure. Even if 90% of the system is equipped for high availability (for example, mirrored disks, redundant network components, and so on), this is worthless if one of the components in the unprotected remaining 10% fails.

   – If the time expected to replace critical hardware components or reconfigure critical software components is more than the period that would put your entire business at risk, you should seriously consider Disaster Recovery [Page 182] using a backup site.

5. You consider whether your business has periods with special availability requirements.

   If there are critical periods of an application that cannot be interrupted at all or where an interruption can be tolerated for less than a couple of minutes only, you might need to take extra precautions. Consider the following:

   – Disk Technology [Page 132] (for example, consider disk mirroring, RAID or LVM)

   – Network components [Page 108]

   – Switchover Software [Page 216] or the use of Replicated Database Servers [Page 208] together with the R/3 DB Reconnect [Page 174] feature (only available for certain databases)

   – Uninterruptible Power Supply (UPS) [Page 148]

6. You consider factors concerning your installation:

   – Age of system

     The approach you need to take depends on whether a new hardware system is being installed with the R/3 System or whether R/3 is to be installed with existing hardware:

     - If a new system is being set up, you should evaluate the high availability requirements at an early stage, and design the new system accordingly.

     - If R/3 is being installed on an existing system, you need to investigate the system for weak points.

   – Evaluate installations options

     You evaluate the installation options for application servers and the database server. Depending on the desired installation and high availability requirements, you need to carefully consider the Mapping of R/3 System Services [Page 14] since this might not be a straightforward task.

   – Expected Data Volume

     The database setup needs to be carefully planned:

     - Proper planning of database layout makes space management a lot easier. See Space Management with Oracle [Page 48], Space Management with Informix [Page 71], Space Management with SAP DB [Page 86], Space Management with DB2 UDB [Page 91] and Space Management with DB2 for OS/390 [Page 100].

     - Large databases (> 100 GB) might already cause problems when it comes to backup. The time spent for restore and recovery is probably too long. You should evaluate your disk technology [Page 132] since mirrored disks give additional options for backups (see RAID and LVM). Standard backup, restore, and recovery procedures might take too much time.

For more information about backup and recovery of databases, see Database Key Issues [Page 45].

– Expected Transaction Load

The transaction load influences the installation options for application servers and the database servers. Refer to Mapping of R/3 System Services [Page 14].

7. You consider your internal resources.

– Budget available to finance improvements

SAP would normally expect that improvements are undertaken in the following order, as finances permit:

- Disk Technology [Page 132]
- Server Network [Page 115]
- Switchover Software [Page 216] for DBMS and central application hosts
- Access Network [Page 120]
- Uninterruptible Power Supply (UPS) [Page 148]

– Availability of qualified personnel

The level of qualified personnel available to monitor the system during “normal operation” hours might influence the level of redundancy you choose:

- Qualified personnel not always available

  Certain technologies, such as disk technology [Page 132], switchover solutions [Page 216], and redundant network components [Page 108], reduce the need to have personnel available to handle errors.

- Qualified personnel always available

  You can probably rely on your personnel to handle errors.

8. You consider your external resources.

– Support contracts

The level of support contracts in place might influence your approach to high availability:

- “Special” maintenance contracts in place

  If you have such contracts with hardware and software vendors, for example, a guaranteed replacement of faulty hardware components within 24 hours, you might choose not to have a disaster recovery [Page 182] site or to implement less comprehensive redundant hardware, such as disk technology [Page 132], networks [Page 108], and so on.

- “Standard” maintenance contracts in place

  If your maintenance contracts are only “standard”, you might choose to have a higher level of availability to cover gaps in maintenance. Then you might choose to set up a disaster recovery [Page 182] site and to implement more comprehensive redundant hardware, such as disk technology [Page 132], networks [Page 108] and so on.

- Access to your system for remote support and maintenance

  Before implementation, consider using the GoingLive [Page 169] service. For proactive and highly qualified R/3 System administration, you can consider using the EarlyWatch [Page 169] service to avoid problems arising. SAP provides both services.

9. You consider environmental or other factors.

   Examples of these factors are:

   - Unstable power supply in your area

     Consider using Uninterruptible Power Supply (UPS) [Page 148].

   - Likelihood of disaster such as earthquake

     Consider using a disaster recovery [Page 182] site.

SPOF Checklist

Purpose

There are a number of single points of failure (SPOFs) in most systems and you should be aware of these before you start to build high availability into your system. However, what constitutes a SPOF depends on your particular system configuration. For example, a disk drive might be a SPOF in a given system configuration but, when mirrored, no longer be a SPOF. The major SPOFs are listed below, grouped into main system areas.

See also the table in “What is System Failure” in R/3 System Failures [Page 27], which lists the components of an R/3 System by system level.

Prerequisites

SAP suggests that, for each component of a planned or installed R/3 System listed in this process, you assess the following:

- Is the component a SPOF in your particular system configuration?
- Can you afford the risk of failure for a particular SPOF?

💡

Risk analysis using checklists

Before you start to build high availability into the systems at your site, SAP strongly advises you to try and quantify which of the items listed in this SPOF checklist and in the general checklist [Page 250] are relevant for you. Having ranked the vulnerable aspects of your system according to the costs of failure (for example, whether you need to call in an external engineer, order a replacement part, and so on), you are then in the best position to start improving availability.

Process Flow

1. You consider redundant configuration of the R/3 services dialog, update, batch, gateway, and spool – that is, on multiple host machines – to improve availability. This means that these services are **not** single points of failure.

You can improve the availability of the enqueue and message services – which cannot be configured redundantly – by the use of switchover software [Page 216].

For more information, see R/3 System Key Issues [Page 10].

2. You consider configuration of the database service to overcome its single points of failure:

   – Loss of connection between application service and database service. Use DB Reconnect [Page 174] to overcome this problem.

   – Loss of database data. For more information about this problem, see Replicated Databases [Page 183]. This is also discussed for each database manufacturer below.

3. You consider the database-specific recommendations in the following table:

Database	Single Points of Failure (SPOFs)
Oracle [Page 46]	• Database Instance – Database background processes (DBWR, LGWR, SMON, PMON...) – Memory structures (SGA, semaphores) You can protect the database instance using Switchover Software [Page 216] or Replicated Database Servers [Page 208] (only available for certain databases). • Database files – Control file – Current online redo log file – Data files You can protect the control file and the current online redo log file by using Oracle or proprietary disk mirroring. You can protect the data files by using disk mirroring. You should also protect all files by doing backups [Page 60]. Oracle Standby Databases [Page 187] can be used for a more comprehensive high availability solution that can withstand a disaster at one site.
Informix [Page 68]	• Database instance You can protect the database instance using Switchover Software [Page 216]. • Database data You can protect all relevant files by using Informix or proprietary disk mirroring. SAP strongly recommends some form of mirroring (preferably Informix) for, at the very least, the “critical” dbspaces (`logdbs`, `physdbs` and `rootdbs`). In any case, you should also perform regular archives and backups [Page 69]. See also Informix High-Availability Data Replication (HDR) [Page 193].

SAP Database System (SAP DB) [Page 82]	• Database instance You can protect the database instance using Switchover Software [Page 216]. • Database data You can protect all relevant devices by using proprietary disk mirroring (RAID 1 preferred) for all data devspaces and the system devspace. The Log mode should either be SINGLE with RAID1 or either NORMAL or DUAL without RAID mirroring. In any case, you should also perform regular backups [Page 83].
DB2 Universal Database (DB2 UDB) [Page 89]	• Database instance You can protect the database instance using Switchover Software [Page 216]. • Database data You should always perform regular backups [Page 91]. See also Replicated Standby Database for DB2 UDB [Page 197].
DB2 for OS/390 [Page 96]	• Database instance You can protect the database instance using Data Sharing for DB2 for OS/390 [Page 210]. • Database data You can protect the data by performing regular backups [Page 97] and using disk mirroring. You can also use a standby database to protect the data against disaster. See also Replicated Standby Database for DB2 for OS/390 [Page 201] and Data Sharing for DB2 for OS/390 [Page 210].
DB2/400 [Page 106]	• Database instance You can protect the database instance using Switchover Software [Page 216]. • Database data You should always perform regular backups [Page 107].
MS SQL Server [Page 108]	See the following high availability solutions: • Microsoft Cluster Server on Windows NT [Page 220] • Microsoft SQL Server Standby Database [Page 206] • Comprehensive Microsoft SQL Server High Availability Solution [Page 214]

4. You consider the network-specific recommendations [Page 108] in the following table:
   - Cabling
   - Active components (hubs, switches, routers)
   - Network Interface Card (NIC)

- SAProuter
- Network File System (NFS) – see “Single Points of Failure” in R/3 System Failures [Page 27]

5. You consider hardware and system software. For more information, see the table in section “What is System Failure” of R/3 System Failures [Page 27], which lists the components of an R/3 System by system level.
6. You consider disk technology [Page 132].

   Possible single points of failure in the hardware of a disk system include the following:

- Power supply
- Fan and cooling
- Internal/external cabling
- SCSI path from host machine to device
- Internal system bus
- Write-cache:
  - Non-volatile SIMMs or battery backup serve to address power failure
  - Mirrored SIMMs to address SIMM failure
- Read-cache: non-volatile SIMMs optional
- Battery power for the device to store cache to disk in case of power failure
- Controller
- Micro code
- Disk-internal storage processors
- RAID internal storage maps
- Disk spindles
- Spindle mechanism

Possible single points of failure in the disk-based data are the following:

- R/3 user data
- R/3 System data
- Software components:
  - The R/3 System
  - DBMS and log files
  - Operating system and swap space
- Root file system