

SAP/ MS SQL Server DBA in CCMS (BC-DB-MSS-DBA)



HELP.BCDBMSSDBA

Release 4.6C



Copyright

© Copyright 2001 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft[®], WINDOWS[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] and SQL Server[®] are registered trademarks of Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®], and OS/400[®] are registered trademarks of IBM Corporation.

ORACLE[®] is a registered trademark of ORACLE Corporation.

INFORMIX[®]-OnLine for SAP and Informix[®] Dynamic Server[™] are registered trademarks of Informix Software Incorporated.

UNIX[®], X/Open[®], OSF/1[®], and Motif[®] are registered trademarks of the Open Group.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA[®] is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT[®] is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax
	Tip

Contents

SAP/ MS SQL Server DBA in CCMS (BC-DB-MSS-DBA)	6
Database Backup	7
Backup Strategy	10
Backup Types.....	11
What Needs to be Backed Up?	13
Backup Device Management	16
Disk Configuration	19
Executing Backups	21
Backing Up with R/3	22
Backing Up with the Enterprise Manager	25
Monitoring Backups	27
Checking Individual Backups with R/3	28
Checking all Backups with R/3	29
Checking Backups with the Enterprise Manager	30
Database Restore	32
Distribution of Files to Disk	35
Restoring the R/3 Database after Disk Crash	37
Backing Up the Current Transaction Log	39
Restoring the Database and Log Backups.....	40
Restoring a Backup from Device	42
Restoring After Executables Disk Crash	44
Restoring the msdb Database from Device.....	46
Database Server Checks	47
Viewing the Error Log with R/3	49
Viewing the Error Log with the Enterprise Manager	50
Monitoring the R/3 Data Disk	51
Monitoring the R/3 Log Disk	52
Checking Consistency of the Database	53
Checking Consistency of Individual Tables	55
Checking for Missing Indexes	56
Database Server Management	57
Starting and Stopping the SQL Server	58
Setting SQL Server Startup Options	60
Changing Passwords	61
Registering a Server	63
Changing the Autogrow Settings	64
Configuring the SQL Server	65
Configuring Databases	66
Setting Single-User Mode	67
Executing Queries	69
Moving R/3 Database Files to New Disks	71
Displaying the SQL Server Version	73
Database Performance Tuning	74

Increasing the Size of the NT Page File.....	75
Optimizing the Virtual Memory Manager.....	76
Removing the Screen Saver	77
Database Management Tools	78
Enterprise Manager	79
R/3 Planning Calendar.....	81
Working with the R/3 Planning Calendar	83
R/3 Backup Monitor (MS SQL Server)	85
R/3 Static Database Monitor (MS SQL Server).....	87
SQL Server Alert Monitor.....	88
Alerts for SQL Server	89
Viewing and Customizing Threshold Values.....	92
SAP/SQL Server Database Monitor	93

SAP/ MS SQL Server DBA in CCMS (BC-DB-MSS-DBA)

Purpose

This component enables you to administer your Microsoft SQL Server database with the R/3 System.

Implementation Considerations

For more information about installing MS SQL Server with the R/3 System see the documentation:

R/3 Installation on Windows NT: Microsoft SQL Server Database

Integration

The Computer Center Management System of the R/3 System and the Enterprise Manager of the MS SQL Server provide functions that help you to manage your database. The functions that are most commonly used are described in this documentation.

If you are using a different database read one of the following guides to find out about available functions:

- [BC R/3 Database Guide: ADABAS for R/3 \[Ext.\]](#)
- [BC R/3 Database Guide: DB2 common server \[Ext.\]](#)
- [BC R/3 Database Guide:DB2/400 \[Ext.\]](#)
- [BC R/3 Database Guide: Informix \[Ext.\]](#)
- [BC R/3 Database Guide: Oracle \[Ext.\]](#)

Features

The functions offered by the Computer Center Management System and the Enterprise Manager support you in performing all the essential tasks involved in managing a database. They enable you to:

- Schedule and create backups
- Restore a damaged database
- Monitor a database
- Configure a database
- Perform routine tasks such as starting or stopping the database

Database Backup

Purpose

Your database is always subject to the risk of possible damage. There are many reasons why data may be lost or become corrupted. Disks may crash, power may be cut off, users may make mistakes, viruses may infiltrate the system or, in the worst case, a fire or earthquake could totally destroy the system. It is therefore vital to work out a security strategy that protects your system against data loss and enables you to restore it to a correct and consistent state. The most important part of any security strategy is to backup the database and its transaction log at regular intervals. This means the data and log from the database must be copied to another storage medium. When the database is damaged, these data copies can be reloaded in order to restore the database to a correct and consistent state.

Prerequisites

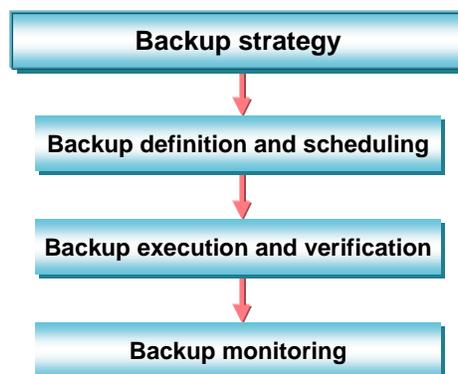
A backup can only effectively contribute to safeguarding your database, if it is performed as part of an overall backup and restore strategy. It is therefore important to first carefully work out a strategy and to test it before applying it in a productive system. It should define details such as the frequency of backups, what should be backed up at various times, which media are to be used and how backups are to be verified to ensure that they can be used for a later restore. When a strategy is planned, many different factors play a role. For example, it is important to take into account the transaction workload, the maximum permissible downtime, the available hardware and, in the worst case, the amount of data loss that is tolerable.

See also:

[Backup Strategy \[Page 10\]](#)

Process Flow

Once you have drawn up a backup strategy, you can begin to implement it. Both the MS SQL Enterprise Manager and the R/3 System provide the functions you need to define, schedule and execute backups and to perform other related tasks. Generally the procedure for backing up includes the following phases:



1. Definition and Scheduling of the Backup

Database Backup

Depending on the tool you decide to use, the procedure for defining and scheduling a backup will vary. However, the technical details that need to be specified in order to be able to perform a backup are always the same:

Backup Details

Backup contents	R/3 database, <i>msdb</i> database, <i>master</i> database and so on.
Backup type	Full database, transaction log, differential database and so on.
Backup destination	The tape devices or disks that the backup is written to.
Volume label	The names assigned to the tapes that the backup is written to. The naming conventions recommended by SAP should be used.
Expiration period	The length of time in which a backup tape is protected from being overwritten.
Execution time	Backups can be started immediately or scheduled to run at a specific time. Normally they are scheduled to run periodically, in accordance with a backup strategy that has already been defined.



Depending on the tool you use, you may have to enter these details in your system manually or they may be preset.

See also:

[Backing Up with the Enterprise Manager \[Page 25\]](#)

[Backing Up with R/3 \[Page 22\]](#)

2. Backup Execution and Verification

Normally backups are executed automatically. They are scheduled in advance in accordance with an overall backup strategy and then run automatically at a predefined time. However, it is the responsibility of the database administrator to always ensure that the correct tapes are inserted in the backup tape devices.

Usually a tape that has been written will be removed from the tape device and stored safely. In the past, however, it has been found that it is not certain that readable data actually exists on the tape that was supposedly written successfully. For this reason, the SQL Server offers the `verify` option. After the backup this always checks whether all the files have been written to the tape and whether they can be read. SAP recommends that this option is used for all backups.

3. Backup Monitoring

During and after the execution of a backup it should be part of the routine to check whether execution has been successful using the tools available.

See also:

[Monitoring Backups \[Page 27\]](#)

4. Consistency Check

The backup verification feature does not check the database for consistency. If a database contains corrupted data, this will be transferred to the backup. Therefore, ideally, consistency checks should be performed at regular intervals.

The database consistency check is scheduled in the R/3 System using the Planning Calendar.

See also:

[Checking Consistency of the Database \[Page 53\]](#)

Backup Strategy

Backup Strategy

The purpose of backups is to ensure that the database system can be restored to a correct and consistent state after it has been damaged. Simply performing any type of backup at arbitrary intervals is senseless. Backups are only meaningful, if they are integrated in an overall strategy that is designed to enable a safe and efficient recovery of a damaged database. An effective strategy normally describes a backup cycle of one month and includes answers to the following:

- Which parts of the system need to be backed up?
- What type of backup is suitable?
- How frequently are backups necessary?
- When should they be performed?
- Which media are they written to?
- How are backup devices managed?
- How are backups monitored and checked?

An effective strategy providing adequate insurance against data loss is always based on an appropriate hardware and system configuration.

See also:

[Backup Types \[Page 11\]](#)

[What Needs to be Backed Up? \[Page 13\]](#)

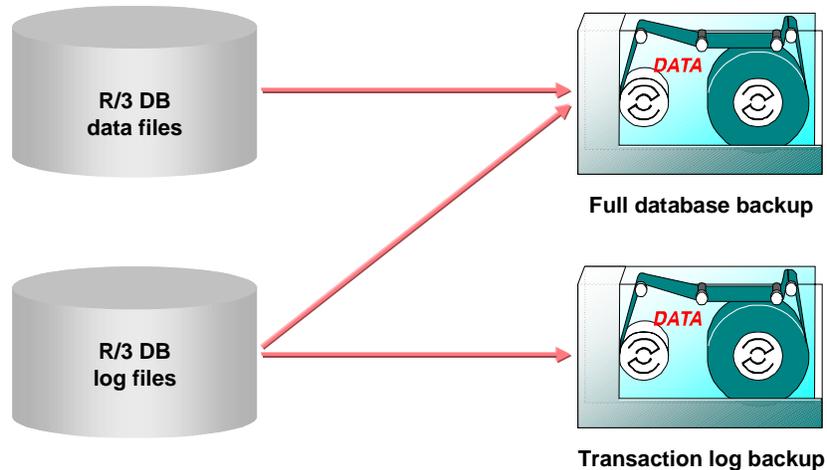
[Backup Device Management \[Page 16\]](#)

[Disk Configuration \[Page 19\]](#)

Backup Types

This subsection introduces the different types of backups that can be performed for the SQL Server system. Both the **Database** and **Windows NT** backups described here, should be part of your backup strategy.

Database Backups



- **Full database Backup**

A SQL Server database is always comprised of data and log files. The data files contain the actual data whereas the log files contain a record of all changes made to the database. In a full database backup, both the data and the log files are written to the backup device.

This type of backup has one drawback; the transaction log is not truncated after it has been backed up. This means that the inactive part of the log containing already completed transactions is not deleted after the backup. There is a danger that the transaction log might fill up making it impossible to continue working with the R/3 System. In contrast, a transaction log backup on its own deletes the inactive part of the log and thus enables space to be re-used thus preventing continual growth of the log files.

Full database backups are made when the database is online and do not require you to stop the R/3 System. However, they result in a significant amount of disk I/O and should therefore be performed at a time when the workload is minimal.

- **Transaction Log Backup**

A transaction log backup writes the log files of the database to the backup device. This enables the re-execution of transactions in the event of a database restore. When this type of backup is completed, the log is automatically truncated. This means log entries made prior to the oldest open transaction are deleted. Space in the log file can be re-used and in this way the continual growth of log files can be prevented.

Backup Types

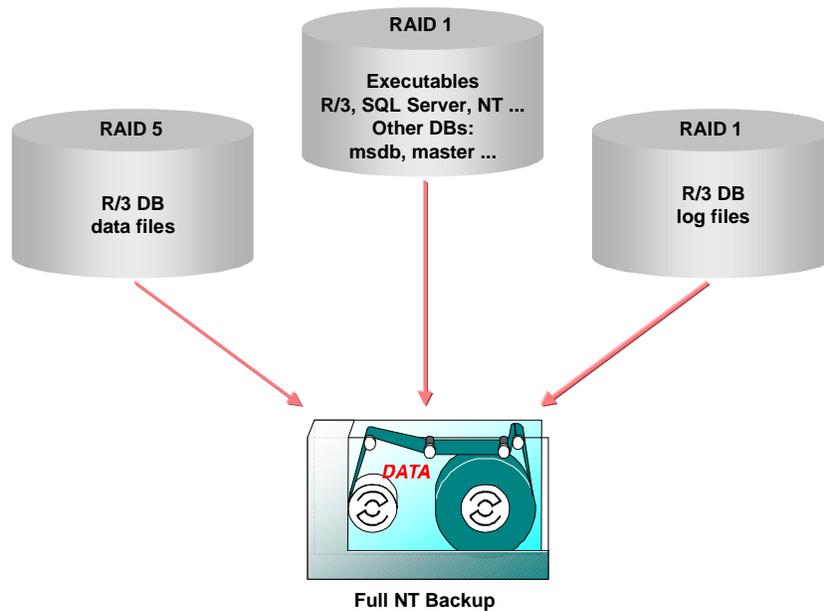
Transaction log backups are made when the database is online. As they only contain the changes since the last transaction log backup, they are much smaller than database backups and therefore have less impact on performance.

- **Differential Database Backup**

A differential database backup only backs up the pages of the database that were modified since the last full database backup. If such a backup is used to restore a database it is worthless on its own, and can only be used in combination with the preceding full database backup. When multiple differential backups are created, each one records changes made since the last full database backup and not only those made since the last differential backup.

Full Windows NT Backup

A full Windows NT backup backs up the entire system including the database, SQL Server and Windows NT files. An NT backup is only possible when the SQL Server is stopped because open files cannot be backed up.



What Needs to be Backed Up?

Resources in your system that need to be protected from loss are the Windows NT configuration and objects, the data in the databases and the corresponding transaction log files.



The following subsection gives general recommendations for a backup strategy. Keep in mind that the frequency and type of backups required vary widely. You need to find a solution that best fits your environment.

Windows NT, R/3 and SQL Server Files

The entire system, including all Windows NT, R/3 and SQL Server files must be backed up regularly using the NT Backup utility. The backup of these files is necessary for a restore operation when the disk on which SQL Server and R/3 executables are located crashes. In addition, it serves as an additional backup that may play a vital role in dealing with emergency situations where other routine backups have been damaged.

- Backup type: Full Windows NT backup.
- Frequency: The windows NT, R/3 and SQL Server files need to be backed up:
 - After installing an NT or SQL Server Service pack
 - Before special actions such as an R/3 Upgrade or Client Copy
- How to Perform: Use the Windows NT Backup tool. The database must be offline and the SQL Server shut down.



Use the [auxiliary NT system \[Page 19\]](#) to backup the files. For information on how to proceed, refer to the Windows NT documentation.



Never use Windows NT backup for regular database backups that are part of a predefined backup and restore strategy. NT database backups cannot be used to fully restore a database because the transaction logs cannot be applied. This means the changes made to the database after the last database backup cannot be redone. The database can therefore only be restored to the state it had at the time of the last NT backup and all the changes since the last database backup are lost.

Data of the R/3 and SQL Server databases

The data of the R/3, *master*, and *msdb* database must be backed up at regular intervals with database backups. These backups are essential to enable restore operations, if the database is damaged.

- Backup type: Full database backup
- Frequency: In general, the R/3 database <SAPSID> should be backed up once a day. It is sufficient to backup the *master* and *msdb* databases once a week or month, depending on the amount of database activity in your system. For administrative reasons, it is however easier to simply backup the *master* and *msdb* databases whenever you backup the R/3 data.

What Needs to be Backed Up?

This hardly has an impact on system throughput as these backups do not require much additional space or time.

- How to perform: Use the CCMS Planning Calendar of the R/3 System for backups that are regularly scheduled. Use the SQL Server Enterprise Manager for exceptional backups that are not part of the normal backup routine.

See also:

[Backing Up with R/3 \[Page 22\]](#)

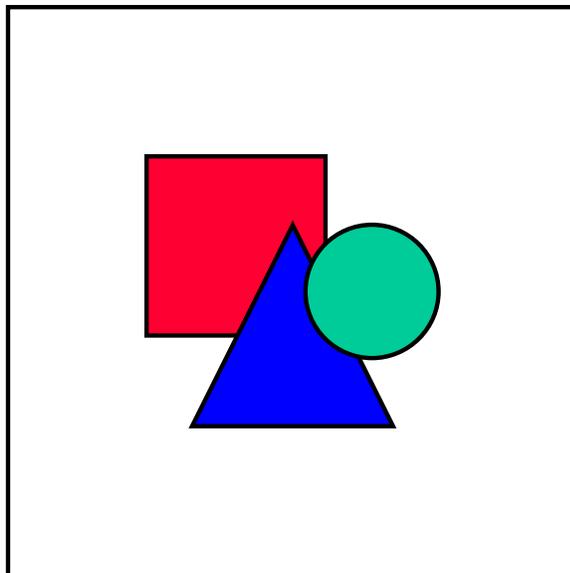
[Backing Up with the Enterprise Manager \[Page 25\]](#)

R/3 Transaction Log

The transaction log of the R/3 database that records changes made to the database must be backed up separately.

- Backup Type: Transaction log backup.
- Frequency: SAP recommends that a transaction log backup be scheduled every 30 to 60 minutes on a normal work day. If your R/3 System(s) are used productively at weekends, schedule additional backups for weekends.

To work out the frequency of transaction log backups, also take the fill rate of the log into account. This depends on the size of the transaction log and the transaction rate on your system. Whenever a transaction log is backed up, it is truncated. This means old entries are deleted so that enough space is available for new entries. Consequently, if the log is not dumped frequently enough, it will grow until the disk is full and no more changes to the R/3 database are possible.



When the disk system containing the transaction log crashes, it is impossible to backup the log that was in use at the time of crash. However, a backup of this log is necessary to restore the database to the state it had prior to the failure. It is only possible to return the database to the state it had at the time of the last transaction log backup. For this reason, SAP strongly recommends very frequent transaction log backups (every 30 to 60 minutes).

What Needs to be Backed Up?

- How to perform: Use the Planning Calendar in the R/3 System, or alternatively, the SQL Server Enterprise Manager.

See also:

[Backing Up with R/3 \[Page 22\]](#)

[Backing Up with the Enterprise Manager \[Page 25\]](#)

Backup Device Management

Backup Device Management

An essential part of a backup strategy is the management of storage devices. You need to decide:

- What type of backup media is to be used, for example, disks or tapes
- How long tapes should be saved before they are overwritten
- How many tapes are needed and their capacity
- How tapes should be labeled
- How backups that need more than one tape should be organized

Type of Backup Media

You can choose to backup either to tape or to disk. Usually tapes are used because they are less expensive and easier to handle.

If you decide to back up to tape, the tape device must be connected directly to the computer on which the SQL Server is running. DLT tapes are preferable to DAT tapes as they offer more security and have a greater capacity.



The term **tape backup device** refers to a tape device. When the R/3 System is installed, three tape backup devices are automatically created; R3DUMP0, R3DUMP1, R3DUMP2.

If you decide to backup to disk, SQL Server supports both local and network backup devices. A local device is a file on a local computer. A network backup device is a file on a shared network directory.

For instructions on how to backup to disk, refer to the *SQL Server Books Online*.

Expiration Period

- Transaction log and database backups
 - Set the expiration period to 28 days. This means the backup cannot be overwritten for 28 days. Both the SQL Server Enterprise Manager and the R/3 Planning Calendar provide options to protect tapes or disk backup devices from being overwritten during the backup cycle.
 - Keep the last database backup of each month for a year and the last database backup in the financial year permanently.
- Complete windows NT System Backup
 - Use at least two tape sets in rotation so that the last two backups are always available.

Number of Tapes Required

The number of tapes you need depends on:

- The number of days in the backup cycle

Backup Device Management

- The number of tapes required for the various database and transaction log backups of the day.

Work out the necessary capacity on the basis of the transaction log volume, the data volume and add approximately 30% to allow for database growth.



You are strongly advised to use 2 tapes per day; one for the database backup and one for the transaction log backup.

Tape Names

If you use the tape naming conventions that SAP recommends, you can identify the contents of a tape simply by looking at the label. Always make sure that the correct name sticker has been placed on the tape cartridge before you insert it into the tape device.

Tape labels used by SQL Servers have up to 128 characters. SAP recommends a naming convention with 5 or 6 characters. The characters have the following meaning:

- Character 1 identifies the database on the tape

R	R/3 database
M	msdb database
S	master database
C	combination of databases

Other databases in the SAP environment may be indicated with other characters.

- Character 2 and, if necessary also character 3, identify the type of backup

D	full database backup
DD	differential database backup
L	transaction log backup

Do not mix transaction log backups and database backups on one tape.

- The next two characters indicate the day of the month
- The last character identifies the tape number when a parallel backup is written to two or more tapes

A	device no. 1
B	device no. 2
C	device no. 3 and so on

When only using one device (no parallel backup), the last character is always set to A, even if the backup extends across more than one tape.



RD15A denotes the first tape (A) of an R/3 (R) database backup (D) on day 15 of the month.

Backup Device Management

The label RL05B shows that the tape contains an R/3 transaction log backup. The backup was performed on day 05 of the month. The B denotes that it is the second tape of a parallel backup.

Storing a Backup on Multiple Tapes

If the database backup no longer fits on one tape, there are two ways of distributing the backup data to several tapes:

- Sequential backup

Using this procedure, a database backup is written to two or more tapes sequentially using a single tape device.



SAP does not recommend this method. It is better to use tapes with a larger capacity so that a database backup fits on a single tape. This makes it easier to administer tapes in a backup cycle and simplifies the restore procedure.

Keep in mind that if large quantities of data are involved, backing up sequentially takes a very long time. If one tape is insufficient for the backup, rather schedule a parallel backup in the R/3 Planning Calendar.

- Parallel backup to several tapes

Several tape devices are required for a parallel backup. The data that is dumped is divided evenly among the tapes. Parallel backups require less time than sequential ones, but may create a greater system load. Note that a parallel backup can only be executed if you have an additional backup device.

See also:

[Backing Up with R/3 \[Page 22\]](#)

Disk Configuration

The following gives recommendations on how to distribute files to disks in an R/3 environment to ensure database security. You should have at least 3 disks.

- Locate **at least** each of the following components of your system on separate disk systems:
 - R/3 database on a RAID5 system (or RAID0+1)
 - R/3 transaction logs on a RAID1 system
 - Other files such as the NT operating system, NT page file, *tempdb* database and executables on a disk or RAID system

Use hardware RAID systems and not the software mirroring offered by the NT disk manager.
- Configure the system with **two NT operating systems**. The second, auxiliary version is used to perform backup and restore operations with the NT backup program.



Set the option NT option *Automatically adjust clock for daylight saving times* in the main NT system, but **not** in the auxiliary system.

- If your system has write back caches for hard disks and RAID controllers, switch these off. Otherwise, in the event of a power failure, the data in the cache could be permanently lost resulting in a corrupted database. In systems with a standby battery buffered cache, you may leave write caches on, if the manufacturer explicitly recommends this for databases.

Disk read caches should always be turned on for maximum performance.

The following table shows a minimal disk configuration for the R/3 System running with a SQL Server Database. Note that the letters for the devices have been chosen to indicate their contents, for example, D: for R/3 data.



Optionally, if you have the disk capacity available it would benefit the throughput of your system, if you place one or more of the following system components on separate disks. The distribution of files could then look like this:

P: The NT page file

T: Files for the *Tempdb*

C: Files for the SQL Server, NT and R/3 executables

Example of Minimal Configuration Table

Folder	Disk	Files
C:\WINNT	1	NT Operating System
C:\HELPNT	1	Auxiliary NT version
C:\MSSQL7	1	SQL Server executables
C:\MSSQL7DB	1	<i>msdb</i> and <i>master</i> databases

Disk Configuration

C:\TEMPDB	1	<i>tempdb</i>
C:\USR\SAP	1	R/3 executables
C:\pagefile.sys	1	NT page file
D:\<SID>DATA	2	R/3 database, data files
L:\<SID>LOG	3	R/3 database log files

For the sake of clarity, this documentation refers to the disks of the example configuration as follows:

Disk 1	Executables disk	RAID1 disk system
Disk 2	R/3 data disk	RAID5 disk system
Disk 3	R/3 log disk	RAID1 disk system

See also:

[Distribution of Files to Disk \[Page 35\]](#)

Executing Backups

Backups can be started in the Enterprise Manager or the R/3 Planning Calendar.

- For Backups that have to be scheduled periodically, in accordance with a predefined backup strategy, SAP recommends the use of the R/3 Planning Calendar.

For more information see [Backing Up with R/3 \[Page 22\]](#)

- For individual backups that are not planned and may be necessary in exceptional situations, use the Enterprise Manager.

For more information see [Backing Up with the Enterprise Manager \[Page 25\]](#)

Backing Up with R/3

Backing Up with R/3

Use

In the R/3 System, you can use the *CCMS DBA Planning Calendar* to schedule backups of the database or transaction log. SAP recommends that you use the Planning Calendar to schedule all regular backups that are part of your overall backup strategy. The calendar offers two approaches:

- You can schedule backups manually entering all technical details yourself.
- You can schedule backups using predefined planning patterns that automatically set all technical details for the execution of the backup.



A backup that has been scheduled in the *CCMS Planning Calendar* is displayed in the *Job List* of the Enterprise Manager. However, a backup that has been scheduled in the *Enterprise Manager* is **not** visible in the *Planning Calendar*.

Procedure

Scheduling Backups Manually

1. To access the Planning Calendar in R/3, choose *Administration* → *CCMS* → *DB Administration* → *DBA Planning Calendar*
Alternatively enter the transaction code DB13.
The Planning Calendar is displayed.
2. Double-click on the header of the day on which you want to schedule a backup.
If actions have already been scheduled on the chosen day, these are displayed.
If not, a list of the actions available for scheduling is displayed.
3. Choose either *Full Database Backup* or *Transaction Log Backup* from the list of actions.
Specify a *Start time* for the backup. In the *Period* field, specify the time interval in which the backup is to be repeated.
If actions have already been planned on the chosen day, you have to choose the *Insert* button before you can access the list of available actions.
4. Now a series of dialog boxes appears requiring you to enter a number of technical details.
Depending on whether you are backing up an entire database or only the transaction log, the entries required are different.

Entire Database

Dialog Box	Entry
<i>Database Selection</i>	Choose the databases you wish to back up.
<i>Backup Device Selection</i>	Select the backup device that the backup will be written to. Generally the default device R3DUMP0 is the correct entry here.

Backing Up with R/3

<i>Details for Tapes</i>	<p><i>Unload Tape</i> The tape is automatically unloaded</p> <p><i>Initialize Tape</i> The existing contents of the tape are overwritten.</p> <p><i>Verify backup</i> Ensures that all files that have just been written are readable.</p> <p><i>Format tape</i> Formats the tape so that the existing contents and header is deleted</p> <p><i>Expiration Period</i> Accept the default value of 28 days, if you are performing the backup in the framework of the overall backup strategy recommended by SAP. Otherwise enter a different value.</p>
--------------------------	--

Transaction Log

Dialog Box	Entry
<i>Backup Device Selection</i>	Select the device that the backup will be written to. Generally the default device R3DUMPO is the correct entry here.
<i>Log Backup Tape Options</i>	<p>Accept the expiration period of 28 days, if you are performing the backup in the framework of the overall backup strategy recommended by SAP.</p> <p><i>Verify backup</i> Ensures that all files that have just been written to the backup device are readable. Always choose this option.</p> <p>If you want to write multiple log backups of one day to a single tape, set the following options: 1st backup: Select <i>Initialize Tape</i> and deselect <i>Unload tape</i> Intermediate backups: Deselect <i>Initialize Tape</i> and <i>Unload Tape</i>. Last backup: Select <i>Unload Tape</i> and deselect <i>Initialize Tape</i>.</p>



If you assume that the backup of the database will not fit on one tape, you need to perform a parallel backup. Prerequisite is that you have installed a second tape device. Select both available devices in the Backup Device dialog box. The backup will automatically be written to both selected tapes in parallel.

5. When you have completed the above steps, the backup is scheduled to run on the chosen day of the week at the specified weekly intervals. To run the backup on every day of the week, repeat the scheduling process for all the remaining days of the week.
6. Before the scheduled backup starts, make sure that the correctly labeled tape volume is inserted in the device. The Planning Calendar automatically assigns the volume a label that complies with SAP naming conventions.

Backing Up with R/3

See also:

[Backup Device Management \[Page 16\]](#)

To find out which label has been assigned to a volume, double-click on the header of the day when the backup is scheduled. In the window that appears, select the backup in question and choose *Volumes Needed*.

The backup device and the name assigned to the tape volume are displayed.



If you wish to modify a backup scheduled in the Planning Calendar, do so within the Planning Calendar and not within the Enterprise Manager. Modifications made within the Enterprise Manager are not visible within the R/3 Calendar.

Scheduling Backups with Planning Patterns

1. To access the Planning Calendar in R/3, choose *Administration* → *CCMS* → *DB Administration* → *DBA Planning Calendar*.
Alternatively enter the transaction code DB13.

The Planning Calendar is displayed.

2. Choose *Planning* → *Planning Pattern*.

A dialog box with a planning pattern appears.

3. Depending on the backup strategy you have decided to implement, choose the pattern.

5 workdays, DB backup + hourly log backup (8am - 6pm) + Monthly Check DB

4. When you have selected the pattern, you are prompted to enter a time for the daily database backups. All other backups that are part of the pattern are scheduled on the basis of this time.



The planning pattern will only be applied, if there are no collisions with existing tasks scheduled at the same time.

All the backups that are scheduled are subsequently displayed in the calendar.

5. Check whether the parameter values that have been automatically set for the scheduled backups meet your requirements. To view the predefined parameters, double-click on the header of the relevant day in the calendar. A list of all the activities scheduled on the chosen day appears. Mark the action for which you wish to check the parameters and then select *Parameters*.

A sequence of dialog boxes appears showing the parameter values that have been set. Confirm each dialog box with *OK* in order to proceed to the next one.

For more information, see [Working with the R/3 Planning Calendar \[Page 83\]](#)

Backing Up with the Enterprise Manager

Use

The SQL Enterprise Manager allows you to backup the entire database or the transaction log on its own. SAP recommends the use of the Enterprise Manager for individual, exceptional backups that are not part of the normal backup routine.



For backups that are scheduled periodically, in accordance with a predefined backup strategy, it is advisable to use the *R/3 CCMS Planning Calendar*. This offers a range of useful functions for scheduling backups and easy access to other R/3 transactions that let you monitor backups.

Keep in mind that a backup scheduled and executed in the Enterprise Manager is not recorded in the R/3 Planning Calendar.

Procedure

Executing a Backup

1. In the Enterprise Manager, expand the server.
2. Expand the databases to display a list of existing databases. Select the database that you want to create a backup for and choose *Action* → *All tasks* → *Backup database*.

The *SQL Server Backup* dialog box appears.

3. Choose the *General* tab. Select the database you wish to backup.
4. Enter a name and description for the backup.
5. Select the type of backup you want to create. In the R/3 environment this can either be *Database - Complete*, *Transaction Log* or *Differential* backup.

Database Complete backs up the entire database including the data and log files. This type of backup does not truncate the transaction log.

Transaction Log backs up the transaction log only. The procedure includes the truncation of the log.

Differential backs up only the changes made to the database since the last full database backup.

6. Under *Destination*, select whether the backup is to be written to tape or disk. Make sure that the devices you intend to use are listed. If not, choose either *Add* to create a new device or *Remove* to delete an existing one. Usually R3DUMP0 is the correct device.
7. The backup can either be executed immediately or scheduled to run at a later time. To schedule the backup, choose the button beside the schedule entry field.
8. The backup can either be executed immediately or scheduled to run at a later time. To schedule the backup, choose the button beside the schedule entry field.

The *Edit Schedule* dialog box appears.

9. Complete the *Edit Schedule* dialog box.

Backing Up with the Enterprise Manager

One Time schedules the backup to run once. Set a date in the *On Date* box and a time in the *At Time* box.

Recurring schedules the backup to run at regular intervals.

To specify the details for the recurring backup, choose the *Change* button. You can then enter information in the *Task Schedule* dialog box that opens.

10. To complete the definition of the backup, switch to the *Options* tab. This offers various options that influence the execution of the backup. Choose the options as required. If you are uncertain which ones are suitable, access the information provided by the *Help* button.



Always choose *Verify Backup on Completion* as this makes sure that all the files are written to tape and can be read.

11. When you have completed the definition of the backup, choose *OK* to execute the backup immediately. If you have specified a particular time for the backup or scheduled it repeatedly, choose *OK* to save your entries. The backup will automatically be started at the time you have specified.

Monitoring the Backup

A backup that has been **scheduled** is executed as a job by the *SQL Server Agent*. To monitor a backup that is currently running, you have to access the *SQL Server Agent* that is responsible for executing scheduled jobs.

1. In the Enterprise Manager, expand the server and then *Management* and *SQL Server Agent*.
2. Select *Jobs*.

The right-hand pane displays an overview of all scheduled jobs.

3. The *Status* and *Last Run Status* column indicate whether the backup is running or has aborted with an error.



Backup tasks scheduled in the Enterprise Manager are not registered in the R/3 *Planning Calendar*.

Monitoring Backups

Purpose

It is not sufficient to simply schedule backups. You must also check regularly whether they were completed successfully. In the past there have been many situations where neglecting to do so resulted in a data loss for days or weeks.

Process Flow

Always check:

- Whether the most recent backup has run successfully

See also:

[Checking Individual Backups with R/3 \[Page 28\]](#)

[Checking Backups with the Enterprise Manager \[Page 30\]](#)

- Whether all the backups in the backup cycle are being executed according to the schedule. Gaps in a backup sequence can have serious consequences in a subsequent attempt to restore the database.

See also:

[Checking all Backups with R/3 \[Page 29\]](#)

Checking Individual Backups with R/3

Checking Individual Backups with R/3

Use

You can check whether your most recent database and transaction log backups are usable in the R/3 *Planning Calendar*. It is best to check the backup tape immediately after the backup has finished, before you remove the tape.

You should at least check whether the transaction log backups are completed successfully on a daily basis. Omitting such a check could have serious consequences. In the event of a hardware crash and a subsequent restore operation, this may result in the loss of several hours of data.

See also:

[Checking all Backups with R/3 \[Page 29\]](#)

Procedure

1. To access the Planning Calendar in R/3, choose *Administration* → *CCMS* → *DB Administration* → *DBA Planning Calendar*.

Alternatively enter the transaction code DB13.

The Planning Calendar is displayed on the screen.

2. Double-click on the scheduled backup task whose history you want to display. A backup that has failed is indicated in red.

A window with information about the task opens.

The upper half shows general information about the task. The lower half shows whether the task was successful and any messages that were generated.

If the backup failed you can sometimes find additional information in the SQL Server Error Log, the SQL Agent Error Log or the NT Application Event Log. In the NT log search for SQL Server and SQL Server Agent entries.

See also:

[Viewing the Error Log with R/3 \[Page 49\]](#)

[R/3 Planning Calendar \[Page 81\]](#)

Checking all Backups with R/3

Use

Checking individual backups is not sufficient to ensure that your backup strategy is being implemented successfully. You should also periodically check all of your database dumps together to make sure that your strategy is being adhered to and there are no gaps in your backup sequence.

In the R/3 System, the *Backup Monitor* offers an overview of all database dumps or of all transaction log dumps scheduled in the Planning Calendar.

Procedure

1. To access the Backup Monitor choose *Administration* → *CCMS* → *DB Administration* → → *Backup logs*.

Alternatively, enter the transaction code DB12.

The initial screen of the *Backup Monitor* appears.

2. Choose *Backup History* and then *All Backups*.

A result list appears showing technical details about all backups that have been executed. Select a backup and choose *History info* for additional details.

3. If a backup failed diagnose exactly what happened. To do so, it is necessary to also look at the SQL Server error logs.

See also:

[Viewing the Error Log with R/3 \[Page 49\]](#)

[Checking Individual Backups with R/3 \[Page 28\]](#)

[R/3 Backup Monitor \[Page 85\]](#)

Checking Backups with the Enterprise Manager

Checking Backups with the Enterprise Manager

Prerequisites

If you want to view information about a backup that you have scheduled, the SQL Server Agent must be started. The agent is responsible for scheduling and executing jobs and provides information about all scheduled jobs and their current status.



If a backup is not scheduled to run at a specific time in the Enterprise Manager, it is not registered as a job that is executed by the SQL Server Agent. For this reason it is not visible in the job overview or job history. In this case, you will only be able to find information in the SQL Server Error Log.

See also:

[Viewing the Error Log with the Enterprise Manager \[Page 50\]](#)

Procedure

1. In the Enterprise Manager expand the Server and then *Management* and *SQL Server Agent*.
2. Select *Jobs*.

The right-hand pane lists all the jobs that have been scheduled and displays detailed information such as the start date, current status and status of the last run. Jobs that have failed are marked with a red icon at the beginning of the line. To view further information, double-click on the job to access a number of tabs that show you how the job was originally defined.

3. To view the history of a particular backup job, select the job and choose *Action* → *View Job History* button.

You are now shown the times when the job ran and whether or not it was successful.

4. To display messages that were generated during the backup, select the option *Show step details*. Then select one of the steps that is displayed.

The messages generated by the SQL Server Agent for the selected step are displayed.

5. When you have looked at all available messages, also look at the SQL Server Error Log. This may contain additional information you need.

See also:

[Viewing the Error Log with the Enterprise Manager \[Page 50\]](#)



You should never clear the job history manually as this permanently deletes the information. By default, a maximum of 100 execution histories is kept for one SQL Server job and a maximum of 1000 histories for all jobs together. If you want to increase these values, you can do so by modifying the *SQL Server Agent Properties* dialog box. To access the box, select the *SQL Server Agent* and then choose *Action* → *Properties*. On the *Job System* tab make sure that the setting for *Maximum job history log size* is always set to at least 10 times that of *Maximum job history rows*.

Checking Backups with the Enterprise Manager

per job. As the history entries are stored in the *msdb* database, the size of *msdb* must be adjusted accordingly.

Database Restore

Database Restore

Purpose

When a database is damaged it can be returned to a correct and consistent state with a restore procedure. This generally involves reloading the most recent database backup and applying succeeding transaction log backups. The best method to restore a database depends on the type of damage that has occurred and on the backups that are available. In an optimal restore operation the database is recovered with minimal downtime and no data loss.

Prerequisites

As a database restore generally involves reloading backups, it is impossible to repair a damaged database without backups. A backup strategy that ensures that logs and data are regularly copied to another storage medium is therefore an essential prerequisite for any database restore operation. A well-planned strategy reduces the risk of data loss and minimizes the downtime required to reconstruct the database.

For more information, see [Backup Strategy \[Page 10\]](#)

Process Flow

There are different reasons why errors may occur that make it necessary to restore the database:

- **Disk failure**
A disk may crash resulting in the loss of the transaction log or database files.
- **User fault**
A user may, for example, unintentionally delete a file or import a wrong transport.
- **Database corruption**
In rare cases, a logical error may be discovered in the database.
- **Move to different hardware**
A disaster might have destroyed existing hardware or a move to faster hardware might be necessary.



After a power failure the database does not have to be restored. When the power is cut off, the system shuts down abruptly and the execution of transactions is interrupted. The SQL Server has an automatic recovery mechanism to deal with this situation. When the database is restarted, an automatic recovery mechanism starts working. This means open transactions are rolled back and completed transactions that were not written to the data files at the time of shutdown are re-executed.

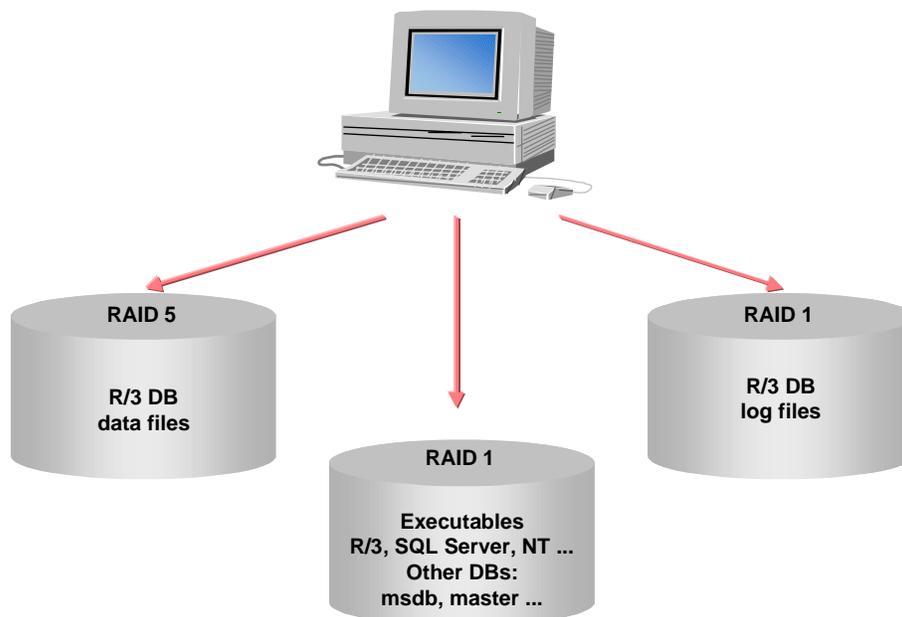
The most effective method to restore a database depends on the type of error that has occurred. Finding out exactly what happened is therefore the first step in any restore process. The following describes different error situations and gives an overview of the steps required to restore the database in these situations.

Restore After Disk Failure

The procedure to follow in order to repair a damaged database after disk failure differs, depending on how files have been allocated to disks and which of these disks have been

damaged. This documentation assumes that you have distributed your system to 3 different disk systems. This complies with SAP's recommendations that **at least** the following components of your system should be located on different disks:

- Log files of the R/3 database
- Data files of the R/3 database
- R/3, SQL Server and NT executables, and other databases, for example, *msdb*, *master* and *tempdb*



See also:

[Distribution of Files to Disk \[Page 35\]](#)

R/3 Data Disk Crash

The R/3 database normally resides on a RAID5 storage system for increased data security. If a single disk fails it simply needs to be replaced and will not disrupt database operation. However, if the whole RAID system crashes, the consequences are serious. The R/3 database will be damaged, work in the system will come to a standstill and a database restore will have to be performed before resuming normal work.

To restore the database after this type of failure, you first have to immediately backup the most recent transaction logs. Once this has been done, you can restore the database by loading the latest database backup and applying the succeeding transaction log backups.

For more information, see [Restoring the R/3 Database after Disk Crash \[Page 37\]](#)

R/3 Log Disk Crash

The R/3 log records all the changes that are made to the database. They enable transactions to be redone when a database is restored and therefore play a central role

Database Restore

in any restore operation. For this reason, SAP recommends that they be written to a RAID1 storage system that ensures data is protected against loss by a mirroring strategy. The transaction log will only be lost, if the original and mirrored log disk fail.

If the entire log disk system crashes, you need to proceed in the same way as for a failure of the R/3 data disk. The only difference is that you will not, in a first step, be able to backup the transaction logs that were in use at the time of the failure. You have to begin by restoring the most recent database backup and then continue with the application of the available transaction log backups.



You can only restore your database to the state it had at the time of the last log backup. All the following changes are lost because the current transaction log is damaged and it is therefore impossible to re-execute the changes it contains.

For more information, see [Restoring the R/3 Database After Disk Crash \[Page 37\]](#).

Executables Disk Crash

If your disk system with all other files except the transaction log and R/3 data files fails, this has far reaching consequences for the system. The NT operating system, R/3 executables, SQL server executables, *msdb* and *master* databases will be gone.

The best way to rectify the situation is to restore the crashed disk on the basis of the last available full Windows NT backup.

For more information, see [Restoring After Executables Disk Crash \[Page 44\]](#)

Restore after User Fault

A user might erroneously delete a table or import a wrong transport. Depending on what has happened, there are different ways in which you can proceed. One of the options that may be useful is a point in time recovery. It can restore the database to the state it had on a specific day at a specific time. Before you implement this procedure, SAP recommends that you first perform a full Windows NT backup. This safeguards you from losing your data, if the tape for restoring the database is unreadable.

For more information see the *SQL Server Books Online*.

Restore after Database Corruption

When you recognize that the data in your database is corrupted you need to diagnose the problem more precisely. The optimal method of restoring the integrity of the database depends on the extent and cause of the damage. You probably need assistance from your R/3 Basis consultant, in order to analyze the error and find the most effective solution. It may be possible to repair the corrupted database. Other options are:

- Database and transaction log restore
- Point in Time Recovery

Move to Different Hardware

There are two reasons why you may need to restore the entire system on different hardware:

- Total loss of hardware.
- Database copy, for example from a test system to a productive system.

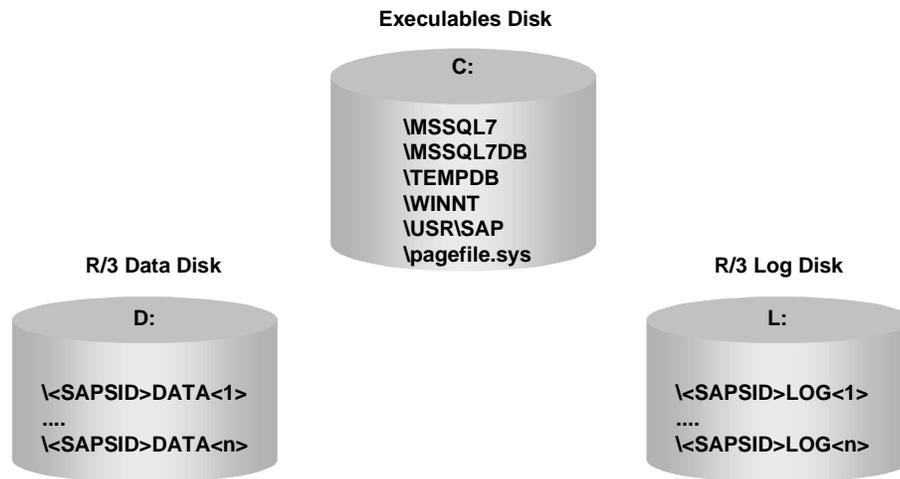
Distribution of Files to Disk

SAP recommends that the R/3 SQL Server System files are distributed to at least 3 different disks. The following illustrates the distribution of files to disks.

Folder	Disk	Files
C:\WINNT	1	NT Operating System
C:\HELPNT	1	Auxiliary NT version
C:\MSSQL7	1	SQL Server executables
C:\MSSQL7DB	1	<i>msdb</i> and <i>master</i> databases
C:\TEMPDB	1	<i>tempdb</i>
C:\USR\SAP	1	R/3 executables
C:\pagefile.sys	1	NT page file
D:\<SID>DATA	2	R/3 database, data files
L:\<SID>LOG	3	R/3 database log files

For the sake of clarity, this documentation refers to the disks as follows:

Disk 1	Executables disk	RAID1 disk system
Disk 2	R/3 data disk	RAID5 disk system
Disk 3	R/3 log disk	RAID1 disk system



Distribution of Files to Disk



If your disk configuration is different to the one illustrated here, you may need to contact your Basis consultant for advice on how to proceed after a disk crash.

Restoring the R/3 Database after Disk Crash

Purpose

The procedure described here should be followed to restore the database when:

- The R/3 database disk system is damaged
- or
- The transaction log disk system is damaged

Prerequisites

The process described here is only applicable to a configuration with three disk systems; one system for the R/3 database, one for the R/3 transaction logs and one for all other files.

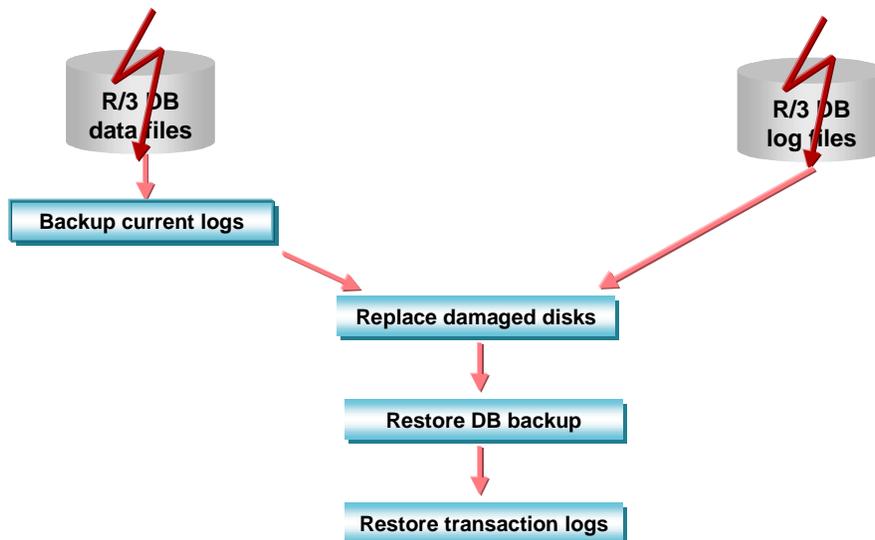
See also:

[Distribution of Files to Disk \[Page 35\]](#)

Process Flow

The database restore is comprised of a number of phases.

Restore Phases



1. Transaction Log Backup

If the disk system on which the R/3 database resides is damaged, it is vital to immediately backup the currently active transaction log to prevent a loss of data. Without a backup of the current log, the database can only be restored to the status it had at the time of the last transaction log backup. If work has been carried out on the R/3

Restoring the R/3 Database after Disk Crash

system since then, this work will be irrevocably lost. Therefore, after the failure of your transaction log disk, backup the current logs without delay.

For more information, see [Backing Up the Current Transaction Log \[Page 39\]](#)

2. Replacement of Damaged Disks

Replacing damaged disks in a RAID disk system is normally a straightforward procedure. If you are uncertain how to proceed, refer to the documentation of your hardware vendor to find out how to handle the disks. The new disks must be formatted and assigned the same drive letter as the old ones.

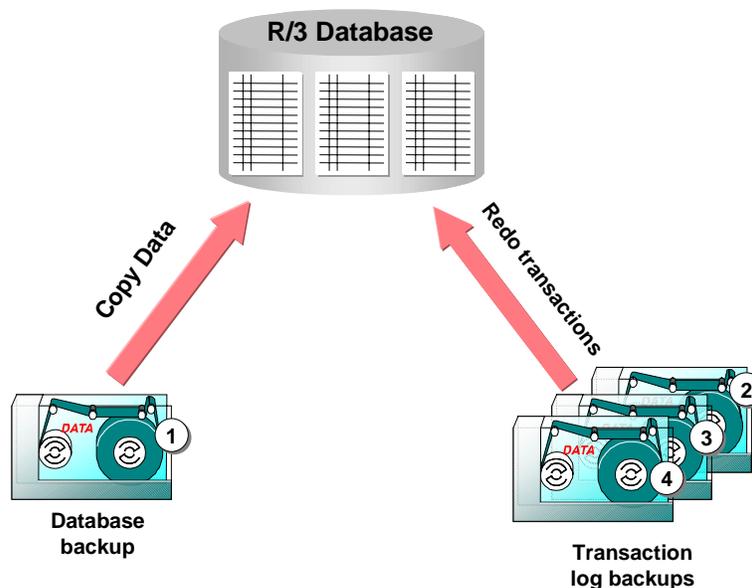
3. Database and Transaction Log Restore

The central phase of a restore operation is the reloading of the database backup and the application of the available transaction logs. When the database backup is reloaded, the database files are automatically recreated and the data is copied from the backup device to the newly created files. Once this has been done, the transaction logs are applied in the same sequence as they were originally made. This means that changes made to the database since the database backup are redone. In a final step, open transactions, that were not completed at the time of the database failure are rolled back.

At the end of the restore operation all transactions that were completed at the time of the database failure are written to the database and all incomplete transactions are been rolled back. Work in the R/3 System can be resumed.

For more information see [Restoring the Database and Log Backups \[Page 40\]](#)

Restoring the Database Backup and Applying Transaction Logs



Backing Up the Current Transaction Log

Use

If the disk on which the R/3 database resides has failed, it is important to immediately backup the current transaction log, otherwise the work carried out since the last log backup will be permanently lost.

Prerequisites

The current transaction log can only be dumped if:

- The disk on which the transaction log resides is undamaged
- The executables disk is undamaged



Do not stop the SQL Server before backing up the current transaction log. If you stop the server or reboot the system, you will be unable to backup the log!

Procedure

1. Insert a **new** tape into the tape device.
2. In the Enterprise Manager, start the *Query Analyzer*. To do this, expand your server and choose *Tools* → *SQL Server Query Analyzer*.

The *SQL Server Query Analyzer* window opens.

3. Enter the Transact SQL statement:

```
BACKUP LOG <SAPSID> TO <backup device> WITH NO_TRUNCATE, FORMAT
```

Where <SAPSID> is the name of your R/3 database and <backup device> the name of the device where you insert the backup tape, for example, R3DUMP0. Device names are case sensitive. For example, R3DUMP0 must be entered in uppercase.



The format option of the backup log command erases all the existing data on the tape. You should therefore be absolutely certain that you have placed the right tape into the tape drive.

4. Choose the execute icon to start the backup.

Restoring the Database and Log Backups

Restoring the Database and Log Backups

Prerequisites

The following explanation of the restore procedure assumes that you have set up your R/3 System on at least three disks as recommended by SAP and that the disk referred to as the [executables disk \[Page 35\]](#) is still accessible. This disk contains the SQL Server executables that enable the use of the Enterprise Manager and the *msdb* database that keeps a record of all backups that have been made. Both the Enterprise Manager and the *msdb* database must be accessible to enable the restore operation described here.

The overall restore process consists of several restore phases. If you do not have valid transaction log backups the restore process may consist of only one phase.

Phases

- Restore of the latest database backup
- Restore of the succeeding transaction log backups
- Restore of the latest transaction log backup. This may be an emergency transaction log backup which is still possible when the data disk crashes



It is absolutely vital to ensure that the correct restore option is set for the restore phases.

- For the last step of the overall restore process you have to set the restore option
Leave database operational. No additional transaction logs can be restored.
- For all other restore operations you have to set the option:
Leave database nonoperational, but able to restore additional transaction logs



In the following procedure you first restore the most recent database backup and then the succeeding transaction logs. During the entire procedure, do not execute any transactions and do not shut down the database server. A Server shutdown would write a checkpoint to the log and, as a result, it would not be possible to restore further transaction logs.

Procedure

1. Ensure that no users are connected to the database.
2. In the *Enterprise Manager*, expand the server and select the R/3 database you intend to restore. Choose *Action* → *All Tasks* → *Restore Database*.

The *Restore Database* dialog box opens.

3. Switch to the options tab and set the correct restore options.
Set the following:

Force Restore over Existing Database

Restoring the Database and Log Backups

Leave database operational. No additional transaction logs can be restored.

4. Switch back to the General tab, select the <SAPSID> database and specify Restore Database.
5. In the field *Show Backups of Database*, make sure that the <SAPSID> database is entered.

All the backups that are necessary to restore the <SAPSID> database are now displayed in a list. This list is generated from the backup history that is stored in the *msdb* database. Check that no backup is missing from the list, especially the last transaction log backup.



If backup history in the *msdb* database is not up to date then you have to restore the database and all the succeeding transaction log backups from device. In this case, follow the instructions given in [Restoring a Backup from Device \[Page 42\]](#).



If only the last transaction log backup is missing from the list of backups suggested for the restore operation, proceed as follows:

- Change the restore options in the *Options* tab to:
Leave database nonoperational, but able to restore additional transaction logs.
 - Continue with steps 6 and 7.
 - Restore the last transaction log from device. Follow the instructions given in the section [Restoring a Backup from Device \[Page 42\]](#). Note that for this last step it is essential to change the restore option to Leave database operational. No additional transaction logs can be restored.
6. Insert the correct tape with the most recent database backup into the tape drive. If you originally backed up the database to several tapes in parallel, you have to insert all the tapes used in parallel.



To find out the label of a tape you have to insert, select the required backup from the list and choose *Properties*. A window with information about the backup opens. *Media Name* shows the label of the tape.

7. To start the restore operation, switch back to the *General* tab. From the list of backups, select the most recent database backup and all the succeeding transaction log backups. Choose *OK* to start the restore operation.

The data from the database backup is copied to the disk and the succeeding log backups are applied to the database.

Restoring a Backup from Device

Restoring a Backup from Device

Use

Use this procedure if your *msdb* database is not up to date and you therefore cannot perform a normal restore using the backup history.

Depending on the overall restore operation that has to be carried out, you may have to repeat the procedure described here several times or only once. In an operation where you fully restore the entire database, repeat the procedure for the database backup and all the succeeding transaction log backups.



If you only need to restore the final log backup from disk, and have already restored all other backups using the [standard approach \[Page 40\]](#) based on the backup history, take special care when setting the restore option. It must be set to *Leave database operational*. No additional transaction logs can be restored.

Procedure

1. Insert the tape with the current backup into the tape drive.
2. In the *Enterprise Manager*, select the R/3 database and choose *Action* → *All Tasks* → *Restore Database*.

The *Restore Database* dialog box opens.

3. Switch to the *Options* tab. Set the options:

Force Restore over Existing Database

Leave database nonoperational, but able to restore additional transaction logs

Confirm your entry with *OK*.



If the actual restore is the last in the overall restore process, set the restore option to: *Leave database operational*. No additional transaction logs can be restored

4. Switch to the *General* tab. Specify that you want to restore *From Device*.
5. Choose *Select Devices*.
The *Choose Restore Devices* dialog box opens.
6. Choose *Tape* and then *Add*.
The *Choose Restore Destination* dialog box opens.
7. Select the device where the tape with the backup is located. Confirm your selection with *OK*.
The *Choose Restore Device* dialog box reappears.
8. Again select the device where the backup is located and choose *OK*.
The *General* tab reappears.
9. Look at the contents of the tape. To do this, select the device and choose *View Contents*.

Restoring a Backup from Device

The list of backups on the tape is displayed.

10. Select the backup and confirm with *OK*.
11. Choose *Database - complete* or *transaction log*
12. On the general tab, check that the <SAPSID> database is specified. Start the restore operation with *OK*.

The current backup is restored from device.

Restoring After Executables Disk Crash

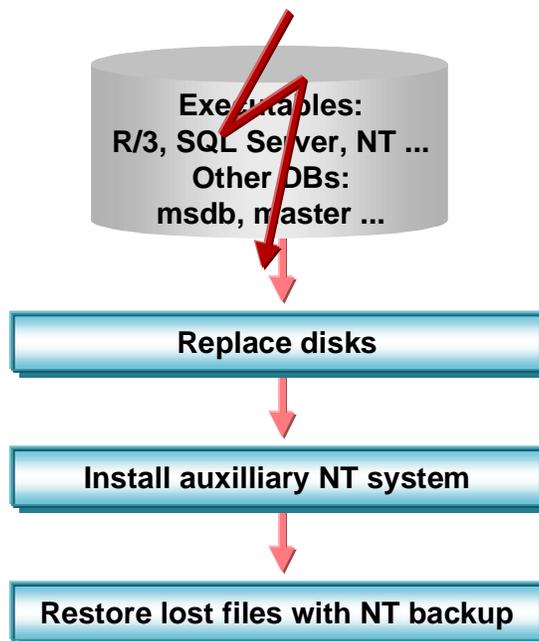
Restoring After Executables Disk Crash

Prerequisites

The procedure described here assumes that all files apart from the R/3 database and transaction logs are on a separate disk. If this disk crashes, you cannot use your R/3 System, your SQL Server or your operating system. Before you can begin to rectify the situation, you need a functioning operating system. In a first step you must therefore install and boot an auxiliary NT System. You can then start the restore procedure from this system.

Procedure

Overview of the Restore Steps



1. Replace the damaged disks.
2. Install an auxiliary NT system.
3. Restore the files that were lost on the crashed disk using the most recent full Windows NT backup. Refer to the Windows NT online documentation to find out how to restore an NT backup.



In this disk crash scenario, the R/3 database and transaction logs are unaffected. Do not reload these from the Windows NT Backup.

4. Reboot your server with the restored, primary NT system.

Restoring After Executables Disk Crash

The executables disk has now been restored to the state it had at the time of the last full Windows NT backup.



The *msdb* databases may not be up to date. It is therefore recommended that you now restore the most recent backup of this database.

For more information, see [Restoring the msdb Database from Device \[Page 46\]](#)

Restoring the msdb Database from Device

Restoring the msdb Database from Device

Use

If you have used a full Windows NT backup to restore the executables disk of your system, use the following procedure to ensure that your *msdb* database is up to date.

Procedure

1. To restore the *msdb* database, insert the tape with the latest *msdb* backup into the tape drive.
2. In the *Enterprise Manager*, select the *msdb* database and choose *Action* → *All Tasks* → *Restore Database*.
The *Restore Database* dialog box opens.
3. On the *General* tab, select the *msdb* database and choose *From Device*.
The *Choose Restore Devices* dialog box opens.
4. Choose *Tape* and then *Add*.
The *Choose Restore Destination* dialog box opens.
5. Select the device where the *msdb* backup is located. Confirm your selection with *OK*.
The *Choose Restore Devices* dialog box reappears.
6. Select the device with the *msdb* backup and choose *OK*.
The *General* tab reappears.
7. Look at the contents of the tape in the tape drive. To do so, select the device and choose *View Contents*.
The list of backups on the tape is displayed.
8. Select the *msdb* backup and confirm with *OK*.
9. On the general tab, check that the *msdb* database is specified as the database that is to be restored. Choose *OK* to start the restore operation.
The *msdb* database is now restored to the state it had at the time of the last backup.

Database Server Checks

Use

An important part of the database administrator's work is to perform regular checks of the system. Any problems must be detected and eliminated as early as possible in order to limit the subsequent damage inflicted on the system. Important tasks that should be part of the normal checking routine are database error log analysis, disk space monitoring and database consistency checks. Both the R/3 System and the Enterprise Manager provide functions to perform these tasks.

Features

Error Log Display

The SQL Server error log informs you whether operations such as backup or restore have been completed successfully and whether any problems have arisen in the system. To access the error log, use either:

- The [R/3 Database Monitor \[Page 49\]](#)
or
- The [Enterprise Manager \[Page 50\]](#)

Disk Space Monitoring

The Enterprise Manager and the R/3 database monitors offer various options to view the size of the R/3 database and transaction log:

- To get a quick overview of the current size of the R/3 data files, use the Enterprise Manager. It shows used space, free space and total space for each database file.
For more information, see [Monitoring the R/3 Data Disk \[Page 51\]](#)
- To find out how the size of the database has changed over a longer period of time, use the R/3 Database Monitor. It gives a tabular overview of the database size history, showing database size and free space for each day of the week.
For more information, see [Monitoring the R/3 Data Disk \[Page 51\]](#)
- To reliably work out the maximum size of the R/3 transaction log, use the R/3 Backup Monitor. It shows a backup history that includes the size of the backups made. By looking at the size of the largest backup you can determine how big an individual log file can become.
For more information, see [Monitoring the R/3 Log Disk \[Page 52\]](#)

Database Consistency Checks

The R/3 System provides a number of checks that enable you to determine whether the database is consistent:

- The database consistency check examines the entire database to determine whether page links and page reservations are correct.
For more information, see [Checking Consistency of the Database \[Page 53\]](#)
- The table check examines individual tables to find out whether they are physically consistent.

Database Server Checks

For more information, see [Checking Consistency of Individual Tables \[Page 55\]](#)

- The missing indexes check determines whether any indexes defined in the ABAP/4 Dictionary are missing in the database.

For more information, see [Checking for Missing Indexes \[Page 56\]](#)

Viewing the Error Log with R/3

Use

The error log is a central file in which all the important actions and error messages concerning the database are recorded. The database administrator should examine the log every day in order to detect any errors that might have occurred as early as possible.

For more information about error messages, see the *SQL Server Books Online*.

Procedure

1. To view the error log, enter the transaction code ST04.
2. Choose *Detail Analysis Menu*.
3. Choose *Error Logs*.
All the available error logs are listed.
4. To display the contents of a particular log, click the button to the left of the entry and then click in the scroll bar.



A new error log is created each time SQL Server is started. SQL Server retains backups of the previous six error logs. If the log file name is E:\MSSQL7\LOG\ERRORLOG, the last 6 log files are archived under the following names:

```
E:\MSSQL7\LOG\ERRORLOG.1  
E:\MSSQL7\LOG\ERRORLOG.2...  
E:\MSSQL7\LOG\ERRORLOG.6
```

Each time the SQL is started, the archived oldest log files are overwritten and are lost, unless you save them first. If you have a serious problem, you should immediately save these files otherwise they will be lost.

See also:

[Viewing the Error Log with the Enterprise Manager \[Page 50\]](#)

Viewing the Error Log with the Enterprise Manager

Viewing the Error Log with the Enterprise Manager

Procedure

1. In the the Enterprise Manager, expand the server, *Management* and *SQL Server Logs*.

All the currently available logs are displayed.

2. Choose the error log that is of interest to you. This might be the *Current* log or one of the archived logs. The date beside the log indicates when it was archived.

The messages in the log are displayed in the right-hand pane. You can display the last six logs in the *Error log* menu.



A new error log is created each time SQL Server is started. SQL Server retains backups of the previous six error logs. If the log file name is E:\MSSQL7\LOG\ERRORLOG, the last 6 log files are archived under the following names:

E:\MSSQL7\LOG\ERRORLOG.1
E:\MSSQL7\LOG\ERRORLOG.2...
E:\MSSQL7\LOG\ERRORLOG.6

Each time the SQL is started, the archived oldest log files are overwritten and are lost, unless you save them first. If you have a serious problem, you should immediately save these files otherwise they will be lost.

Monitoring the R/3 Data Disk

Use

It is vital to regularly check the space used by the server. If the data file or log file of the database is full, the R/3 System will come to a standstill. Although, after the installation of R/3, the autogrow option is set and files automatically grow whenever they are full, this still does not guarantee that disks will not run out of space. Checking the amount of free space available on the disk must always be one of the routine tasks performed by a database administrator.



When you look at the amount of free space specified for logs, keep in mind that the data displayed reflects snapshots taken at a specific point in time. As logs are backed up and truncated at hourly intervals, a large amount of free space might be displayed because the log has just been truncated. For more information see [Monitoring the R/3 Log Disk \[Page 52\]](#).

Procedure

Monitoring the Current Database Size

Use the following procedure in the Enterprise Manager, to get a quick overview of the current size of the database files and the free space available.

1. In the Enterprise Manager, expand the server and *Databases*.
2. Select the <SAPSID> database.

The right-hand pane displays general information about the <SAPSID> database.

3. From the bar at the top of the pane, choose *Space Allocated*.

The free space, used space and total space is displayed for each database file.

Monitoring the Database Size History

Use this procedure to get an overview of the space used by the R/3 database over a longer period of time.

1. To access the *R/3 Database Monitor*, enter transaction code ST04.
2. Choose the *Detail analysis* menu.
3. Choose *DB Space History*.

The *View database history* window opens.

4. Use the available pushbuttons to change the period for which you want to view space history.

Monitoring the R/3 Log Disk

Monitoring the R/3 Log Disk

Use

The size of the transaction log must be checked regularly to work out how much free space is available on the log disk. There should always be enough free space to allow the next file extension. When the R/3 System has been installed the autogrow increment is set. At least the size of this increment should be available on the log disk to permit the next file extension. If less space is available and the transaction log file fills up, the R/3 System will come to a standstill.

Ideally, the transaction log should never be filled to more than 60-70%. If the transaction log regularly exceeds this level between 2 transaction log backups, the transaction log must be saved at more frequent time intervals.

The size of the log can be assessed on the basis of information given for completed backups in the R/3 Backup Monitor.

Procedure

1. To access the Backup Monitor choose *Administration* → CCMS → DB Administration → Backup logs.
Alternatively, enter the transaction code DB12.

The initial screen of the *Backup Monitor* appears.

2. Choose *Backup history* and then *Logs Backup*.
3. A result list appears. Find the largest transaction log backup of the past week. Select a column and then *History info* to find out the number of pages that were processed during the backup. To work out the amount of space used in the transaction log, multiply the number of dumped pages by 8 KB. You can then work out how much free space is left on the transaction log disk.



If you use a RAID1 disk system exclusively for the R/3 transaction log and create hourly log backups, you will rarely encounter space problems. The R/3 log file is initially created with a size of 1.5 GB. The smallest RAID disk normally has 4 GB space and the log file can therefore grow to 4 GB.

Checking Consistency of the Database

Use

A database consistency check performs a thorough check of the entire database. It examines all tables in the database to find out whether index and data pages are correctly linked and indexes are in proper sorted order. It also checks that all pointers are consistent and that the data information on each page, and page offsets are reasonable. It enables the early recognition of problems and thus prevents problem escalation and possible loss of data.

Schedule the consistency checks with the R/3 Planning Calendar. Checks should run at the weekend when you have time. When the check runs, no other task should be scheduled.



At the SQL Server level, the R/3 database consistency check executes the DBCC CHECKDB command. This typically locks user tables, indexes and system tables when running. In addition, it is very I/O intensive and should therefore not be run during normal operation, but a times when the system load is low.

For more information, see R/3 Note 142731 *DBCC Checks for SQL Server 7.0*

Procedure

1. Start the R/3 Planning Calendar with *Administration → CCMS → DB Administration → DBA Planning Calendar*.
Alternatively enter the transaction code DB13.
2. Double-click on the header of the day on which you want to schedule the consistency check.
If actions have already been scheduled on the chosen day, these are displayed.
If not, a list of the actions you can schedule is displayed.
3. Choose *Check database consistency* from the list of actions that can be performed. If actions have already been scheduled on the day, you have to choose *Insert* to access this list.
4. Enter the time when the check is to start and, if required, the period in which it is to be repeated. Confirm your entries with OK.
The database consistency check is now scheduled. It is entered in the calendar and visible on the days when it will be executed.
5. When the check has completed, you can find out whether it ran successfully in the Planning Calendar. If it is highlighted in red, inconsistencies were discovered or errors were generated during the execution of the check. To find out more, double-click on the task.

The *SQL DB Administration in the SAP Environment* window opens. It displays information about the task and gives a short overview of the task history. The task log is displayed.



Error messages for the DBCC check are also written to the file `\MSSQL7\CCMS_CHECKDB_HIST.TXT`. If errors occurred, always look at this file because it contains all messages generated and is easier to read than the list of errors in the Planning Calendar.

Checking Consistency of the Database

6. Some error information is also recorded in the NT Error Log. To locate the information in the log, look for SQL Server and SQL Executive entries made at the time the task ran.
7. If any errors are discovered, contact your Basis Consultant for support. Errors must be corrected immediately. If they are left, they can lead to serious errors in your system.

To find out how to change scheduled actions see:

[Working with the R/3 Planning Calendar \[Page 83\]](#)

Checking Consistency of Individual Tables

Use

You may suspect that a table is no longer physically consistent, due to entries in the error log. In this case, you can check the table on its own, without examining the consistency of the entire database. Start a table check within the R/3 System using the R/3 Static Database Monitor.



When the consistency of a table is checked, a shared table lock is held on the table.

Procedure

1. To start the R/3 Static Database Monitor, enter the transaction code DB02
2. Choose the *Detail Analysis* button.

A popup appears.

3. Enter the name of the table you want to check.

Information about the table is displayed.

4. Choose *Check Table* to start the table analysis.

The results of the analysis are shown on the screen. All the error messages that were generated during the check are displayed.

See also:

[Viewing the Error Log with R/3 \[Page 49\]](#)

[R/3 Static Database Monitor \[Page 87\]](#)

Checking for Missing Indexes

Checking for Missing Indexes

Use

It is useful to compare the indexes that exist on the database with those that exist in the ABAP/4 Dictionary after an R/3 upgrade.

Due to the size of the tables in the SAP environment, tables need to be accessed via indexes to sustain an acceptable level of performance. If an index is missing or has been changed, it can have a negative effect on the performance of certain SAP reports.

Use the R/3 Static Database Monitor to find out whether any indexes that are defined in the ABAP/4 Dictionary do not exist in the database.

Procedure

1. To start the R/3 Static Database Monitor, enter the transaction code DB02
2. Choose the *Database analysis* button.
A list of available database checks is displayed.
3. To start the search for missing indexes, choose *DB <-> ABAP/4 Dict.* In the dialog box that appears choose *Recheck*.
4. When the check has finished, the results are displayed on the screen.

Result

In the results screen, the first section shows the indexes which are declared in the SAP ABAP/4 Dictionary, but which do not exist in the database. The second section shows the indexes that exist in the database but are not declared in the SAP ABAP/4 Dictionary. Indexes that are declared in the SAP ABAP/4 Dictionary but do not exist in the database can be created as follows: Position the cursor on the index and click on the button *Create in DB*.



Always create a new index within the ABAP/4 Dictionary, otherwise errors will be generated during the missing indexes check.

See also:

[R/3 Static Database Monitor \[Page 87\]](#)

Database Server Management

There are a number of basic tasks that have to be performed frequently when managing a database. The following topics give step by step instructions on how to perform certain fundamental tasks.

For information about other administrative tasks, please refer to the *SQL Server Books Online*.

[Starting and Stopping the SQL Server \[Page 58\]](#)

[Setting SQL Server Startup Options \[Page 60\]](#)

[Changing Passwords \[Page 61\]](#)

[Registering a Server \[Page 63\]](#)

[Changing the Autogrow Settings \[Page 64\]](#)

[Configuring the SQL Server \[Page 65\]](#)

[Configuring Databases \[Page 66\]](#)

[Setting Single-User Mode \[Page 67\]](#)

[Executing Queries \[Page 69\]](#)

[Moving R/3 Database Files to New Disks \[Page 71\]](#)

[Displaying the SQL Server Version \[Page 73\]](#)

Starting and Stopping the SQL Server

Starting and Stopping the SQL Server

Prerequisites

When you start up SQL server you actually start the Windows NT Service MSSQLServer and the corresponding process *sqlservr.exe*.

In addition to starting the server, you also need to start SQL Server Agent in order to be able to schedule and execute jobs within the Enterprise Manager. Other services offered by the SQL server are not required in the R/3 environment and should therefore not be started.

Overview of SQL Server Services

Service	Process	Comment
MSSQLServer	sqlservr.exe	Always required
SQLServerAgent	sqlagent.exe	Always required
MSDTC	msdtc.exe	Do not run in the R/3 environment
Microsoft Search	mssearch.exe	Do not run in the R/3 environment



You can view a list of all installed services and their current status in the NT Service Manager. Choose *Control Panel* → *Settings* → *Control Panel* → *Services*.

You can start the SQL Server using the SQL Server Service Manager or the Enterprise Manager.

Procedure

1. In the NT task bar, double-click on the server icon located on the right beside the time display.

The *SQL Server Service Manager* dialog box appears.

2. Select the server you want to start or stop.
3. Select the service *MSSQLServer*.
4. Choose either *start* or *stop* as required.

The server is started up or shutdown immediately.
During startup, an automatic recovery takes place.
During shutdown a checkpoint is performed in every database.



Never *Pause* a server. This option prevents any further connections being established to the server and may cause R/3 errors.

Repeat the procedure for the service *SQLServerAgent*. The *SQL Server Agent* must be started to allow the scheduling of tasks in the Enterprise Manager.



Starting and Stopping the SQL Server

Optionally, SQL Server and SQL Server Agent can be started automatically whenever Windows NT starts. If you wish to implement this option, use the NT Service Manager to specify that the services should be started together with Windows NT. For more information see the Windows NT documentation.

Setting SQL Server Startup Options

Setting SQL Server Startup Options

Use

When the SQL Server is started, parameters are read from the Windows NT Registry that influence how it is started. In the course of your administrative work, you may need to modify or check these parameters using the Enterprise Manager.

Procedure

1. In the Enterprise Manager, select the server and then choose *Action* → *Properties*.

The *Server Properties* dialog box opens.

2. On the General tab choose *Startup Parameters* button.

The *Startup Parameters* box appears. It shows all the currently set startup parameters.

3. To add a parameter, enter the name in the *Parameter* field and choose *Add*. To remove a parameter, select it and choose *Remove*.

Confirm your entries with *OK*.

To get information about available parameters, search for the topic "sqlservr application" in the *MS SQL Server Books Online*.



In the Registry, the parameters are located in:
HKEY_LOCAL_MACHINE/SOFTWARE/Microsoft/MSSQLServer/MSSQLServer/Parameters

Changing Passwords

Use

When R/3 and the SQL Server are installed, the users `sa`, and `sapr3` are created.

The user `sa` has system administration privileges for the SQL Server. Anyone who logs on as `sa`, can perform all the tasks available on the server without any restrictions. Initially no password is assigned to `sa`. It is therefore important to assign a password to this user directly after the installation of R/3 to prevent unauthorized use of the associated privileges. Before you can assign a password to `sa`, you have to make sure that the server involved has been registered.

The second user that is created during the installation, `sapr3`, is the owner (dbo) of the R/3 database. `sapr3` has system administrator privileges that allow the execution of all tasks on the R/3 database, including the definition of privileges for other users. Initially `sapr3` is assigned the password `sap`. To prevent inappropriate use of this user, change the password immediately after the installation.



The R/3 System connects to the database with the NT accounts `SapService<SAPSID>` or `<SAPSID>adm`. At this level, Windows NT authentication is used to ensure the security of the connection to the database and no password is required. However, when you initially log on to NT as `SapService<SAPSID>` or `<SAPSID>adm`, you need to enter a password. It is important to ensure the security of this password to protect the database against unauthorized access. If necessary, change the password following the instructions given in the Windows NT documentation.

Procedure

1. In the Enterprise Manager, expand the server and *Security*.
2. Under *Security* choose *Logins*.

The results pane shows the users `sa` and `sapr3`.

3. Select the user for which you want to change the password and choose *Action* → *Properties*.
4. The SQL Server *Login Properties* dialog box opens.
5. On the *General* tab, enter a new password for the user you have selected.
6. When you change the password for `sa`, the master database is specified as the default database for the login.

When you change the password for `sapr3`, the R/3 database `<SAPSID>` is specified as the default database for the login.

7. Choose *OK* and confirm your password.

The new password is now valid.

Changing Passwords



After changing the password of the login `sa`, you have to re-register the server in the Enterprise Manager.

Registering a Server

Use

Before you can administer a server with the Enterprise Manager, you have to register it in the Enterprise Manager. To register a server you need to specify the:

- Name of the server
- Security used to log on to the server
- Login name and password
- Name of the group where the server is to be installed

Procedure

The following assumes that you are not using the wizard to register the server.

1. In the Enterprise Manager, select a server group and choose *Action* → *New SQL Server Registration*.

The dialog box Registered SQL Server Properties appears.

2. In the text box *Server*, enter the name of the server.
3. Select the option *Use SQL Server authentication* to specify the type of authentication for logging on to the server. Enter *sa* as the *Login Name* and the corresponding *Password*.
4. Select the *Server Group* that the server is to be associated with.
5. Select any further options that you might require.



Set the option *Show system databases and system objects*. This makes the system databases such as *master* and *msdb* visible and enables you to administer them in the Enterprise Manager.

Confirm your entries with *OK*.

The SQL Server is now registered and will appear in the Enterprise Manager under the server group you have specified.



If you later wish to change the options you have selected, simply highlight the server in the Enterprise Manager and choose *Action* → *Edit SQL Server Registration Properties*. The *Registered SQL Server Properties* dialog box opens and you can edit the options as required.

Changing the Autogrow Settings

Changing the Autogrow Settings

Use

SQL Server 7.0 offers an autogrow option for files. This allows database files to automatically grow when they are full provided that enough space is still available on the disk. The amount a file can grow is determined by a growth increment that you can specify for each file. When autogrow is enabled, the work involved in monitoring the size of the database is reduced significantly. It no longer necessary to continually monitor the free space in files. It is however important to always keep track of the amount of free space on the disk. There must always be enough space to allow the next required files extension. If the disk with the R/3 database or log files is full, processing in the R/3 System stops. It is therefore essential to know well in advance, when a shortage of disk space will arise.

When R/3 is installed with the SQL Server, the R/3 database is normally created with three data files that each have a size of 2 GB and one log file that has a size of 1.5 GB. If the growth increment for the data files and log files is set to 100 MB, a file automatically grows by 100 MB when it is full.

Procedure

To change the autogrow settings for the different databases in your system, proceed as follows:

1. In the Enterprise Manager, expand the server and *Databases*.
2. Select a database and choose *Action* → *Properties*.
The *Database Properties* dialog box opens and displays the autogrow settings.
3. To change the settings for data files switch to the *General* tab. To change the settings for the log file switch to the *Transaction Log* tab.
4. In the *File growth* section, specify the amount the selected file must grow when it becomes full. You can specify file growth in MB or as a percentage of the total file size.
5. In the *Maximum file size* section, select *Restrict filegrowth* and enter the size the selected file can grow to, or choose *Unrestricted file growth*.
6. Confirm your entries with *OK*.

Configuring the SQL Server

Use

You can optimize the performance of the SQL Server by setting or adapting server configuration options. If you intend to change options, keep in mind that SQL Server 7.0 has internal features that enable it to manage and optimize resources automatically, depending on runtime conditions. Only change configuration settings, if you are sure they will benefit the performance of your system.

For recommendations on how to tune the SQL Server in the R/3 environment see [Database Performance Tuning \[Page 74\]](#).

The most commonly used server options can be modified directly in the Enterprise Manager. Other, more specialized options, can be set using the `sp_configure` stored procedure as described in the *SQL Server Books Online*. Also refer to the books to find out which options are self-configuring.



When server properties are modified this affects the entire server. If you want to adjust settings that only affect a particular database, you have to change the configuration of the database. For more information see [Configuring Databases \[Page 66\]](#)

Procedure

1. In the Enterprise Manager select the server and choose *Action* → *Properties*.

The *SQL Server Properties* dialog box opens. It offers a tab for each category of available configuration options.

2. Choose the tab for the resource you wish to configure.
3. Enter the information required. Choose *OK*.



Some settings only take effect after the server has been restarted. If a server restart is necessary, a dialog box appears asking you whether you wish to stop and restart the server.

Configuring Databases

Configuring Databases

Use

Database options allow you to configure each database individually. If you set options for a particular database this does not affect any of the other databases. The most commonly used configuration options can be set in the Enterprise Manager. Other, more advanced options, can be set with the stored procedure `sp_dboption` as described in the *SQL Server Books Online*.

When you set a database option, it immediately takes effect.

Procedure

1. In the Enterprise Manager expand your server and *Databases*.
2. Select a database. Normally this will be the <SAPSID> database. Choose *Action* → *Properties*.

The *Edit Properties* dialog box opens for the selected database.

3. Switch to the *Options* tab.
4. Select and deselect the options as required. For information on the individual options choose *Help*. Keep the following recommendations in mind:

<i>Truncate log on checkpoint</i>	This deletes the inactive part of the transaction log whenever a checkpoint occurs. In a production system, set this option for the <code>master</code> and <code>msdb</code> databases. No logs are required for these.
<i>Select into/bulk copy</i>	Set this for the <code>tempdb</code> . It allows special operations to be performed that are not logged.



Never set *Truncate log on checkpoint* or *Select Into/Bulk copy* for the R/3 database in a productive system. When *Truncate log on checkpoint* is set, the inactive part of the transaction logs is deleted whenever a checkpoint is processed. This means logs are deleted before they can be backed up and will therefore not be available when a damaged database has to be restored.

Select Into/Bulk switches off logging for special operations. It is used to prevent the logs from filling up when large amounts of data are transferred.

If one of the parameters has been set, you need to deselect it and backup the complete database immediately.

Setting Single-User Mode

Use

Single-user mode can be set for the entire SQL Server or for individual databases using the Enterprise Manager. Once this mode has been set, only one connection to the SQL Server or specified database is allowed. Some administrative tasks can only be performed if single-user mode has been set.

Procedure

Single-User Mode for the SQL Server

1. Before you set single-user mode for the server make sure no users are connected to the server.
2. In the Enterprise Manager select the server.
3. Choose *Action* → *Properties*.

The *SQL Server Properties* dialog box opens.

4. On the General tab choose Startup parameters.

The *Startup Parameters* box appears.

5. Enter the parameter `-m` and choose *Add*.

The new startup parameter is displayed in the *Existing Parameters* box.

6. Press OK to confirm the entry and restart the server.

Only one connection to the server is now possible.



When single user mode has been set, only one connection to the server is permitted. Therefore, keep the following in mind:

If the SQL Server Agent is running it occupies the only connection available to the server and you will be unable to start the Enterprise Manager or Query Analyzer. Stop the agent to enable a connection to the Enterprise Manager.

If you are working with the Enterprise Manager and want to start the Query Analyzer, you first have to close the Enterprise Manager. Then start the Query Analyzer via the NT Start menu.

Single-User Mode for a Database

1. Before you set single-user mode for the database, make sure no connection to the database is open. For example, if the Query Analyzer is active, close it.
2. In the Enterprise Manager, expand the server. Expand *Databases*.
3. Select the <SAPSID> database and choose *Action* → *Properties*.

The *Properties* dialog box appears.

4. On the *Options* tab select *Single user* and confirm your entry with *OK*.

Setting Single-User Mode

Once single user-mode has been set, only one connection to the database is permitted. Each utility, including the Enterprise Manager, Query Analyzer and isql opens a separate connection to the server.

Executing Queries

Use

There are three utilities that can be used to execute Transact-SQL statements and stored procedures:

- The SQL Server Query Analyzer
- isql utility
- osql utility

Of these utilities, the SQL Server Query Analyzer is the most frequently used. It has a graphical user interface and is therefore easy to use. In addition it offers quick access to help on the syntax of commands.

isql and osql are command line utilities. They have the advantage that they allow the specification of input and output files for scripts and therefore allow you to automate and schedule the execution of scripts.



Always use osql. The isql utility is included in SQL Server 7.0 for compatibility reasons. It does not support all SQL Server 7.0 features.

Procedure

The SQL Server Query Analyzer

1. To start the Query Analyzer in the Enterprise Manager, select your server and choose *Tools* → *SQL Server Query Analyzer*.

Alternatively, if the Enterprise Manager is not running, you can start the *Query Analyzer* from the Windows NT *Start* menu. Choose *Programs* → *Microsoft SQL Server 7.0* → *Query Analyzer*

The *SQL Server Query Analyzer* window opens.

2. In the database field, select the database on which you want to execute a query.
3. Enter the query. To execute the Query choose *Query* → *Execute*.

The results are displayed in a second pane that appears.



If you need help on the syntax of a statement, simply highlight the keyword and press `SHIFT F1`. This takes you directly to the *Transact SQL-Help* that gives detailed information on the statement and illustrates it with examples.



A convenient feature allows you to type in several queries on the screen, but only to execute one of them. Simply highlight the statement you wish to execute before choosing the execute icon.

Executing Queries

The osql Utility

The osql utility allows you to execute Transact-SQL statements, stored procedures and script files. The utility is started in the command prompt.

To execute a script with the osql utility enter a command as follows:

```
osql -S<server_name> -d<SAPSID> -Usa -P<password> -i<script_name>  
-o<output_file>
```

For details refer to the topic "osql utility" in the *MS Books Online*.

Moving R/3 Database Files to New Disks

Use

If your R/3 database or log files are running out of disk space, you have to add new disks to the file system. Depending on the disk configuration currently in use, two different approaches are possible:

- You might be able to add a new disk to an existing RAID array. Some RAID controllers allow you to add a disk online. However, normally, you have to recreate the the RAID system to include the new disk. To do this, you first have to backup the existing database, then re-create the RAID system and, in final step, restore the database. When the array has been re-created, the additional space on the new disk will be accessible for the operating system and can be utilized by the SQL Server. For instructions on how to reconfigure an array to include new disks, refer to the documentation provided by your disk vendor.
- If an existing RAID system cannot be extended, you have to add an entire new disk array to your system. In this case, it is necessary to first configure the new array as an NTFS partition. Then existing R/3 database files have to be moved to the new disk in a process that involves detaching and re-attaching the database from the SQL Server.

The following describes the procedure for moving files to a new disk array. It assumes that you have already configured the array as an NTFS partition according to the instructions provided by the disk vendor.

Procedure

1. In the Enterprise Manager select the server and choose *Tools* → *SQL Server Query Analyzer*.

The *Query* window of the *SQL Server Query Analyzer* opens.

2. Detach the database from the SQL Server with the following stored procedure:

```
sp_detach_db '<SAPSID>' 'true'
```

Where <SAPSID> is the name of your R/3 database.

3. Execute the stored procedure with *Query* → *Execute*.

The R/3 database is detached from the SQL Server and is no longer visible in the Enterprise Manager when the display has been refreshed.

4. Switch to the Windows NT Explorer and copy the R/3 database files to the newly configured disk drive as required.



Do not move the primary data file <SAPSID>DATA1 .mdf to the new drive. If you move it, the *master* database will be modified. As a result, the current backup of the *master* database will no longer be up to date and you will have to make a new backup immediately.

5. When you have moved the files to the new drive, re-attach the R/3 database to the SQL Server. To do this, switch to the Query Analyzer and execute the *sp_attach_db* stored procedure. You must specify the location of all the database files, including the log file:

Moving R/3 Database Files to New Disks

```
sp_attach_db <SAPSID>, '<filename>', '<filename>'...
```



```
sp_attach_db PRD, 'D:\PRDDATA1\PRDDATA1.mdf',  
'N:\PRDDATA2\PRDDATA2.ndf', 'N:\PRDDATA3\PRDDATA3.ndf',  
'L:\PRDLOG1\PRDLOG1.ldf'
```

The R/3 database reappears in the database list of the Enterprise Manager and is again visible and accessible for the SQL Server. The newly configured disk array can now be used by the SQL Server.

Displaying the SQL Server Version

Use

When problems arise in your system, it is always important to check whether the system is up to date and all available service packs have been imported.

Procedure

To find out the version of your SQL Server, look at the first line of the current *SQL Server Error Log*. To view the log proceed as follows.

1. In the Enterprise Manager, expand the server, *Management* and *SQL Server Logs*.

A list of all the available logs is displayed in the right-hand pane.

2. Choose the current log.

The contents of the log are displayed in the right-hand pane.

3. Find the first line of the log. It shows the exact version of the server. For example:

```
Microsoft SQL Server 7.00 - 7.00.xxx
```

The last 3 digits (xxx) indicate the *build number* which designates a certain Service Pack level.

Database Performance Tuning

Database Performance Tuning

If you notice that the response times of your system are too slow and you are dissatisfied with the throughput, there are a number of NT and SQL settings you can adapt manually to improve the performance. Changing these settings can be of benefit not only to a system experiencing problems, but also to a system running satisfactorily. By making the recommended adjustments, you can effectively avoid bottlenecks at peak processing times and ensure continual smooth operation of your system.

NT Configuration

SAP recommends the following adjustments:

- [Increase the size of the NT page file \[Page 75\]](#)
- [Optimize the Virtual Memory Manager \[Page 76\]](#)
- [Remove the screen saver \[Page 77\]](#)

SQL Server

SQL Server 7.0 optimizes many configuration settings automatically. This means configuration options are dynamically adapted to meet requirements at runtime. For example, the allocated memory and lock resources are increased or decreased depending on the nature of the current workload. Despite the performance benefits provided by automatic tuning, it is nevertheless still worthwhile to manually adjust some settings to improve the throughput of the system in the R/3 environment.

For recommendations on settings for the SQL Server, refer to the R/3 Note 126808.

Increasing the Size of the NT Page File

Procedure

1. From the Windows NT Start menu choose *Settings* → *Control Panel*
2. Choose *System*.
The *System Properties* dialog box opens.
3. On the *Performance* tab, in the *Virtual Memory* section, choose *Change*.
4. In the *Initial Size field*, enter the size of the paging file in MB. This must be at least 1 GB. Ideally it should be three times the size of the RAM installed on the server.
5. Confirm your entries with *OK*.

Optimizing the Virtual Memory Manager

Procedure

The following describes the procedure for the NT Server 4.0.

1. From the Windows NT *Start* menu choose *Settings* → *Control Panel*.
2. Choose *Network*.
The *Network* dialog box opens.
3. On the *Services Tab* choose *Server* and then click on the *Properties* button.
4. Select *Maximize Throughput for Network Applications*.
5. Confirm your selection with *OK*.

Removing the Screen Saver

Procedure

1. From the Windows NT *Start* menu choose *Settings* → *Control Panel*.
2. Choose *Display*.
The *Display Properties* dialog box opens.
3. Switch to the *Screen Saver* tab. Select *None* as screen saver.
4. Choose *OK* to activate the new setting.

Database Management Tools

Database Management Tools

The SQL Server and the R/3 System provide useful tools that facilitate database management. Some of the most commonly used tools are:

- [Enterprise Manager \[Page 79\]](#)
- [R/3 Planning Calendar \[Page 81\]](#)
- [R/3 Backup Monitor \[Page 85\]](#)
- [R/3 Static Database Monitor \[Page 87\]](#)
- [SQL Server Alert Monitor \[Page 88\]](#)
- [R/3 SQL Server Database Monitor \[Page 93\]](#)

Enterprise Manager

Use

The Enterprise Manager is a graphical tool included in the Microsoft SQL Server that provides a user-friendly way to administer SQL Servers. It is useful for performing the majority of daily administrative tasks.

Prerequisites

Before you can access a database server with the Enterprise Manager for the first time, you must register the server. To register the server, you need a login ID and a password. Normally the database administrator uses the login ID *sa*. This enables full access to each database on the registered server.

Features

With the SQL Enterprise Manager you can perform almost all of the tasks required to administer a database. These are comprehensively documented in the *SQL Server Books Online*.

In an R/3 environment, you can perform the following tasks with the Enterprise Manager:

- Start and stop the SQL Server
- Configure the SQL Server
- Configure databases
- Back up and restore databases
- Create and manage jobs
- Create and manage database objects
- Monitor current server activity
- View the SQL Server error log
- View the SQL Server Agent error log
- Manage local and remote servers



This SAP documentation includes instructions on how to perform certain tasks in the Enterprise Manager, but does not intend to replace the *SQL Server Books Online*. It focuses on some central tasks that frequently need to be executed in an R/3 environment and supplements explanations with recommendations based on the experience gained from R/3 on SQL Server implementations.

Menu Options

The Enterprise Manager allows you to access options in different ways. The starting point is normally the hierarchy of server components displayed in the left-hand pane of the Enterprise Manager. When the hierarchy is fully expanded, you can select a component and then choose an action to execute for the component. The actions available for a selected component can be accessed in different ways:

Enterprise Manager

- From the *Action* menu

All the options available for a particular component are listed in the main menu under *Action*. As soon as you change your selection, for example to <SAPSID> database, the list of options offered changes dynamically.
- From the context menu

When you click on a component in the hierarchy with the right mouse button, all options available for that specific item are listed in a popup menu. The menu appears directly beside the selected component and can therefore be very quickly and easily activated.
- From the pane on the right-hand side.

This approach allows you to choose actions while you are navigating through information displayed in the right hand pane. For example, if you have selected a database and are viewing information about free space, you can immediately choose to expand the database. The option you need is displayed directly beside the information and you simply need to click on it.

This method of choosing to perform an action is probably the most straightforward. However, only a subset of functions is offered, so you will not always see the option you require.

This documentation explains actions using the main menu. It is however advisable to experiment with different methods to find out which suits you best.

Query Analyzer

Although the Enterprise Manager is primarily a graphic tool that enables you to perform tasks by navigating through a series of menus and dialog boxes, there are many situations where direct access to the database is desirable. The Query Analyzer that can be started from the Enterprise Manager allows you to access the database directly with SQL Statements and stored procedures. You can perform all database operations by entering queries, for example, you can read and modify data or create and drop tables.

See also:

[Database Backup \[Page 7\]](#)

[Database Restore \[Page 32\]](#)

[Database Server Checks \[Page 47\]](#)

[Database Server Management \[Page 57\]](#)

R/3 Planning Calendar

Use

The Planning Calendar is an R/3 tool that automates the execution of important database administration tasks. Its central feature is a calendar that covers the entire screen and records all the tasks that have been performed or scheduled for the database system. The calendar is particularly useful to:

- Schedule important tasks that need to be repeated periodically, for example, backups or consistency checks
- Obtain an overview of all performed or scheduled tasks and their status

Once a task has been planned, it is automatically executed at the time specified without requiring any additional attention. When it has completed, the calendar indicates any problems that occurred by highlighting it in red and providing a brief task history.

To access the Calendar choose: *Administration* → *CCMS* → *DB Administration* → *DBA Planning Calendar*. Alternatively, enter the transaction code DB13.

Features

The Calendar provides options to schedule the following actions:

- Full database backup
- Differential database backup
- Transaction log backup
- Statistics update
- Database consistency check
- Tables consistency check

These actions can be scheduled to run on their own; once or repeatedly. Alternatively, they can be scheduled to run in combination with other tasks by selecting a predefined *Planning pattern*. A *Planning pattern* specifies a combination of tasks that are executed at predefined intervals and can be scheduled together. The Calendar offers the *Planning pattern*:

5 workdays, DB backup + hourly log backup (8am - 6pm) + Monthly Check DB

To schedule the pattern, you simply need to choose it and specify a time. The system automatically assigns values to the parameters required for the execution.

The *Planning patterns* do not necessarily provide the best solution for the systems. The requirements of an individual system must be considered carefully before you decide to implement a predefined strategy.

Activities

You can use the Planning Calendar to:

- Schedule a new action

You can select the day on which the action is to be started and then specify how the action is to be performed by assigning values to the parameters offered. You can

R/3 Planning Calendar

schedule actions to run periodically by specifying a time interval in which they should be repeated (in weeks).

- Change a scheduled action
You can modify or delete scheduled actions.
- Display results of executed actions
You can see the status of a completed action and view a brief task history.
- Display or change parameters
You can view and edit the parameters that have been set for an action.
- Execute an action immediately
You can start scheduled actions immediately, for example, to repeat planned actions that have failed.

See also:

[Working with the R/3 Planning Calendar \[Page 83\]](#)

[Database Management Tools \[Page 78\]](#)

Working with the R/3 Planning Calendar

Use

The following explains how to use the R/3 Planning Calendar to perform basic database administration tasks. It explains how to:

- Modify scheduled actions
- Delete scheduled actions
- Display information about scheduled actions

Procedure

1. To start the Planning Calendar, choose *Tools* → *Administration* → *Computing Center* → *Management System* → *DB Administration* → *Time Scheduling*.

Alternatively, enter the transaction code DB13.

A window showing a calendar with the days of the week opens.

2. Double-click on the header-bar of the day that is of interest to you.

A list of all the actions scheduled on that day and their status is displayed.

3. Select a scheduled action and choose the appropriate button, depending on whether you simply want to display information or wish to modify or delete an action.

If an action has already been executed, only the *Parameters* and *Action Log* buttons are available.

Working with the Planning Calendar

Action	Choose Button	Comments
Display parameter settings	<i>Parameters</i>	The current parameter settings are displayed in a series of windows, but cannot be changed. To view all settings, confirm each window that appears with OK.
Display volumes assigned for a backup	<i>Volumes needed</i>	The name of the tape device and the volume that the backup will be written to are displayed.
Change parameter settings	Choose <i>Change</i> and then <i>Parameters</i> in the following window	The parameters are displayed in a series of windows and can be modified as required. To proceed to the next available window, confirm your entries with OK.

Working with the R/3 Planning Calendar

Delete scheduled action	<i>Delete</i>	The selected action is deleted. If the task you wish to delete is scheduled periodically, for example, in the framework of a planning pattern, then deleting it on a particular day of the week also deletes it on the same day of all subsequent weeks.
View results of an executed task	<i>Action logs</i>	Information about the task and its execution are displayed. The information includes the time taken to complete the task and whether the task was successful or not.

See also:[Executing Backups \[Page 21\]](#)[Monitoring Backups \[Page 27\]](#)[Checking Consistency of the Database \[Page 53\]](#)[Database Management Tools \[Page 78\]](#)

R/3 Backup Monitor (MS SQL Server)

Use

The Backup Monitor is an important source of information for the database administrator. It collects and summarizes information related to backups, database restore operations and tasks scheduled in the Planning Calendar. It is particularly useful for monitoring the backup strategy that has been implemented in a system.

Prerequisites

Access the backup monitor with:

Administration → *CCMS* → *DB Administration* → *Backup logs*

Alternatively, enter the transaction code DB12.

Features

On the initial screen of the monitor you can, at a glance, get a quick overview of important information related to backups. It shows the last successful backups for the R/3, master and *msdb* databases and R/3 transaction logs. It also indicates the amount of log space allocated and how much is still free.



When you look at the amount of free space specified for logs, keep in mind that the data displayed reflects snapshots taken at a specific point in time. As logs are backed up and truncated at hourly intervals, a large amount of free space might be displayed because the log has just been truncated. For more information see [Monitoring the R/3 Log Disk \[Page 52\]](#).

Further functions available enable you to access technical details about past backups, restore operations and other actions planned in the DBA calendar:

Backup Monitor Options

<i>Backup history</i>	Displays technical details about backups that have been executed. The date and time of execution, size of the backup and the name of the physical device the dump was written to are displayed. The pushbuttons in the menu let you choose different views. For example <i>Logs Backups</i> only displays the log backups.
<i>Restoration history</i>	Gives an overview of all the restore operations performed in the system. Full technical details are specified, including the size and type of dump restored, the date when the dump was created, the name of the device and volume used.
<i>SQLAgent Properties</i>	Technical data about the Agent
<i>Backup device list</i>	Lists the backup devices available in the system. It shows whether the devices are disks or tapes, gives their logical names and their corresponding physical location.

R/3 Backup Monitor (MS SQL Server)

<i>Tapes needed for restore</i>	Gives an overview of all the backups that are available for a restore operation together with all relevant technical details such as the time of the backup, the size, the tape label and so on.
<i>SQL Server jobs</i>	Displays technical details of all the jobs scheduled for the SQL Server. Options in the menu allow you to focus on a subset of jobs, for example, the <i>CCMS jobs</i> .

See also:

[Database Management Tools \[Page 78\]](#)

R/3 Static Database Monitor (MS SQL Server)

Use

The R/3 Static Database Monitor displays information about the current status of the database. In addition it offers a number of functions that execute checks or return detailed information about the database.

Prerequisites

To access the R/3 Static Database Monitor, enter the transaction code DB02.

Features

The initial screen of the monitor displays information on the status of the database grouped in four different categories: *Database System*, *Database*, *Files*, and *Tables Indexes, Stored Procedures*. The date and time when the information was generated is shown at the top of the screen. A *Refresh* button is provided to update the information.

Information on the Initial Screen

Screen Area	Information
<i>Database System</i>	Name of the database and the time when the information on the screen were generated
<i>Database</i>	Space analysis for the R/3 database and transaction logs
<i>Files</i>	Lists the files of the R/3 database, their location and size and other data.
Tables, Indexes, Stored Procedures	Specifies how many of these exist in the system and their size.

Options

Pushbutton	Function
DB Space history	Displays the history of space usage in the system distinguishing between space allocated to tables and space allocated to indexes.
DB analysis	Offers a number of options to execute checks on the database.
Detail Analysis	Executes a consistency check for a specified table and returns detailed information about the table.

See also:

[Checking Consistency of Individual Tables \[Page 55\]](#)

[Checking for Missing Indexes \[Page 56\]](#)

[Database Management Tools \[Page 78\]](#)

[R/3 SQL Server Database Monitor \[Page 93\]](#)

SQL Server Alert Monitor

Use

The *Alert Monitor* for the SQL Server enables you to get a quick overview of important information that is relevant for the operation of the database. It draws attention to existing and potential problems with colored alerts and offers easy access to in-depth information about the system.

Prerequisites

To access the *Alert Monitor* for the SQL Server, you choose *Administration* → *CCMS* → *Control/Monitoring* → *Alert monitor*. In the monitoring tree, you choose *SAP CCMS Monitor Templates* → *Database*.

Features

Information in the monitor is presented in the form of a tree that can be expanded to get a quick overview of the current status of the database. When a given threshold value defined for the database is exceeded, this is shown by alerts that are red or yellow. Red indicates an existing problem that must be resolved and yellow a situation which could develop into a problem.

The information provided in the tree can be viewed from different perspectives.

- The *Current Status* view shows the most recent system alerts.
- The *Open Alerts* view informs you of the total number of alerts open for each category of information. It also offers a *Display Alerts* option which allows you to access a list of open alerts for a selected element of the monitoring tree.

In addition, a further option allows you to start an *Analysis Tools* which accesses comprehensive information about a selected aspect of database or system performance.

Categories of Information

Information and alerts for SQL Server in the monitoring tree are grouped into the following main categories:

- Space Management
- Performance
- Backup Restore
- R/3 Consistency
- Health

For more information on these categories see [Alerts for the Database Server \[Page 89\]](#)

For instructions on how to handle the R/3 Alert Monitor see [Detailed Tutorial \[Ext.\]](#)

Alerts for SQL Server

Definition

Alerts for the database server indicate when threshold values that have been defined for the SQL Server have been exceeded. Existing threshold values reflect SAP recommendations for the database server, however they can be adjusted to meet individual system requirements.

See also [Viewing and Customizing Threshold Values \[Page 92\]](#).

Use

The database alerts, together with the R/3 monitoring functionality, help to optimize database administration. They reduce the amount of work involved by routinely checking the system for significant events and drawing attention to problems as soon as they arise. The amount of time that has to be invested for routine checks is significantly reduced and system inconsistencies can be eliminated more quickly.

Alerts marked in red indicate that there is a system problem that must be rectified. Alerts marked in yellow are warnings that indicate system events that might develop into problems later, if they are ignored.

Structure

The database alerts are displayed in the *R/3 Alert Monitor* template for the SQL Server. Together with other essential system information, the alerts are presented in the form of a tree that is subdivided into the following main categories:

- Space Management
- Performance
- Backup Restore
- R/3 Consistency
- Health

The information provided can be viewed from different perspectives that offer varying levels of detail. The following tables summarize the information that is displayed in the *Current Status* view which is opened when you initially access the monitor. This view displays only the most recent alerts.

For information on other views, see [The SQL Server Alert Monitor. \[Page 88\]](#)

To find out the threshold values which trigger alerts in the system, see [Viewing and Customizing Threshold Values \[Page 92\]](#).

Space Management

Status information and alerts related to disk space that is available for the server are displayed for both the <SAPSID> Database and the *tempdb* database. For each physical database file you can check the following:

Alert	Information Displayed
-------	-----------------------

Alerts for SQL Server

Autogrowth	Autogrowth setting. When autogrowth is enabled, the pre-allocated file size is automatically increased by a predefined number of megabytes whenever required.
Stats in MB	Space currently allocated for the file.
	Space already used for the file.
	Growth setting which is active. This is the amount of space automatically allocated to the file when it fills up.
Free disk space	The amount of free space left on the device where the file is located.

The values are refreshed every 8 hours.

Double-Clicking one of the alerts takes you to the [Static Database Monitor \(transaction DB02\)](#).
[\[Page 87\]](#)

Performance

Factors which influence database performance and are automatically checked by the alert monitor are disk I/O, cache hit ratios and CPU usage.

Alert	Information Displayed
Disk Statistics	Double-clicking on the node displays information on the disk with the slowest response times and swap space.
CPU used by SQL	Percentage of CPU used by the SQL Server.
Data hit ratio	The total number of requests satisfied by the data cache divided by the total number of data cache accesses.
Procedure hit ratio	The total number of requests satisfied by the procedure cache divided by the total number accesses of the procedure cache.

The values are refreshed every 4 minutes.

Double-clicking one of the alerts takes you to the [Operating System Monitor \[Ext.\]](#) (transaction ST06).

Backup Restore

Information in this category assists the database administrator in determining whether the backup strategy is being implemented successfully.

Alert	Information Displayed
R/3 DB status	Date of last successful backup.
master DB status	Date of last successful backup.
msdb DB status	Date of last successful backup.
Backups running	Backups currently in progress.
Minutes since last backup	Time elapsed since the last backup.

Backup dates are refreshed once a day. The transaction log backup is checked every 4 minutes.

Double-clicking on an alert accesses the [R/3 Backup Monitor \[Page 85\]](#) (transaction DB12)

R/3 Consistency

This category of information shows the objects that are defined in the ABAP/4 Dictionary, but do not exist on the database.

Alert Item	Information Displayed
Tables	Number of tables defined in the ABAP/4 Dictionary that do not exist on the database.
Views	Number of views defined in the ABAP/4 Dictionary that do not exist on the database.
Indexes	Number of indexes defined in the ABAP/4 Dictionary that do not exist on the database.

Values are refreshed once a day.

Double-clicking on an alert accesses the [Static Database Monitor \[Page 87\]](#) (transaction DB02).

Health

The health of the system can be judged by monitoring the effectiveness of parameter settings and watching out for a number of common errors.

Alert	Information Displayed
SQL Server trace flags	Trace flags that have been set. Trace flags can be used to analyze different aspects of the SQL Server.
SQL Server memory setting	The type of SQL server memory setting that is active. Memory can have a fixed setting or vary within predefined limits.
Setup error	Error occurs when statistics are automatically computed for the VBHDR, VBDATA or VBMOD tables. In a SAP System, statistics for these tables must be disabled by setting the parameter sp_autostats to off.
Disk I/O error	Number of disk read or write errors that have been detected.
Network packet error	Errors detected during the transmission of data packets through the network.
Error log messages	Total number of error messages that have a severity level of 17 or higher. Errors of these levels must be resolved by the system administrator.

Double-clicking on an alert takes you to the [SAP/SQL Server Database Monitor \[Page 93\]](#) (transaction ST04).

Viewing and Customizing Threshold Values

Viewing and Customizing Threshold Values

Purpose

In the *R/3 Alert Monitor* for the SQL Server, alerts draw attention to critical system events or situations that may develop into problems later. The alerts are triggered when threshold values that are predefined for the database are exceeded. These values are preset and based on SAP recommendations for the SQL Server, but can be changed to meet individual system requirements. The following explains how you can view and change threshold values.

Process Flow

1. Start the R/3 Alert monitor with *Administration* → *CCMS* → *Control/Monitoring* → *Alert monitor*. In the monitoring tree, choose *SAP CCMS Monitor Templates* → *Database*.
2. Expand the monitoring tree and select the alert for which you want to view or change a threshold value. Choose *Customizing*.
The customizing screen appears.
3. Choose the *Thrshld* tab.
The threshold values that are currently active for the monitor are displayed.
4. To change a value, choose *Customizing* → *Display* <-> *Change*.
5. Make changes as required and save your entries.

SAP/SQL Server Database Monitor

Definition

With the R/3 Database Monitor for SQL Server, you can display the important parameters and performance indicators in SQL Server, such as database size, database buffer quality, and database configuration.

Choose *Tools* → *Administration* → *Monitor* → *Performance* → *Database* → *Activity*. Alternatively, use transaction code ST04.

The following functions let you see changes that occur in a short space of time:

<i>Reset</i>	Temporarily sets the displays in the Server Engine/Elapsed and SQL Requests sections to zero.
<i>Since Reset</i>	Displays the values in the Server Engine/Elapsed and SQL Requests sections since you reset the display.
<i>Refresh</i>	Returns the main screen to the current values.
<i>Totals per Second/Absolute Totals</i>	Toggles the values in the Server Engine/Elapsed and SQL Requests sections

The information on the main screen *Database Performance Analysis* is divided into the following sections:

The top section of the screen displays information about the database version and hardware. The *DB startup* field shows the date and time when SQL Server was started. Your analysis of the information in the Database Monitor is only meaningful if the database has been running for several hours with a typical database workload.

CPUs available for SQL Server is particularly important, as it shows the number of CPUs installed in the computer, and the number of CPUs used by SQL Server. The CPUs that are available for SQL Server are determined in the SQL Server configuration. See also the SQL Server option *affinity mask*.

The remainder of the screen is divided into the following sections:

Memory Usage [Ext.]	Overview of memory configuration and performance.
Space Usage [Ext.]	Shows disk space availability for the R/3 database.
Server Engine/Elapsed [Ext.]	Shows CPU consumption and disk I/O.
SQL Requests [Ext.]	Overview of SQL Server request statistics.

See also [Detail Analysis Menu \[Ext.\]](#).