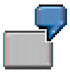
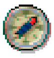
Computing Center Management System (BC-CCM)

HELP.BCCCM

Release 4.6C

Copyright

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax
	Tip

Contents

Computing Center Management System (BC-CCM)

Computing Center Management System (CCMS)

Purpose

The Computing Center Management System (CCMS) allows you to monitor, control, and configure your SAP System.

The CCMS tools support 24-hour unattended system management functions from within the SAP System. Using these tools, you can analyze and distribute client workloads and report on resource consumption for system components.

Features

The CCMS provides a series of monitors and management utilities for:

- Starting and Stopping the SAP System [Seite 29]
- Unattended 24 hour system management using instances [Seite 229] and operation modes [Seite 192]
- System monitoring [Seite 814] and automatic reporting of alerts
- Dynamic logon load balancing [Seite 174]

- SAP System configuration: Profiles [Seite 240]
- Processing and controlling background jobs [Seite 74], scheduling database backups

See also:

Setting Up the CCMS [Seite 26]

Setting Up the CCMS

Use

The Computing Center Management System (CCMS) is shipped to every customer site pre-configured so that all internal functionality is intact. The consistency and/or accuracy of this functionality depends heavily on how the customer first configures the operation mode [Seite 192] environment.

Prerequisites

To work with the CCMS, you require authorization `S_RZL_ADM`.

`S_RZL_ADM 1` gives you administrator authorization

`S_RZL_ADM 3` gives you display authorization.

If operation modes, instances, or the timetable are not correctly defined, the CCMS will not display meaningful data. For example, the job scheduling monitor will not be able to determine where background processes are running, the alert monitor may report improper operation of modes.

Before you can work with the CCMS, it must be set up correctly. To ensure the data accuracy of the system being monitored, you must define at least one operation mode and maintain the timetable.

If no operation modes have been defined, the test operation mode DUMMY will be displayed. This operation mode is configured automatically so that system functions such as the control panel [Seite 956] and background job scheduling [Seite 74] can be used. The operation mode DUMMY cannot be used for operation mode switching.

Existing operation modes are listed as they were previously defined. For example, you may see one or more operation modes for daytime operation and one or more for nighttime operation.

Procedure

1. To set up the CCMS, choose *CCMS* → *Configuration* → *Operation modes/instances*. Alternatively, call Transaction RZ04.
2. The system displays the *CCMS:Maintain Operation Modes and Instances* screen. From here you can create and change operation modes and instances.

See also:

Computing Center Management System (CCMS) [Seite 24]

Defining Operation Modes [Seite 211]

Operation Mode Switches [Seite 194]

Creating an Instance Definition for All Servers [Seite 234]

Creating an Instance Definition for One Server [Seite 232]

Maintaining Instance Definitions [Seite 238]

Starting and Stopping the SAP System

Purpose

Starting an SAP System involves starting the database of the system and the application servers and associated processes that make up the system. These associated processes may include the following:

- CPI-C gateway server
- System log collection process, `syslog` (not applicable in systems running on Windows NT hosts)
- Spool processes
- Performance collectors (operating system and/or network performance collectors)

In the simplest case, an SAP System consists of only a single application server.

Process Flow

A start profile determines which processes are to be started or stopped. This profile is generated automatically when you install your system, but you can modify it later.

You can start and stop the SAP System and its database with operating system tools. Special scripts are available for this or you can use the appropriate program icons.

You can start and stop SAP instances with the Computing Center Management System (CCMS) tool. To do this, the database and at least one SAP instance must have been started.

See also:

Starting an R/3 System: UNIX [Seite 52]

Stopping an R/3 System: UNIX [Seite 54]

Starting an R/3 System: Windows NT [Seite 30]

Stopping an R/3 System: Windows NT [Seite 39]

Starting the R/3 System: AS/400 [Seite 59]

Stopping the R/3 System: AS/400 [Seite 60]

Checking the Operation Mode with the Control Panel [Seite 225]

Profiles [Seite 240]

Starting an R/3 System: Windows NT

The following describes how to start the R/3 System or individual instances using the *SAP Service Manager*. You can also perform these tasks using the *Microsoft Management Console (MMC)*. SAP recommends the use of the MMC as it provides many new functions and simplifies system management. For more information, see Microsoft Management Console: Windows NT [Seite 1090].

To start an R/3 System that only consists of one central instance:

1. Log on to Windows NT as the SAP administrative user (`<SAPSID>ADM`, for example `C11ADM`).

   The R/3 service (`SAP<SAPSID>_<Instance number>`, for example `SAPC11_00`) is always started automatically when the Windows NT system is booted. The R/3 administrator does not therefore need to perform any additional actions to start the service. The information in the window *Services* (*Control Panel* → *Services*) allows you to check that the R/3 service has been started (if this is the case, the service has the *Status Started*).

   If the R/3 service has not been started, carry out the steps described in R/3 System on Windows NT: Problem Analysis [Seite 48].

2. Start the R/3 System:

   – Select *Programs* → *Administrative Tools (Common)* → *SAP* → *SAP Service Manager for <SAPSID>_<Instance number>* for the R/3 instance you wish to start. This takes you to the SAP Service Manager [Seite 34] window.

   – In the *SAP Service Manager* window, select *Start* to start the instance.

     The start procedure starts the database system as well as the central R/3 instance. If the database system is already running, only the instance is started.

     When the stoplights for all processes shown in the *SAP Service Manager* window are "green," the R/3 System is started and is ready to be logged on to. The R/3 System continues to run even if you close the window or log off of the Windows NT system.

   If a process did not start correctly, analyze the problem as described in R/3 System on Windows NT: Problem Analysis [Seite 48]. Stop the instance, correct the error, and restart the instance. It is not possible to restart an individual process.

To start an R/3 System that consists of several instances:

1. Start the database and at least the central R/3 instance. To do this, proceed as described in the section *To start an R/3 System that only consists of one central instance*.

2. You can start additional instances with the *SAP Service Manager* or with the CCMS from the R/3 System.

   To reach the CCMS start-up functions, select *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *System Monitor* in the R/3 main menu.

   See also: Starting and Stopping Instances [Seite 69]

If problems occurred, analyze these as described in R/3 System on Windows NT: Problem Analysis [Seite 48], correct the error, and repeat the start procedure.

See also:

SAP Service Manager [Seite 34]

Start and Shutdown Processing under Windows NT [Seite 43]

Example: Script for Starting the R/3 System

```
@rem echo off
REM
REM   substitute following strings:
REM      string to subst    substitution value  typical values
REM        <sapsid>       SAP R/3 System name, e.g., C11 case sensitive
REM        <central>      your central host's host name
REM        <app>          your applications server's host name
REM        <instance>     the instance name, e.g., DVEBMGS00
REM        <instance1>    the app server's instance name, e.g., D00
REM        <no>           the central system's SAPSYSTEM, e.g., 00
REM        <no1>          the app server's SAPSYSTEM, e.g., 00
REM
REM  for your values see the definitions in the profiles.
REM  all profile names and values including the hostnames are
cASeSenSiTivE!!!
REM

REM   check the users which will definitely not work
if %USERNAME% == "SYSTEM" goto bad_user

REM   first check existence of environment setup file
if not exist <sapsid>env.cmd
\\<central>\sapmnt\<sapsid>\sys\exe\run\ntreg2cmd <sapsid>

REM setup the environment for the process so that he is able to
REM run also from poor telnet daemons and Schedule Service
call <sapsid>env.cmd

REM STARTUP FOLLOWING

REM DATABASE startup (not necessary, normally done through
REM service) for example    sapdba -startup
sapdba -startup
```

```
REM start central instance service here...

ntscmgr start SAP<sapsid>_<no> -m <central>


REM you may also want to start additional application server services
here...

REM if so use : ntscmgr start SAP<sapsid>_<no1> -m <app>


REM if Autostart=1 is specified in the startup_profile you need

REM not to execute the next two lines


REM start the central instance

sapstart
pf=\\<central>\sapmnt\<sapsid>\sys\profile\start_<instance>_<central>
SAPDIAHOST=<central>


REM you may want to startup additional application servers here...

REM if so use : sapstart
pf=\\<central>\sapmnt\<sapsid>\sys\profile\start_<instance1>_<app>
SAPDIAHOST=<app>


goto finito


REM ERROR_HANDLING


:bad_user

      echo this batchfile (NTstartR3andDB) will not run under user
%USERNAME% >\batcherr.err

      echo please configure the Schedule Service to run under
<sapsid>adm >> \batcherr.err

:finito
```

SAP Service Manager

The *SAP Service Manager* has been replaced by the *Microsoft Management Console*. Although the *Service Manager* is still available, SAP recommends the use of the MMC as it provides many new functions and simplifies system management. For more information, see Microsoft Management Console: Windows NT [Seite 1090].

The *SAP Service Manager* (program `sservmgr`) allows you to monitor the R/3 System directly or remotely in a simple manner. If the *SAP Service Manager* is to be used remotely, you need to copy the program `sservmgr` to the workstation (operating system required: Windows NT or Windows 95) from where administration of the R/3 System is to be carried out.

You can perform the following functions with the *SAP Service Manager*:

- **Start and stop R/3 instances**

See also:

Starting an R/3 System: Windows NT [Seite 30]

Stopping an R/3 System: Windows NT [Seite 39]

- **Monitor all processes started for the R/3 System**

  The *SAP Service Manager* uses the following color scheme to display the status of the SAP standard processes (dispatcher, message server):

  – **Yellow**: A process started synchronously is running (for example, `sapcpe`) or the dispatcher started asynchronously is running, however the R/3 System is not yet ready for logon.

  – **Green**: A process started asynchronously is running. The dispatcher is an exception. It is not displayed in green until the system is ready for logon.

  – **Gray**: The process has ended.

  – **Red**: An error has occurred and the process could not be started.

  Detailed information on a/synchronous processes can be found in Windows NT Start Profile [Seite 46].

- **Display the Windows NT start profile of the instance**
- **Display the log file generated when the system was started and stopped**

  For routine management of your R/3 System, SAP recommends using the R/3 CCMS tool (*Tools* → *Administration* → *Computing Center* → *Management System*).

To start the SAP Service Manager:

The *SAP Service Manager* can be started in the following ways:

- Select *Programs → Administrative Tools (Common) → SAP → SAP Service Manager for <SAPSID>_<Instance number>* for the R/3 instance concerned.
- Execute program **`sservmgr`**.

  `sservmgr <Parameter1>` `[`**`<Parameter2>`**`]`

`Parameter1`: **`<SAPSID>_<Instance number>`** (e.g. `C11_00`)

`Parameter2`: <**`Computer name`** on which the R/3 service runs> (optional, however `Parameter2` must be specified if the *SAP Service Manager* is to run remotely).

See also:

Start and Shutdown Processing under Windows NT [Seite 43]

Registry Entries

Definition

The registry entries required for the R/3 System are made during installation. These include the environment variables, the definition of the R/3 services, and the name of the Windows NT Start Profile [Seite 46].

When started, the R/3 service reads the required environment variables from the registry *HKEY_LOCAL_MACHINE\SOFTWARE\SAP\<SAPSID>\Environment\<variable>*.

All services started under the R/3 service have the system environment at the time when booted and the environment variables found in the registry key *HKEY_LOCAL_MACHINE\SOFTWARE\SAP\<SAPSID>\Environment\<variable>*.

SAP Auxiliary Programs

SAP provides a number of auxiliary programs that can be used to compare the registry entries and the environment variables of a user if required. These auxiliary programs are located in the following executable directory after installation:
\USR\SAP\<SAPSID>\SYS\EXE\RUN.

- **NTENV2REG**

  The user environment variables are written to the registry key *HKEY_LOCAL_MACHINE\SOFTWARE\SAP\<SAPSID>\Environment\<variable>*.

  If desktop operations are performed under a specific user, the appropriate processes are started with the system environment and this user-specific environment. If processes are started as a service, these only have the system environment and the account under which the service is started. However, the user-specific environment variables are often required as well. To adapt the registry entries to these requirements, SAP provides the auxiliary program NTENV2REG.

- **NTREG2ENV**

  The SAP-specific environment variables (from the registry key *HKEY_LOCAL_MACHINE\SOFTWARE\SAP\<SAPSID>*) are transferred to the environment of the current user.

  If the R/3 System was installed under a user other than <SAPSID>ADM, this user often does not have the environment variables required to start the R/3 System. By starting the program NTREG2ENV as user <SAPSID>.ADM, the necessary environment variables are transferred from the registry to the user's environment.

  SAP recommends that you do not change the registry entries directly but only using the appropriate SAP programs.

Changes in the registry to the environment variables do not become effective until the R/3 service has been stopped and restarted. Bear in mind that when the R/3 service is stopped the R/3 System is stopped as well. However, the R/3 System is generally not started when the service is started.

- `NTREG2CMD <SAPSID>`

  The environment variables that were set in the registry are transferred to the command file `<SAPSID>ENV.CMD`. The file `<SAPSID>ENV.CMD` is required to set the necessary environment for processes started by other services (for example, `telnet`, `at` schedule service).

See also:

SAP Service Manager [Seite 34]

Example: Script for Starting the R/3 System [Seite 32]

Stopping an R/3 System: Windows NT

The following describes how to stop the R/3 System or individual instances using the *SAP Service Manager*. You can also perform these tasks using the *Microsoft Management Console (MMC)*. SAP recommends the use of the MMC as it provides many new functions and simplifies system management. For more information, see Microsoft Management Console: Windows NT [Seite 1090].

To stop an R/3 System that only consists of one central instance:

1. Log on to Windows NT as the R/3 administrator (`<SAPSID>ADM`, e.g. `C11ADM`).
2. Stop the central instance of your R/3 System:
   - Select *Programs* → *Administrative Tools (Common)* → *SAP* → *SAP Service Manager for <SAPSID>_<Instance number>* for the R/3 instance you want to stop. This takes you to the window SAP Service Manager [Seite 34].
   - Select *Stop* in the *SAP Service Manager* window to stop the instance.

     When the stoplights for all processes displayed in the *SAP Service Manager* window are "gray", the R/3 System has been stopped.

     The *SAP Service Manager* checks the activity in an R/3 instance only when the processes of the instance have ended. Make sure that there are no active users or background jobs running before you shut down the instance. To do this, use the CCMS or the tools under menu options *Tools* → *Administration* → *Control.*
3. The database is not stopped in this stop procedure. Therefore stop the database for the R/3 System after the stop procedure if necessary. To do this, use the database-specific commands or administration tools that may be available, for example SAPDBA for ORACLE/INFORMIX.

To stop an R/3 System that consists of several instances:

1. Stop all dialog instances. To do this, you can use the *SAP Service Manager* or the R/3 tool CCMS.

   To access the CCMS stop functions, select *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *System Monitor* in the R/3 main menu.

   See: Starting and Stopping Instances [Seite 69]
2. Stop the central instance. To do this, proceed as described in the section *To stop an R/3 System that only consists of one central instance*.
3. The database is not stopped in this stop procedure. Therefore stop the database for the R/3 System after the stop procedure if necessary. To do this, use the database-specific commands or administration tools that may be available, for example SAPDBA for ORACLE/INFORMIX.

See also:

SAP Service Manager [Seite 34]

Stopping an R/3 System: Windows NT

Start and Shutdown Processing under Windows NT [Seite 43]

Example: Script for Stopping the R/3 System

```
@rem echo off
REM
REM   substitute following strings:
REM      string to subst    substitution value  typical values
REM         <sapsid>       SAP R/3 System name, e.g., C11 (case
sensitive)
REM         <central>      your central host's host name
REM         <app>          your applications server's host name
REM         <instance>     the instance name, e.g., DVEBMGS00
REM         <instance1>    the app server's instance name,e.g., D00
REM         <no>           the central system's SAPSYSTEM, e.g., 00
REM         <no1>          the app server's SAPSYSTEM, e.g., 01
REM
REM  for your values see the definitions in the profiles.
REM  all profile names and values including the hostnames are
cASeSenSiTivE!!!

REM   check the users which will definitely not work
if %USERNAME% == "SYSTEM" goto bad_user

REM   first check existence of environment setup file
if not exist <sapsid>env.cmd
\\<central>\sapmnt\<sapsid>\sys\exe\run\ntreg2cmd <sapsid>

REM setup the environment for the process so that he is able to
REM run also from poor telnet daemons and Schedule Service
call <sapsid>env.cmd

REM you may want also to shutdown the application server here...
REM if so: repeat the next two lines per app-server to stop them
REM sapsrvkill <app>_<sapsid>_<no1>
REM sapntwaitforhalt
pf=\\<central>\sapmnt\<sapsid>\sys\profile\START_<instance1>_<app>
SAPDIAHOST=<app> 120
```

Example: Script for Stopping the R/3 System

```
REM shutdown central instance here...

sapsrvkill <central>_<sapsid>_<no>

sapntwaitforhalt
pf=\\<central>\sapmnt\<sapsid>\sys\profile\START_<instance>_<central>
SAPDIAHOST=<central> 120


REM you may also want to stop additional application server services
here...

REM if so use: ntscmgr stop SAP<sapsid>_<no1> -m <app>


REM stop central instance service here...

ntscmgr stop SAP<sapsid>_<no> -m <central>


REM DATABASE_SHUTDOWN....

REM for example    sapdba -shutdown

sapdba -shutdown


REM SHUTDOWN COMPLETE


goto finito


REM ERROR_HANDLING


:bad_user

      echo this batchfile (NTstopR3andDB) will not run under user
%USERNAME% >\batcherr.err

      echo please configure the Schedule Service to run under <sid>adm
>> \batcherr.err

:finito
```

Start and Shutdown Processing under Windows NT

Start Processing

1. Start the R/3 services.

   When the Windows NT system is booted, the R/3 service (`SAP<SAPSID>_<Instance number>`, e.g. `SAPC11_00`) is always started automatically. The service evaluates entries from the Windows NT Start Profile [Seite 46] and reads the necessary environment variables from the registry.

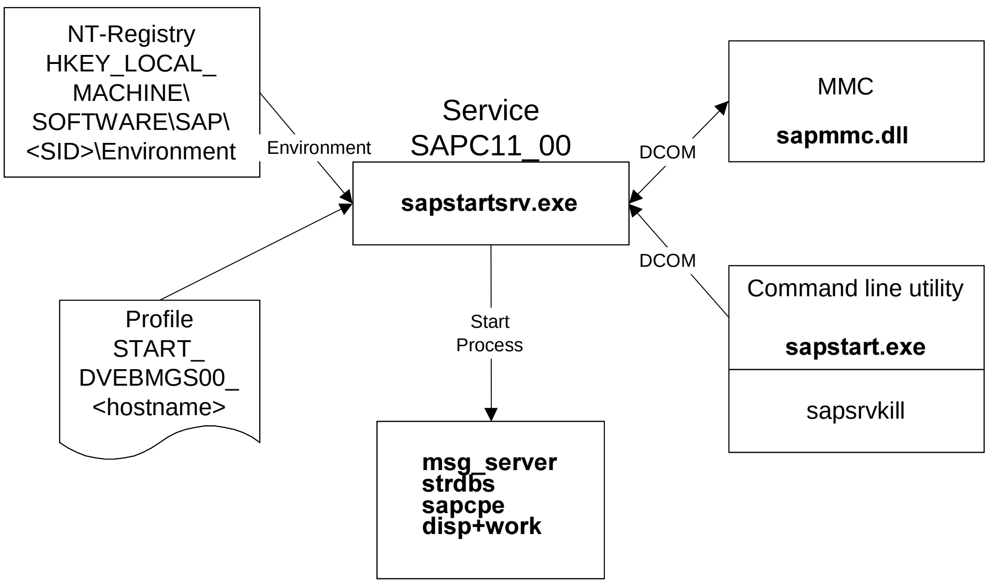
2. Start the R/3 processes.

   You can start the R/3 processes using the Microsoft Management Console (MMC). To do this, select an instance and choose *Start* from the context menu. For more information, see Microsoft Management Console: Windows NT [Seite 1090].

   DCOM communication is set up with the R/3 service and messages for starting the appropriate R/3 instance are sent to the service. The R/3 service interprets the Windows NT start profile (see the section *Evaluating the start profile* below) and starts the processes the instance consists of (the processes are started by the program `sapstartsrv.exe`).

   The message for starting the appropriate instance can also be issued by the command line program `sapstart.exe`.

   The following diagram illustrates the interaction of the individual processes in the start procedure:



See also:

Registry Entries [Seite 36]

Windows NT Start Profile [Seite 46]

SAP Service Manager [Seite 34]

Windows NT: R/3 Program Overview [Seite 49]

Evaluating the Start Profile

When you start an instance, the R/3 service executes all commands in the Windows NT start profile that have an `Execute_` statement.

The R/3 service then starts the processes of the R/3 instance in the order in which the `Start_Program_` statements are listed in the profile.

It is possible to set the parameter `autostart = 1` in the Windows NT start profile. If this parameter is set, the R/3 service **and** the R/3 System are started when the Windows NT system is booted. Bear in mind, however, that the individual processes are not synchronized automatically. This means, for example, that the administrator must ensure that the application servers are not started until the database server has been started successfully.

Shutdown Processing

The R/3 system can be stopped as follows:

- **Stop using the MMC or other auxiliary programs**

  The R/3 service waits for a stop message from the MMC or a command line program (for example, `sapsrvkill`). It then stops the R/3 System as described in the section *Evaluating the Start Profile*. The service itself is not stopped.

- **Stop the R/3 service via the Windows NT window *Services***

  From the list of services, select the service of the R/3 instance (`SAP<SAPSID>_<Instance number>`, for example `SAPC11_00`) you want to stop. The R/3 service then stops the R/3 System as described in the section *Evaluating the start profile*. The service itself is then stopped.

  The database is not stopped in this shutdown procedure.

Evaluating the Start Profile

The R/3 service executes stop commands for the R/3 processes it has started itself. Only those processes started with `Start_Program_` statements are stopped. Processes started with `Execute_` statements are not taken into account.

The R/3 service logs the start and shutdown of the R/3 System. You can display this log, for example, from the *SAP Service Manager*. For further information on error situations, see R/3 System on Windows NT: Problem Analysis [Seite 48].

The R/3 System can also be started and stopped using appropriate scripts.

The processes under which these scripts are started (for example, `at` schedule service) may **not** be started under the user *ADMINISTRATOR* or with the *SYSTEM Account*. In the following you will find examples of these scripts. The variables (name of the R/3 System, names of the instances…) still need to be adapted to the specific R/3 System:

Example: Script for Starting the R/3 System [Seite 32]

Example: Script for Stopping the R/3 System [Seite 41]

Example: Script Using DCOM Interface

See also:

Starting an R/3 System: Windows NT [Seite 30]

Stopping an R/3 System: Windows NT [Seite 39]

R/3 System on Windows NT: Problem Analysis [Seite 48]

Windows NT Start Profile

The start profile determines which actions the R/3 service executes. The profile can start all processes which are required by the R/3 System in this Windows NT host system. The start profile used by the service is specified when you install the service.

Start profiles are stored in the central profile directory of your R/3 System so that they can be called from any application server.

A start profile is generated automatically during system installation.

A distinction is made between processes started synchronously and asynchronously.

- Processes started synchronously

  The next step in the start profile is not processed until the process has been completed. The system does not check how (that is with which `Exit Code`) the process was completed.

  Synchronous processes can be recognized by the addition `immediate` in the start profile.

  Program `strdbs.cmd` - starts the database

  Program `sapcpe` - copies the necessary executables to the application server.

- Processes started asynchronously

  The next step in the start profile is processed as soon as the process has been started.

  Program `msg_server.exe` - starts the message server

  Program `disp+work.exe` - starts the dispatcher

  Program `gwrd.exe` - starts the gateway (provided this is just a gateway instance)

Changing or Replacing the Windows NT Start Profile

You can change entries in the start profile using the CCMS, for example. Changes to the start profile do not become effective until you have stopped and restarted the R/3 service. Note that when the R/3 service is stopped, the appropriate R/3 System is also stopped but not the database.

In the *Microsoft Management Console*, you can activate the new profile settings by restarting the R/3 start service without stopping the R/3 System. For more information see Microsoft Management Console: Windows NT [Seite 1090].

To stop an R/3 service:

- Start the Windows NT window *Services* (*Control Panel* → *Services*).
- From the list of services, select the service of the R/3 instance (`SAP<SAPSID>_<Instance number>`, e.g. `SAPC11_00`) you want to stop. The service has already been stopped if it does not have the *Status Started*.

If you want to start an R/3 instance under another start profile, you must de-install the R/3 service and reinstall it.

See also:

Start and Shutdown Processing under Windows NT [Seite 43]

R/3 System on Windows NT: Problem Analysis

If problems occur when the R/3 System is started or stopped, you have a number of analysis options.

- Select *Programs → Administrative Tools (Common) → Event Viewer* to access the Windows NT event log. In the menu bar, select *Log → Application*. You are shown a list of errors, warnings, and information generated by the application software. You can display detailed information by clicking on a particular log.
- Check that the R/3 service (`SAP<SAPSID>_<Instance number>`, e.g. `SAPC11_00`) was started.

  To do this, go to the window *Services* (*Control Panel → Services*) and check the status of the R/3 service. The service has been started if it has the *Status Started*. If this is not the case, select *Start* to start the service.
- Check the start log in the MMC [Seite 1090]. To do this, select the instance involved and from the context menu choose *Task → View Developer Traces*.
- Check the error files **`stderr<n>`** (n is the number of one of the programs in the Windows NT start profile: `Start_Program_<0n>...`). You can find the files in `USR\SAP\<SAPSID>\<Instance number>\work`.

  The files can also be accessed from the MMC. To do this, select the instance involved and from the context menu choose *Task → View Developer Traces*.
- Check the trace files of the individual R/3 processes:
  - `dev_ms`: Developer trace of message server
  - `dev_disp`: Developer trace of dispatcher
  - `dev_w<m>` (`m` number of work process): Developer trace of work processes

Windows NT: R/3 Program Overview

Program	Use
sapcpe	Replication program called when the application server is started to copy the executables to the directory <DRIVE>:\USR\SAP\<SAPSID>\INSTANCE\EXE
strdbs	Starts the database
sapmmc.dll	Microsoft Management Console (MMC)
sservmgr	SAP Service Manager
sapsrvkill	Auxiliary program for stopping the R/3 System
sapntwaitforhalt	Auxiliary program for synchronizing the individual start/stop processes
msg_server	Starts the message server
disp+work	Starts the dispatcher and work processes
gwrd	Starts the gateway
sapstartsrv.exe	SAP R/3 service
sapstart	Program for sending start messages to the R/3 service
ntscmgr	Generates/deletes R/3 service definitions, starts/stops services
ntenv3reg	User environment variables are written to the appropriate registry key
ntreg2env	SAP-specific environment variables are transferred from the registry to the environment of the current user
ntreg2cmd	Environment variables set in the registry are transferred to the command file <SAPSID>ENV.CMD

The program **sapstartsrv.exe** is release-dependent. It can only be used in the release for which it is delivered.

See also:

[Example: Script for Starting the R/3 System [Seite 32]]

[Example: Script for Stopping the R/3 System [Seite 41]]

Starting and Stopping the Database System: UNIX

Use

You can use the following commands under UNIX to start and to stop the R/3 central instance and the database system.

Procedure

To...	Use the command or function...
Start the database system	`startsap db`
Stop the database system If necessary, check that the network service of the database system is also stopped.	`stopsap db`
Start the R/3 System The database system is also started if it is not already running.	`startsap` or `startsap all`
Stop the R/3 System The database system is also stopped if it is still running.	`stopsap` or `stopsap all`
Start one R/3 instance only	`startsap r3`
Stop one R/3 instance only	`stopsap r3`

Commands under UNIX to start or stop the database system

You enter the commands `startsap`, `stopsap`, `sapstart` and `sapstop` in the command line of the host operating system.

Once the startup procedure has run, you can use the CCMS to start or stop all other application instances.

See also:

Controlling System Startup [Seite 61]

Starting and Stopping the R/3 System: Overview [Seite 29]

Manual: *System Administration Guide*

Starting an R/3 System: UNIX

Procedure

To start an R/3 System:

1. Log onto UNIX as the SAP administrative user (`<SAP System name>adm`).
2. Enter the following command from your home directory:

   `startsap [DB|R3|ALL]`

   where:

   – `DB` starts the database system

   – `R3` starts the instances and associated processes of the R/3 System (using the `sapstart` program)

   – `ALL` starts both the database system and the R/3 System. ALL is the default and can be omitted.

To start the R/3 System, the `startsap` script calls the `sapstart` process with the start-up profile specified in the `START_FILES` variable in the script.

Once the R/3 System is running, you can perform all other actions from within the CCMS.

To check that all of the application servers of an R/3 System have started correctly:

Choose *CCMS, Control/Monitoring* → *System Monitor.*

Stopping an R/3 System: UNIX

Procedure

To stop an R/3 System:

1. Log onto UNIX as the SAP administrative user (`<SAP System name>adm`).
2. Enter the following command from your home directory:

   `stopsap [DB|R3|ALL];`

   where:

   – `DB` stops the database system

   – `R3` stops the instances and associated processes of the R/3 System

   – `ALL` stops both the database system and the R/3 System. All is the default and can be omitted.

To stop the R/3 System, the `startsap` script calls `kill.sap`. `kill.sap`, in turn, activates shutdown processing in `sapstart`.

Program SAPSTART

Definition

The startsap and stopsap scripts start the R/3 System with the sapstart program.

Syntax

The syntax of the sapstart program is as follows:

```
<SAP-executables>/sapstart pf=<start profile>;
```

where

<SAP-executables> is the path name of the directory which contains the executable files of the R/3 System.

<profile name> is the name of the start profile with which you wish to start the system.

```
/usr/sap/C11/SYS/exe/run/sapstart
pf=/usr/sap/C11/SYS/profile/START_DVEBMGS00
```

You can use the sapstart command in a batch program under Windows NT to start the R/3 System automatically. You can also start an R/3 System on a remote host using the sapstart command and the pre-defined variable SAPDIAHOST=host.

Start Profiles

Your choice of profile determines which actions sapstart carries out. For example, if you are starting the entire R/3 System, you could select a profile that starts all of the processes of the system on all of the host systems in which the processes run. Or, if you have just restarted a host system, you could select a profile that starts only the processes that execute in that host.

A start-up profile is an ASCII file and can be entered or edited with an ASCII editor. Start profiles are stored in the shared central profiles directory of your R/3 System (for example /usr/sap/C11/SYS/profile). They can therefore be accessed from any application server.

A start profile is generated automatically when you install your system.

Start Processing

When you start it, the sapstart program executes any preliminary commands that you may have entered in the profile with Execute_ statements.

It then starts R/3 System processes in the order in which the Start_Program_ statements in the profile are numbered. sapstart starts processes asynchronously. It does not attempt to determine the status of one process before starting the next one. It does not issue a start command for a process that is already running.

All processes of a running R/3 System are daughters of the `sapstart` process.

Shutdown Processing

After it executes all of the commands in a start profile, `sapstart` waits until either of the following occurs:

- all of its daughter processes terminate
- it receives a stop signal.

  `stopsap` works by sending a stop message to `sapstart` with `kill.sap`.

In the latter case, `sapstart` issues commands for graceful shutdowns to the SAP processes it started. Only processes started with `Start_Program_` statements are shut down. Processes started with `Execute_` statements are not affected.

`sapstart` issues the orders asynchronously in the order in which the processes were started. SAP processes are therefore shut down in parallel.

In both shutdown modes, `sapstart` then performs any post-processing that you have requested. It issues the commands in any `Stop_Program` statements in the start profile in the order of the numbers in the statements. `sapstart` issues post-processing commands synchronously (one after another) and shuts down as soon as the last command has been issued.

`sapstart` logs the start-up and shutdown.

Starting the R/3 System: AS/400

To start an R/3 System:

1. Log on to the AS/400 system using the SAP administrative user profile. This user requires the rights provided by group profile `<SID>OPRGRP` (for example, user profile `<SID>OPR`).
2. Issue the AS/400 **`STARTSAP`** command and press *`F4=Prompt`* to prompt for parameters.
3. For SAP system ID, enter the name of the R/3 System (for example, “C11”).
4. For R/3 instance, enter the instance number (for example, “90”). To start all instances on a host (or hosts), select **`*ALL`**.
5. For R/3 instance hostname, enter the name of the host on which the instance is to be started. To start all instances on all hosts, select **`*ALL`**. (You must have also selected **`*ALL`** for R/3 instance.)
6. Press `ENTER` to confirm your entries. For each instance started, a subsystem R3_nn will be started (where “nn” is the instance number), along with all the associated R/3 services (such as dispatcher, work processes, spool).
7. Use AS/400 command **`WRKACTJOB SBS(R3_nn)`** to check that your R/3 System or Systems started successfully.
   In the next screen, look at the entry in the first line of the *Status* column on the right of the screen. If an R/3 System started correctly, this field should contain the value `RUN` (running), `SELW` (select wait) or `DEQW` (waiting for the completion of a dequeue operation). It may take several minutes for the file system to reach this state.

Stopping the R/3 System: AS/400

To stop an R/3 System:

1. Log on to the AS/400 system using the SAP administrative user profile. This user requires the rights provided by group profile <SID>OPRGRP.
2. Enter the AS/400 command **STOPSAP** and press *F4=Prompt* to prompt for parameters.
3. For *SAP system ID*, enter the name of the R/3 System to be stopped.
4. For *R/3 instance*, enter the number of the instance to be stopped, for example “90”. If you want to stop all instances on a host (or hosts), select ***ALL**.
5. For *R/3 instance hostname*, enter ***LOCAL** to stop an instance or instances on the local host. To stop all instances on all hosts, select ***ALL**. (You must also have selected ***ALL** for *R/3 instance*.)
6. If you enter ***YES** for *Wait for instance to end*, the STOPSAP command waits until the R/3 instance has gone down before stopping the R/3 System. (The instance is considered to be down when the number of active instance user jobs in the instance subsystem, not including the SAPOSCOL job, is zero.)
7. For *Maximum wait time (seconds)*, you can specify the maximum time the command is to wait for the instance to come down. The default value is 120 (two minutes). If the instance takes longer than two minutes to come down, an exception message is sent.
8. Press ENTER to confirm your selections.

Controlling System Startup

Procedure

Before implementing the R/3 System, you need to decide how you want to control system startup.

You have two alternatives:

- Use operation modes [Seite 63] to start the R/3 System
- Use startup profiles [Seite 67] to start the R/3 System

See also:

Operation Modes [Seite 192]

Instances [Seite 229]

Starting the R/3 System Using Operation Modes

Use

You can use operation modes to start your R/3 System.

Using operation modes to start/stop the remaining instances has the following advantages:

- Simplifies the maintenance of the startup profiles. Since only one control instance has to be started "manually", you do not need to create or maintain an extensive startup profile to start up all the instances.
- Enables you to manage the different configurations of your system within the CCMS.
- Enables you to reconfigure your system automatically at set intervals to adapt it to different situations.
- Enables you to explicitly reconfigure your system by selecting an operation mode and executing an operation mode switch.
- Enables you to manage and monitor your system more easily. The instances and services required at any one time are fixed in advance. Deviations are easy to recognize.

Prerequisites

To utilize this instance startup technique, the central R/3 instance connected to the database must be running.

Procedure

To start the R/3 System:

1. Log onto the control instance.
2. In the CCMS, select an operation mode and start the remaining instances.

See also:

Platform Specificity [Seite 65]

Platform-Specificity

Definition

Platform-specific restrictions currently apply only to the functions for starting servers remotely from the CCMS.

- To start an R/3 instance at a remote UNIX system, the CCMS uses the UNIX **rexec** command. In order to use **rexec**, you must enable a remote logon to the target UNIX system. See Prerequisites for Starting Remote Instances [Seite 236].
- R/3 instances on remote Windows NT servers are started by way of a message to the service manager on the remote Windows NT system. If the R/3 System has been correctly installed, then no additional maintenance should be required to enable remote start-ups and shutdowns.
- R/3 instances on remote AS/400 are started using AS/400 mechanisms. If the R/3 System has been correctly installed, then no additional maintenance should be required to enable remote start-ups and shutdowns.

Starting the R/3 System Using Start Profiles

Use

You can also start profiles to start your R/3 System.

Procedure

To start the R/3 System, maintain an extensive start profile for every system configuration you wish to use. You can then use this start profile to start all the system instances.

Alternatively, proceed as follows:

1. Log onto each of the host systems in which SAP instances are running.
2. Enter the `startsap` command with a suitable start profile.

    Your R/3 System must be set up for interactive work during daytime. The system must be optimized for your end users. At night and at weekends you want to set up the system to process background jobs and for maintenance purposes.

    To facilitate reconfiguration, you simply define the required operation modes. When an operation mode switch occurs, the system can reconfigure itself in line with the selected operation mode.

Starting and Stopping Instances

Use

You can perform these tasks from the control panel [Seite 956].

Procedure

To start or stop instances from the control panel:

1. Select the instances [Seite 229] you want to start or stop.

   You can select specific instances, or all instances.

2. Choose *Control* → *Start SAP instance* **or** *Stop SAP instance*.

   The selected instances are then started or stopped in the current operation mode.

3. Ensure that all instances are started or stopped correctly.

   You can see the status of the instances from their color in the system monitor:

   green successful startup

   red startup error

   gray an instance has been stopped (or is inactive)

4. Check whether start and stop commands were carried out correctly

   To do this, choose *Refresh* from the control panel. Active instances are indicated by a cross in the *Active* column.

To start the inactive instances of an operation mode:

1. Choose *Select* → *All stopped.*
2. To start the instances, choose *Control* → *SAP instances*

To stop the instances for the selected operation mode:

1. Choose *Select* → *All configured*.
2. To stop the instances, choose *Control* → *SAP instances*

See also:

Starting and Stopping the SAP System [Seite 29]

Starting and Stopping the Database System: UNIX [Seite 50]

Operation Modes [Seite 192]

Platform-Specificity [Seite 65]

Errors During Startup

Use

If an instance crashes or cannot be started up, it is highlighted in red in the system monitor. In the control panel, the *Services* and *Status* columns remain empty or the entries are deleted.

Procedure

To analyze the error:

1. Select the instance.
2. Display the start log file (choose *Logs/Traces/Info* or *Trace files*).

   Error messages and a return code that indicate why the system could not be started can be found in the start log and in the dispatcher and work process trace files.

Return codes that can appear in the start log

Return Code	Meaning
`0`	OK. Instance is started.
`1`	Start terminated, instance is already running.
`2`	Child process of the SAPSTART program completed.
`3`	Child process of the SAPSTART program is no longer available.
`-101`	Unknown SAP system name.
`-102`	*Change Directory* command unsuccessful.
`-103`	Access to an operating system resource denied (*access permissions* set incorrectly).
`-104`	No start profile found.
`-105`	Executing a command in the start profile on remote computer failed.
`-106`	Executing a command in the start profile on local computer failed
`-107`	Error in signal handling.
`-108`	Internal control error in the SAPSTART program. Control files with process ID numbers are missing.
`-109`	Kernel is not complete. Executable files not copied in full to the target host.
`-110`	Internal error in the SAPSTART program.

You can display trace files and log files, including the start log, at any time, regardless of whether the instance is running.

See also:

Checking the Operation Mode with the Control Panel [Seite 225]

Background Processing

Purpose

This component enables you to process background jobs using the Computing Center Management System (CCMS). With CCMS you get extensive support in scheduling and managing background jobs.

Integration

The CCMS is a standard part of the SAP System.

Features

You can use background processing in CCMS to:

- Configure and monitor the background processing system
- Schedule and manage background jobs

Background Processing: Concepts and Features

For a very high-level introduction to background processing and job scheduling, see Getting Started with the R/3 System [Extern].

Purpose

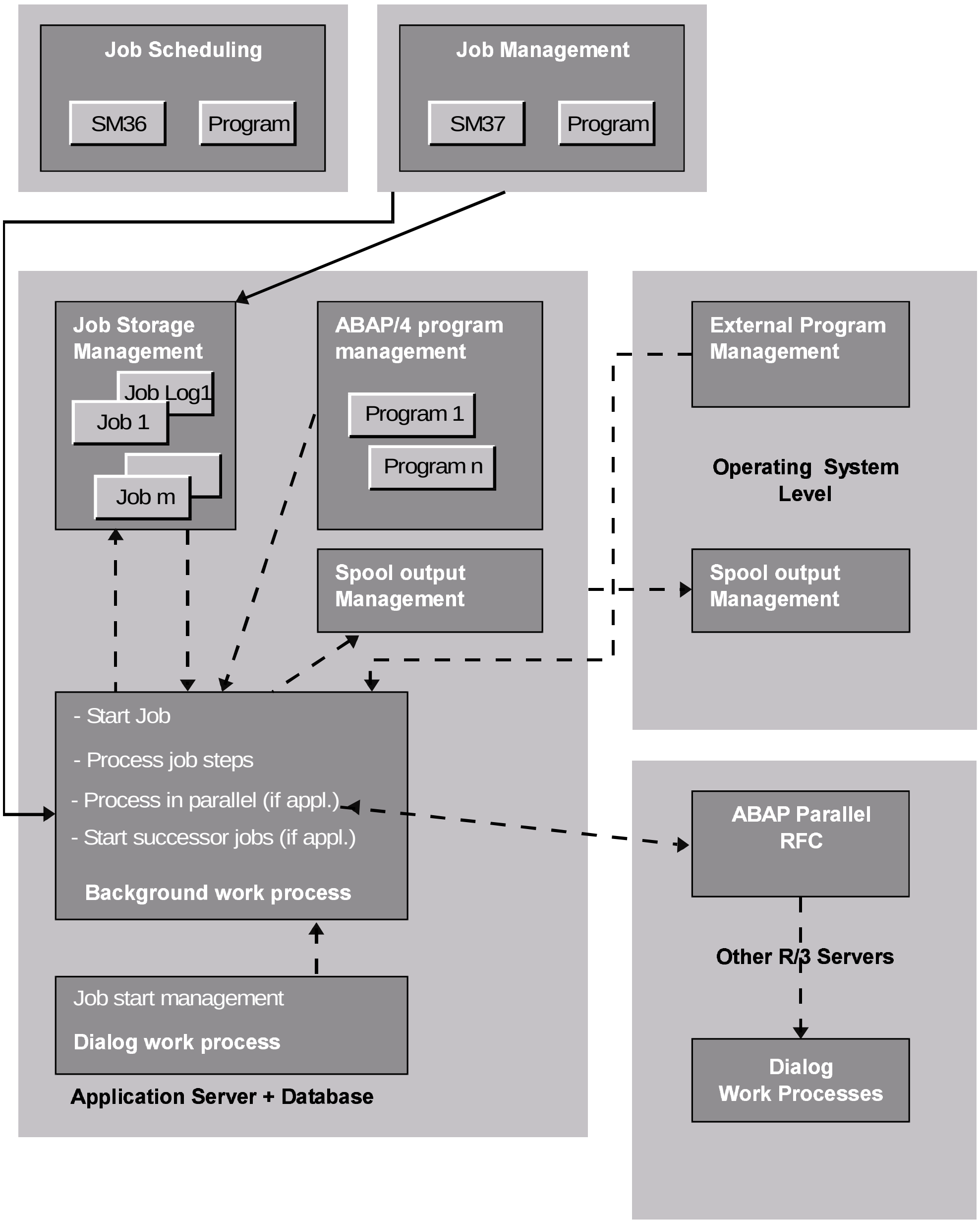
SAP background processing automates routine tasks and helps you optimize your organization's SAP computing resources. Using background processing, you tell the SAP System to run programs for you. Background processing lets you move long-running or resource-intensive program runs to times when the system load is low. It also lets you delegate to the system the task of running reports or programs. Your dialog sessions are not tied up, and reports that run in the background are not subject to the dialog-step run-time limit that applies to interactive sessions. For more information, see Background Work Processes Explained [Seite 78].

The SAP System offers sophisticated support for background processing. You can choose from a variety of methods for scheduling [Seite 106] and managing jobs. You can run both SAP-internal and external programs [Seite 93]. And, for easier scheduling and management, you can run related programs as "job steps [Seite 90]" within a single background processing job, allowing a single background job to accomplish a complex task that consists of multiple processing steps.

The system includes sophisticated tools for managing jobs and diagnosing problems [Seite 137] that occur, including a graphic monitor [Seite 135] and a powerful and easy-to-use job programming interface [Extern] for developing your own background-processing applications. There is also a job scheduling wizard [Seite 109] that automates basic background job definition and can walk novice users through the entire process.

Finally, the background processing system has an interface to external management tools so you can integrate your SAP background processing into an external tool. Certified implementations of this interface are available for several external system management tools.

The following diagram shows the components that are involved in processing various types of background jobs.



Monitoring Background Processing

You can track background processing activity in your SAP System with both list-oriented and graphical monitors.

The list-oriented monitor (Transaction SM37) offers an administrator full control over background processing. From the main monitor screen, you can display current status and job details, change or withdraw scheduling and releases, display job logs, and use debugging and error-analysis tools. With appropriate authorization, you can also display the spool requests generated by ABAP job steps. For more information, see Managing Jobs with the Job Overview [Seite 129].

The graphical monitor (Transaction RZ01) is specialized for system operators. It provides a Gantt-chart view of background processing, showing all completed, active, and released jobs on a timeline across all available background servers and work processes and tracking previous job performance so that repeated jobs are always shown with Gantt chart lengths that reflect previous run times. For more information, see Using the Graphical Job Scheduling Monitor [Seite 135].

Setting Up the Background Processing System

Purpose

To ensure that your background processing system is set up correctly, complete the following tasks.

Process Flow

1. When you first set up the SAP System, you should decide on a strategy for giving important background jobs priority in processing. For more information, see Prioritizing Class A Jobs [Seite 116].
2. When you set up a new background server, make sure that background processing is activated and that the server is integrated into your operation modes. For more information, see Activating Background Processing [Seite 107].
3. You will need to assign the correct user authorizations when you set up either of the two following users:
   - A new background processing administrator
   - A background-processing end user with more capabilities than simply scheduling jobs and checking job status

   For more information, see Authorizations for Background Processing [Seite 87].
4. You have the option to create RFC groups for specifying which servers may be used for processing particular parallel backgrounds. For more information, see Defining RFC Groups for Parallel-Processed Jobs [Seite 127].

Background Work Processes Explained

Definition

Background jobs run in a special type of work process—the background work process—that is different from dialog work processes in two ways:

1. A dialog step has a run-time limit that prevents users from interactively running especially long reports. By default, the system terminates any dialog step in a transaction that exceeds 300 seconds. Although the limit can be changed (in the system profile parameter `rdisp/max_wprun_time`), it is always in effect for dialog work processes. No such limit applies to background work processes.
2. Background work processes allocate memory differently than dialog work processes so that background work processes can become as large as they need to in allocated memory to allow for processing large volumes of data.

Having special work processes for background processing provides an additional dimension for separating background processing and interactive work. Typically, background processing and interactive work on the system is separated by time of day. For example, interactive users have the system during the day, large background jobs have the system by night. But the background work process also lets you separate interactive from background use by server, since background jobs can only run on those servers that offer background work processes.

Use

Using background work processes also lets you optimize both background or dialog processing. In “night” mode, more servers can offer more work processes to accommodate background processing jobs that are scheduled to run when there are lower numbers of interactive users. In “day” mode, you could limit background processing to fewer servers and/or fewer background work processes.

When a time-driven scheduler is activated, it starts the jobs for which it has free work processes. If there is another background processing server, then the scheduler on that server starts the remaining jobs for which it has capacity, and so on.

Although a job can specify to use a particular background server (an application server that has at least one background work process), it is best to allow the background processing system to use load balancing to distribute the workload among the available servers.

Parallel processing, or asynchronous RFC (remote function call), is the only exception to the rule that job processing only occurs in background jobs. A report that has been programmed for parallel processing runs in a background work process, but it can also dispatch work to available dialog processes. For more information, see Parallel Processing of Jobs with Asynchronous RFC [Seite 125].

To monitor background work processes, use the graphical job scheduling monitor [Seite 135] or the server and work process overviews.

ABAP Programs

To run an ABAP program in the background, you must provide a variant for reports with selection screens or the ABAP program must supply its own runtime parameters. You can specify the R/3 user under whose authorizations the program should run, as well as how the spool system should handle output from the program.

Output generated by an ABAP program is held in the SAP spool system as a spool request. The job step printing and archiving specifications specify how to handle this output. Spool requests can also be automatically mailed via SAPoffice to any user designated in the job definition, but because output can be rather large, this option should be used with care.

External Commands and External Programs

To run an external, non-SAP program, you must identify the host to run the program on and the path to the program as well as any arguments the program needs. The background processing system runs the external program by starting the SAP server program SAPXPG on the target host system and then uses RFCs to communicate with SAPXPG.

You can also specify how to manage the execution of the external program. For example, in synchronous execution, the background job waits until SAPXPG returns with the final status of the external program, while in asynchronous execution, the job proceeds immediately to the next job step once it has started SAPXPG. Synchronous execution also allows you to retrieve error output and a return code from the external program.

The background processing system makes a distinction between external commands for normal users and external programs for system administrators. You can see this distinction in the job scheduling transaction [Seite 106], with its separate fields for external commands and external programs.

External commands are pre-defined commands for end users and can run on any operating system. They are protected by system administrator-defined authorizations, so end users are restricted to scheduling only the commands they've been authorized to run.

External programs are unrestricted commands that can be entered into a job step by any user with administrator authorizations. Administrator authorization allows a user to run any program. For more information, see External Commands and External Programs [Seite 93].

Required Basis Background Jobs

Definition

There are several background jobs [Seite 90] that should run periodically in a production SAP System. These jobs perform housekeeping chores such as deleting outdated spool requests of background jobs.

You should schedule each of these jobs for periodic execution according to the schedule shown in the following table.

This list does not include housekeeping and reorganization programs that belong to SAP applications. See the customizing system for more information on these programs.

By following the naming conventions shown in the table, you help ensure that SAP will be able to verify that these programs are properly scheduled in the event of any problems.

Periodic Jobs Required for Basis Housekeeping:

Job Name	ABAP Program	Required Variant	How Often?
SAP_REORG_JOBS Deletes old background jobs.	RSBTCDEL	You must create a variant.	Daily
SAP_REORG_SPOOL Deletes old spool requests.	RSPO0041	You must create a variant.	Daily
SAP_REORG_BATCHINPUT Deletes old batch input sessions. This job may not run at the same time as normal batch input activity. Schedule this job for periods during which no batch input sessions are run.	RSBDCREO	You must create a variant.	Daily
SAP_REORG_ABAPDUMPS Deletes old dumps produced by ABAP abnormal terminations. **Alternative:** To keep from needing to schedule this job, run the ABAP report RSNAPJOB from the ABAP editor instead. This schedules RSSNAPDL as follows: Job name: RSSNAPDL Variant name: DEFAULT (you must create this variant) Start time: 0100 AM Repeat interval: Daily	RSSNAPDL	You must create a variant.	Daily

SAP_REORG_JOBSTATISTIC Deletes job statistics for jobs not run since the specified date (statistics no longer needed since job was a one-time occurrence or is no longer run)	RSBPSTDE	You must create a variant.	Monthly
SAP_REORG_UPDATERECORDS Deletes old completed update records (automatic delete deactivated); deletes incomplete update records (automatic delete deactivated) Run this job ONLY if: • You have deactivated the default automatic deletion of update records once they have been processed. This function is controlled by the system profile parameter rdisp/vb_delete_after_execution • You have deactivated the default automatic deletion of incomplete update records (records that are partially created when an update header is created and saved but the generating transaction then ends abnormally). This function is controlled by system profile parameter rdisp/vbreorg • You have deactivated processing of V2 update components after the processing of the associated V1 updates. This function is controlled by system profile parameter rdisp/vb_v2_start.	RSM13002	None.	Daily
SAP_COLLECTOR_FOR_JOBSTATISTIC Generates runtime statistics for background jobs	RSBPCOLL	None.	Daily
SAP_COLLECTOR_FOR_PERFMONITOR Collects system performance statistics This job was previously called COLLECTOR_FOR_PERFORMANCE_MONITOR. When scheduling this job, be sure to use the new name. RSCOLL00 schedules all reports that need to run for the performance monitor using table TCOLL to determine what to run. See the CCMS Guide [Seite 24] for more information on setting up RSCOLL00.	RSCOLL00	None.	Hourly

Standard Jobs

Standard jobs are those background jobs that should be run regularly in a production SAP System These jobs are usually jobs that clean up parts of the system, such as by deleting old spool requests.

Use

As of Release 4.6C, the *Job Definition* transaction (`sm36`) provides a list of important standard jobs, which you can schedule, monitor, and edit.

Features

SAP delivers the following standard jobs including recommended variants and job intervals:

Job name	**Report name**	**Info**
SAP_CCMS_MONI_BATCH_DP	RSAL_BATCH_TOOL_DISPATCHING	Dispatches monitoring architecture methods
SAP_COLLECTOR_FOR_JOBSTATISTIC	RSBPCOLL	Generates runtime statistics for background jobs
SAP_COLLECTOR_FOR_PERFMONITOR	RSCOLL00	Collects data for the performance monitor
SAP_REORG_ABAPDUMPS	RSSNAPDL	Deletes old ABAP short dumps
SAP_REORG_BATCHINPUT	RSBDCREO	Deletes old batch input sessions
SAP_REORG_JOBS	RSBTCDEL	Deletes old background jobs
SAP_REORG_JOBSTATISTIC	RSBPSTDE	Deletes old data from the job runtime statistics
SAP_REORG_SPOOL	RSPO0041	Deletes old spool data
SAP_REORG_UPDATERECORDS	RSM13002	Deletes old update requests
SAP_REORG_XMILOG	RSXMILOGREORG	Deletes XMI logs

Activities

You can use the functions to:

- Monitor the Status of Scheduled Standard Jobs [Seite 84]
- Schedule Standard Jobs [Seite 85]
- Define New Standard Jobs [Seite 86]

For more information about standard jobs, see SAP Note 16083.

Overview of Standard Jobs

Definition

The status overview of standard jobs displays all jobs with status *Scheduled, Finished, Canceled, Ready,* or *Active*.

For canceled and finished jobs, only the latest of each is displayed.

Scheduling Standard Jobs

Prerequisites

Before a standard job can run, you must schedule the job.

Procedure

If you want to schedule all the standard jobs, choose *Default scheduling*. The system schedules all standard jobs that are defined in table REORGJOBS along with their specified variants and intervals.

If other jobs exist that execute the programs contained in the standard jobs, the system displays a message.

To schedule the individual jobs, choose a job using the *Possible entries* help and enter the job interval. Save your entries.

Result

The saved job along with its status appear in the overview of standard jobs.

Defining New Standard Jobs

Use

To define a new standard job that is not contained in table REORGJOBS, proceed as follows:

Procedure

1. Choose *Predefine new job*.
2. Enter the required data.
3. Save the definition.

Result

You can now schedule [Seite 85] the new standard job.

Authorizations for Background Processing

Definition

Authorizations for accessing background processing jobs can be set up for two types of users: administrators and end users.

Give administrator authorization only to a small number of trustworthy employees. This level of authorization allows a user to:

- Have unrestricted access to background processing jobs
- Perform any operation on any job, including:
  - Selecting jobs from all clients (from the *Job Overview*, Transaction SM37)
  - Deleting any job
  - Releasing jobs to start
  - Changing jobs, including copying, canceling, checking, repeating, and capturing and debugging.
  - Displaying jobs, job steps, and job logs
  - Triggering events manually (Transaction SM64)
  - Editing system events
  - Using restricted job classes A or B
  - Scheduling an external program in a job step. (External programs are directly entered external commands for which no predefinition in the SAP system is required and for which no authorization test is run.)
  - Maintaining SAP definitions for external commands

All users can schedule, cancel, delete, and check the status of their own jobs with no additional special authorizations. By default, a job runs under the authorizations of the user who scheduled it. Special user-level authorizations are needed for:

- Any operation on a job scheduled by another user
- Displaying the job log
- Displaying a spool request generated by a background job
- Releasing a job to start
- Using an external command

A user's jobs are defined and run in the user's current logon client, regardless of whether the user's background processing authorizations are set for user or for administrator.

Setting Up Authorizations

Administrator authorization setup requires the following authorization objects:

Authorization Object	Value
S_BTCH_ADM (*Batch Processing: Batch Administrator*) Allows all of the activities listed above except for maintaining external command definitions. No default profile with ONLY this authorization is currently shipped with SAP, but the standard SAP_ALL profile contains this authorization.	Y
S_RZL_ADM (*CC Control Center: System Administration*) Allows an administrator to maintain external command definitions and to trigger commands from the external command function (Transactions SM49 and SM69).	01
S_BTCH_JOB (*Batch Processing: Operations on Batch Jobs*) Allows an administrator to view job-generated spool requests. To protect sensitive data, this authorization is not included in the standard administrator authorization.	LIST
S_DEVELOP (*ABAP Workbench*) Allows an administrator to capture and debug background jobs by providing access to ABAP debugging tools	

User authorization setup beyond job scheduling and status checking requires the following authorization objects:

Authorization Object	Value(s)
S_BTCH_JOB (*Batch Processing: Operations on Batch Jobs*) Allows all of the activities listed above except for maintaining external command definitions. No default profile with ONLY this authorization is currently shipped with SAP, but the standard SAP_ALL profile contains this authorization.	DELE (delete other users' jobs) LIST (display spool requests) PLAN (copy other users' jobs) PROT (display anyone's job log) SHOW (display job details) RELE (release other users' jobs to start; a user's own jobs are automatically released when scheduled.)
S_BTCH_NAM (*Batch Processing: Batch User*) Allows a user to specify other users for runtime authorization for a job.	permissible users

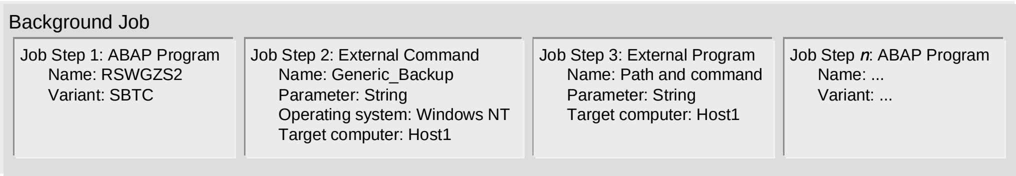
S_LOG_COM (*Authorization to Execute Logical Operating System Commands*) Allows a user to run external commands.	
S_ADMI_FCD (*System Authorizations*) For special functions, such as debugging active jobs.	

For complete information, see authorization object documentation from Transaction SU21.

Jobs and Job Steps Explained

Definition

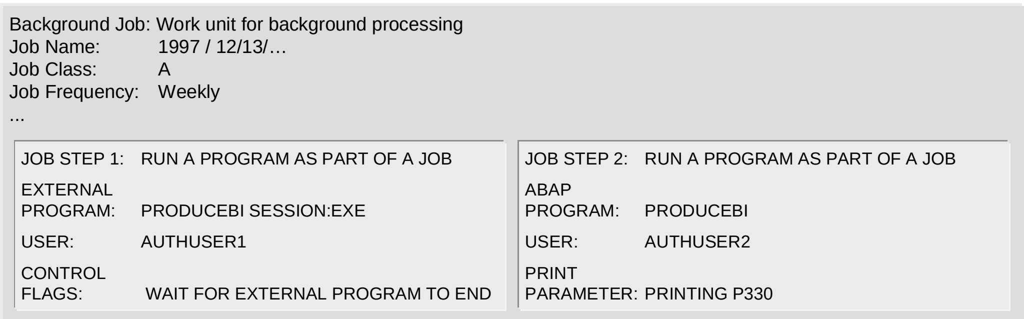
The work unit of the background processing system is the *background job,* each of which consists of one or more *job steps*.



Jobs and job steps enable you to treat complex tasks as single units. That is, you can schedule several programs needed to complete a particular task as steps within a single job, with the advantage of the job being single logical container for all the steps needed to complete the task. You need to schedule or review only one background job in order to schedule or review any of the individual steps necessary for completing the task.

Assume that a particular data transfer with batch input requires that you start two programs, an external program to prepare the batch-input session and an internal program to process the session. Creating a job made up of two steps lets you handle the two programs as a single unit. Scheduling that one job schedules both programs. The results of each program's run can be seen in the job log.

Some background processing attributes apply to entire jobs and, therefore, to all job steps within a job. For example, the earliest possible start time for any job steps will be the start time for the job. Frequency of repetition, priority, and other global attributes also apply to the whole job.



To ensure that you can flexibly run individual programs, you can set important attributes individually for each job step, too. Each job step can:

- have its own spool, or output, specifications
- run under the authorizations of a separate user
- use a different language
- have its own runtime options (for programs external to the SAP System), such as handling of error output and synchronicity

In general, job steps run sequentially and synchronously in the order they're entered in a job: the first step starts, runs, and is completed, then the second step starts, and so on. The only exception is when you schedule an external program to run asynchronously. In this case, the background processing system starts the next job step without waiting for a return code from the external program. If the external program runs long enough, then the start of the next job step may overlap it.

Job steps run *partially* independently of each other's status. That is, the abnormal termination of one job step does not roll back the work of a previously completed job step if this previous step was executing a commit. If any job step fails, however, then the entire job fails. No further job steps are carried out, and the job's status changes to *Canceled*.

There are two types of job steps:

- **An executable ABAP program**

  Only type 1, or executable, ABAP programs can be used as job steps. Module pools and function groups, which are collections of ABAP modules, are not allowed. The specifications required for an ABAP job step are:

  ABAP program **+** *Variant* **+** *Print and archiving parameters* **+** *Language*

- **An external command or external program**

  This type of job step allows you to run programs outside the SAP System. *External commands* are predefined, authorization-protected commands for end users. *External programs* are unrestricted, directly entered commands reserved for system administrators.

  The type of external command and external program is unrestricted, meaning that you can use either compiled programs or scripts. Such programs can be run on any computer that can be reached from the SAP System. Parameter passing to non-SAP programs is completely unrestricted except by the predefinition mechanism for external commands.

  Output of non-SAP programs, particularly error messages, is included in the job's log file.

  Specifications required for an external command or program are:

  – *External command* **+** *Type of operating system* **+** *(Parameters)* **+** *Target host system*

  – *External program* **+** *Parameters* **+** *Target host system*

Job Step Language

If your SAP System runs with an Arabic, Cyrillic, or Asian character set, then you may need to change the language specified in the standard background jobs listed above. Languages are assigned to jobs based on job steps. Be sure that the language specified in the job step definition in each of these jobs is set to English (EN).

This change is required because not all languages are available with all code pages. If the job's specified language is not available, the job will not run.

See also:

External Commands and External Programs [Seite 93]

Required Basis Background Jobs [Seite 80]

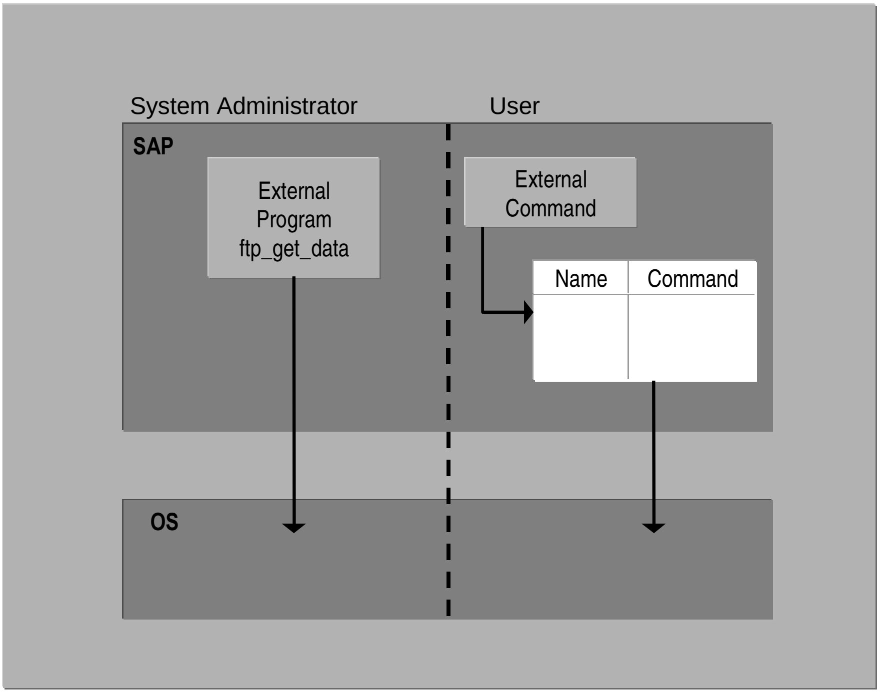
Job Scheduling Explained [Seite 105]

Managing Jobs from the Job Overview [Seite 129]

External Commands and External Programs

Definition

The background processing system makes a distinction between *external commands* for normal users and *external programs* for system administrators. You can see this distinction when scheduling a job from Transaction SM36, with separate fields for external commands and external programs.



External commands

External commands are predefined commands for end users. They are operating-system independent and are protected by authorizations, so that normal end users can schedule only those commands that the system administrator permits them to.

With an external command, an ordinary end user—any user without background processing administrator authorization—may run a host system command or program that has been pre-defined by the administrator in the SAP System. The user who schedules the external command must have the authorization required for the external command.

External commands let you control what your users do outside the SAP System. End users can run only the commands and arguments that you specify in external command definitions. And you can control access to external commands with SAP authorizations.

For additional security, external command definitions are operating-system specific. For example, you can define variants of a command for UNIX and Windows NT hosts. A user who schedules an external command must specify the type of operating system in which the command is to run. The system then automatically selects the correct operating system variant or issues an error if the required variant has not been defined.

External programs

External programs are unrestricted commands that are neither pre-defined or restricted by authorizations. A user with administrator authorization can enter any of these in a job step.

With an external program, a system administrator can enter any desired host operating system command or program in a job step. No SAP authorizations test is carried out before executing the command.

External programs give an administrator—a user with the background processing administration authorization (authorization object S_BTCH_ADM *Batch processing: Batch Administrator*)—the flexibility to run any required host system command without any administrative preparation in the SAP System.

The purpose of this distinction is to let system administrators execute any required *external program* while restricting normal users to authorizations-tested *external commands*.

See also:

Prerequisites for Running External Commands and External Programs [Seite 95]

Analyzing Problems With External Commands and Programs [Seite 154].

Prerequisites for Running External Programs

The prerequisites for starting external commands and external programs [Seite 100] from background jobs depend on:

- Whether the remote shell functions (`remsh` or `rsh`) are available in the operating system. (They are not available in standard Windows NT systems)
- Whether an application server from the same SAP System is running on the target host system

If there is an SAP server on the target host for an external program, then the SAP gateway on that server is called and starts SAPXPG, the external program controller. If there is no server on the target host, then the gateway at the SAP server running the background job uses a remote shell facility to start SAPXPG on the target host.

UNIX and Other Systems with a Remote Shell

Systems with a remote shell can run an external command or program [Seite 93] without an active SAP application server. With no application server present, the gateway uses the remote shell and RFC to start the SAPXPG program. SAPXPG, in turn, runs the external command or program.

The prerequisites for running an external command or program depend on whether the target host has an SAP application server or not.

If an application server from the same SAP System is running on the target host, then the following requirements must be met:

- The user with which the SAP gateway was started (usually the standard SAP host system user *<sid>adm*, such as *c11adm*) must have permission to run the external program.
- The command or program must either:
  - Lie in the standard search path of the target system
  - Lie in the search path of the SAP standard user
  - Be entered with an absolute path name in the job step or external command definition.

If no application server is present, then the following apply:

- If you do not have the remote shell `remsh` on your systems, then you must set the profile parameter *gw/remsh* in the instance profile of the application server on which the job is running to the name of your remote shell program, for example, '`rsh`'.

  On a UNIX system you can use the command **`which rsh`** or **`which remsh`** to find out which remote shell program is available. You may need to specify the complete path name of your remote shell program in the profile parameter *gw/remsh*.

  - The user with which the gateway was started (usually the standard SAP *<sid>adm* user, such as *c11adm*) must exist on the remote host
  - The SAP control program *sapxpg* must be found in the search path of the standard SAP user on the remote host. SAPXPG is stored in the SAP executables directory.

    If the external program was not specified with an absolute path name, then the external program must also be found in the search path of the standard SAP user. In

UNIX you can check this in the standard user by calling the following command: `which <program name>` while logged on as the standard user.

– Both the SAP control program *sapxpg* and the external program to be started must be executable under the standard SAP user. For example, on a UNIX machine this means that <*sid*>*adm* must have execute permission ('x') for the programs.

– The host on which the job is running must have permission to start programs on the remote host. There has to be an appropriate entry in the .rhosts file in the home directory of the gateway ID on the remote host. This consists of a line including the name of the host on which the job is running.

  For more information, read the documentation, such as the UNIX man pages, about the remote shell (`remsh`).

  You can test this by entering the command '*remsh <Name of remote host> date*' on the host on which the job is running. On a UNIX platform, the output should then be the system date on the remote host.

There is one special case: Assume that you have set up the environment on a UNIX system correctly for starting external programs by way of an SAP application server. However, the SAP system is started using an operating system specific mechanism.

You must, therefore, ensure that the user's path names are set up correctly for the current system environment. For example, if the Bourne shell is being used, the paths are entered into the *.profile* file. The use of the C-Shell (csh) is assumed.

For example, the UNIX command *'cron'* allows you to execute a command at a defined time, including the commands to start or stop an SAP system. This uses the Bourne shell (program *'sh'*) for executing a command. The user-specific settings which define the environment in which the command is to be executed—that is, the path names mentioned above—are then stored in the *.profile* file in the user's home directory.

Microsoft Windows NT

To start an external command or program on a Windows NT system, the following requirements must be met:

- An SAP application server from the same SAP System must be running on the host on which the external command or program is to run.

  In systems without `remsh`, like Windows NT, the SAP gateway must use an SAP application server to start the SAPXPG program. (SAPXPG, in turn, runs the external command or program.)

- The external command is run by the user that started the SAP application server on the target host. This is usually the standard SAP host system user <*sid*>*adm*, such as *c11adm*. This user must have permission to run the external program.

- The command or program must either:

  – Lie in the standard search path of the target system

  – Lie in the search path of the SAP standard user

– Be entered with an absolute path name in the job step or external command definition.

OS/400

To start an external command or program on an IBM OS/400 system, the following requirements must be met:

- An SAP gateway service must be running on the AS/400 on which the external program is to run, which will be the case wherever an SAP application server is running on AS/400.
- The external program is run under the user profile that is used to start the SAP standalone gateway/server, which is to say the OS/400 user for SAP. This user must be authorized for the SAPXPG command and the external program.
- SAPXPG must be found in the library list of the OS/400 user for SAP.
- If the path to the external program is not absolutely specified, then the command must be found in the library list of the OS/400 user for SAP.

Defining External Commands

Use external commands to allow your background processing users to schedule external executables (commands, scripts) in job steps. For more information on external commands and external programs, see External Commands and External Programs [Seite 93].

Prerequisites

You must first be authorized to define external commands and your end users must also be authorized to schedule the external commands that you define. See Authorizations for Background Processing [Seite 87]. For external commands to be carried out, the host system on which a command is to run must also meet certain requirements. See Prerequisites for Running External Commands and External Programs [Seite 95].

Procedure

1. Choose *CCMS* → *Configuration* → *External Commands* (Transaction SM69). This displays a list of the commands that are already defined.
2. Choose *Command list* → *Display* <-> *Change* to enter the change mode.
3. Choose either the *Create* or *Change* function.
4. To create a command, fill out the fields in the definition screen as follows:
   – *Command name*:
     Assign a name for your command. If you define versions of a command for more than one operating system, use the same name for each so the system can automatically select the correct version for the target system.

   – *Operating system*:
     If you need to define different versions of a command for different operating systems, select the appropriate operating system for the version you are defining. If a single version of the command will work on any operating system, enter **ANYOS**.

   – *Type*:
     This field automatically shows whether the command was defined by you or by SAP.

     Changes that you make directly to SAP commands may be overwritten. To ensure that such changes will be preserved, copy the SAP command and make your changes to your copy. This copy will have the type *Customer*, which means it will not overwritten by SAP upgrades and releases.

   – *Operating system command*:
     Enter the full path and name of the command unless the command resides in the path normally checked by SAPXPG, the program that starts external commands. Enter no other arguments for the command here; those will be entered in the *Parameters* field. This command can be any type of executable: program, script, .bat file, etc.

   – *Parameters for operating system command*:
     Enter the arguments or parameters for the command.

     If no argument is needed when the external command is scheduled in a job, leave this field blank.

- *Additional parameters allowed*:
  If users can specify additional arguments when they schedule an external command in a background job, mark this field. These additional parameters are added to any parameter string specified in the previous field, *Parameters for operating system command*.

- *Trace*:
  Leave this field blank. To trace the execution of an external command, use the Trace parameter for function module `SXPG_COMMAND_EXECUTE` in Programming with External Commands [Extern].

- *Check module*:
  To define an extra authorization check for any external command, you must program it in a copy of a special function module.

  If you have defined an extra authorization check, enter the name of the function module that makes the check here. The system calls the function module automatically if a user tries to schedule this external command in a background job. For more information on programming an extra authorization check, see SXPG DUMMY COMMAND CHECK: Interface for Extra-Check Function Modules. [Extern]

5. Save the external command and exit.

6. Be sure to give your users appropriate authorization to schedule the external command in background jobs. See Authorizations for Background Processing [Extern].

Starting External Commands and Programs

This section describes special procedures for starting service programs or daemons that are to remain active in the target system and multiple cascading programs.

Starting Service Programs and Daemons

Once you've used the background processing system to start service programs, such as daemons in UNIX systems, they remain active rather than terminating and returning to the SAP background control program as other programs do.

When you start a service program, you should use the control flag setting *Job to wait for external program to end* (found in Transaction SM36, the *control flags* function in the Job Step Definition) when you schedule the job. Since the external program isn't supposed to immediately terminate, you should also change the default *Wait* setting. The SAP control program will then terminate as soon as it has started the external program. You can continue to get trace data back from the control program up until the time that it starts the external program.

Starting Cascading Programs

Cascading programs run one or more other external programs. However, you may not be able to obtain useful feedback from the initial external program. If the external program terminates without waiting for the other programs, the return code in the job log provides no useful information on the additional processing done in the target system. You may be able to remedy this in a couple of ways:

- If you can modify the external program, have it issue an event [Seite 102] to the background processing system rather than starting the additional programs. You can then schedule individual jobs [Seite 106] for each of the "cascaded" programs that are to be started. When the initial program issues the event, the background processing system starts each of the jobs that is waiting for the event. The advantage to this is that you can receive return codes, output, and/or trace data from each of the external programs in the job logs.
- If you cannot modify the external program, then modify the "cascaded" programs to trigger events to signal their outcomes to the SAP System. You can use the event to have the background processing system start additional processing.

  For example, during data transfer into your SAP System, you could start each external data transfer program from a script or small control program that waits for the transfer program to terminate.

Upon termination of the transfer program, the control program checks the return code and issues the appropriate event to the SAP System with the SAP program SAPEVT. You can have the background processing system use the event to start the ABAP program that processes the transferred data.

See also:

External Commands and External Programs [Seite 93]

Prerequisites for Running External Programs [Seite 95]

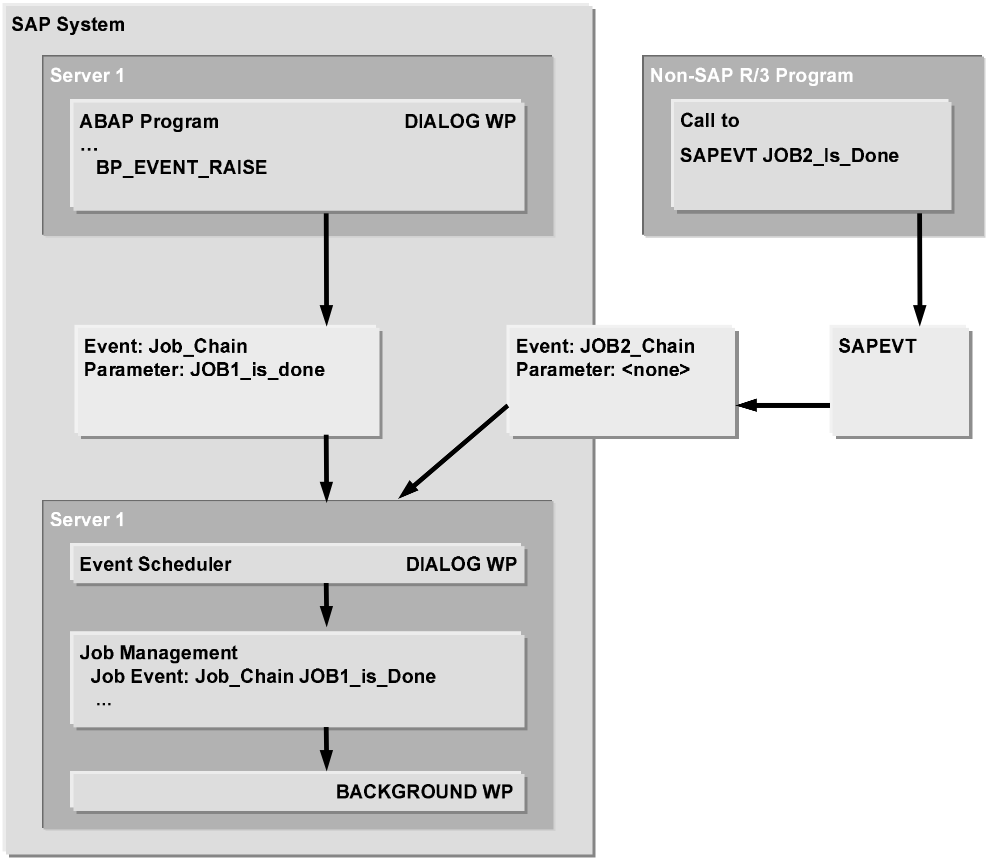
Defining External Commands [Seite 98]

Events in Background Processing Explained [Seite 102]

Events in Background Processing Explained

Definition

In addition to time- and calendar-based job scheduling, the background processing system supports event-driven scheduling. The image below illustrates how background processing events can be used and processed.



Use

Events have meaning only in the background processing system and can be used only to start background jobs.

Triggering an event notifies the background processing system that a particular condition has been met. The background processing system reacts by starting any jobs that were waiting for that event.

Events are not saved. Once an event occurs and triggers any jobs that were waiting for that event, the event is discarded.

Integration

Event Types

There are two types of events:

- **System events:** Defined by SAP and triggered automatically by system changes, such as the activation of a new operation mode. You cannot modify system events.
- **User events:** Defined by users. These events must be triggered from ABAP or from external programs. For example, use an external program to trigger a background processing user event to signal the arrival of external data to be read into the SAP System.

Event Arguments

An event argument is an optional text string you can use to qualify an event. You can specify an event argument when:

- Scheduling a job to wait for the event
- Triggering an event

Unlike event IDs, event arguments are not defined in the SAP System.

If you specify an argument when you schedule a job, then the job is eligible to start only when the event is triggered with the argument.

If you don't specify an argument when you schedule a job, then the job can start as soon as the event occurs. The job is eligible to start no matter what argument string is supplied with the event.

When a Job Waiting for an Event May Start

Job	Event	Result
Job scheduled with Event ID "JSTART" Argument "A"	JSTART triggered, no argument	Job does not start, continues to wait for JSTART A
	JSTART triggered with argument "A"	Job starts
	JSTART triggered with argument "B"	Job does not start; continues to wait for JSTART.
Job scheduled with Event ID "JSTART", no argument	JSTART triggered, no argument	Job starts
	JSTART triggered with any argument	Job starts

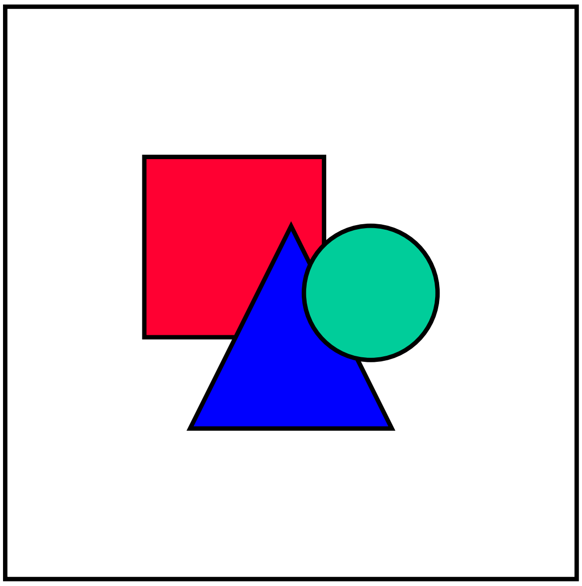
An ABAP program that is running as a background job can find out what event and argument were presented when it was started. This allows ABAP programs running in the background to react intelligently to an event depending on the argument string that was supplied with it.

An Example of Events and Event Arguments

Switching to a different operation mode triggers a system event in the background processing system. The event is an SAP System event named SAP_OPMODE_SWITCH. As an argument, the event carries the name of the new operating mode.

If you schedule a job to wait on the event SAP_OPMODE_SWITCH with argument NIGHT, then your job is eligible to start when the operating mode NIGHT becomes active the next time.

You can also schedule jobs to be repeated each time event occurs. In the previous example, your job could be run each time the operating mode NIGHT becomes active, not just the first time the event occurs.



The *Operation mode* button allows you to schedule a job to wait for a particular operation mode to become active.

Job Scheduling Explained

Definition

Before any background processing can actually begin, background jobs must be defined and scheduled. When a job is scheduled to run is one part of the job's definition. There are several ways to schedule jobs:

- From Transaction SM36 (Define Background Job)
- With the "start program in the background" option of either Transaction SA38 (ABAP: Execute Program) or Transaction SE38 (the ABAP editor)
- Through the background processing system's own programming interface. (Many SAP applications use the internal programming interface to schedule long-running reports for background processing.)
- Through an external interface.

See also:

Scheduling Background Jobs [Seite 106]

Specifying Job Start Conditions [Seite 110]

Job Start Management [Seite 111]

Periodicity: Specifying Automatic Job Repetition [Seite 114]

Managing Jobs from the Job Overview [Seite 129]

Jobs and Job Steps Explained [Seite 90]

Scheduling Background Jobs

Use

You can schedule background jobs in two ways from the *Job Overview*:

- Directly from Transaction SM36. Best for users already familiar with background job scheduling.
- The Job Wizard. Best for users unfamiliar with SAP background job scheduling. From Transaction SM36, either select *Goto* → *Wizard version* or simply use the Job Wizard button.

Procedure

1. Call Transaction SM36 or choose *CCMS* → *Jobs* → *Definition*.
2. Assign a job name. Decide on a name for the job you are defining and enter it in the *Job Name* field.
3. Set the job's priority, or "Job Class":
   - High priority: Class A
   - Medium priority: Class B
   - Low priority: Class C
4. In the *Target server* field, indicate whether to use system load balancing.
   - For the system to use system load balancing to automatically select the most efficient application server to use at the moment, leave this field empty.
   - To use a particular application server to run the job, enter a specific target server.
5. If spool requests generated by this job are to be sent to someone as email, specify the email address. Choose the *Spool list recipient* button.
6. Define when the job is to start by choosing *Start Condition* and completing the appropriate selections. If the job is to repeat, or be periodic, check the box at the bottom of this screen.
7. Define the job's steps by choosing *Step*, then specify the ABAP program, external command, or external program to be used for each step.
8. Save the fully defined job to submit it to the background processing system.

   Note: Release the job so that it can run. No job, even those scheduled for immediate processing, can run without first being released.

   For a simple job scheduling procedure, see the R/3 Getting Started Guide [Extern].

Activating Background Processing

Use

By default, background processing is inactive in the SAP System. To enable background processing, you must set up at least one CCMS operation mode to contain one or more background processing servers, SAP application servers with a minimum of two dialog work processes and one background processing work process. You need to perform these procedures only once for each background processing server that you set up.

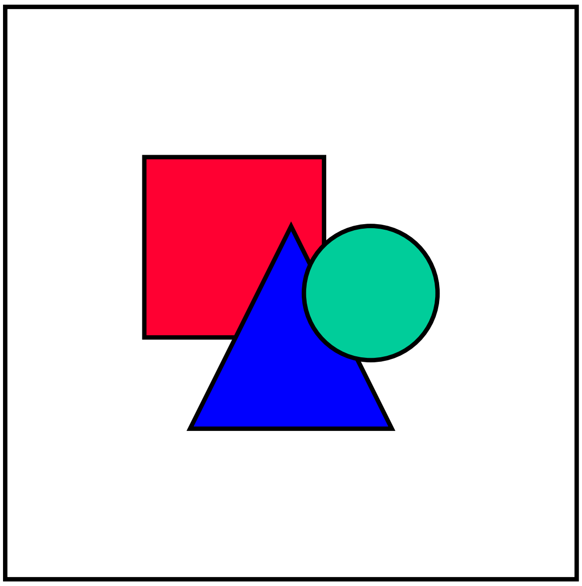
Procedures

Define a background processing server by defining an SAP instance (an application server) with at least two dialog work processes and at least one background work process:

1. Call Transaction RZ04 or choose *CCMS* → *Configuration* → *Operation Modes / Instances.*
2. To create a new operation mode, choose *Operation mode* → *Create.* Name and define the new operation mode and save the definition.
3. To change an existing operation mode, select the operation mode you want to change and choose *Operation mode* → *Change.* Rename or redefine the operation mode and save the new definition.
4. To review the current configuration of SAP instances and operation modes and begin setting up background processing servers, choose *Instances/operation modes*.
5. If the server you want to set up as a background processing server doesn't appear in this list, create a new instance. Choose *Instance* → *Create new instance.* Complete the instance definition in this *Maintain Instance Data* window, then save.
6. Once the server you want to set up as a background processing server appears in this list, select the mode for that server then click "Choose" (F2). In the *Maintain Work Process Distribution* screen that appears, set the number of work process by selecting the current number and increasing or decreasing it with the "-" or "+" buttons. Save here, return to the previous screen, and save again.

Alternative procedure:

1. Call Transaction RZ10 or choose *CCMS* → *Configuration* → *Profile Maintenance.*
2. Select the profile you want to edit, then choose *Extended Maintenance* and click *Change*. You should be able to identify the correct profiles by the host name of the server. Instance profiles follow the naming convention <SAP SYSTEM NAME>_<INSTANCE>_<HOSTNAME>, as in C40_D00_HOST1.
3. Set the parameter value of *rdisp/wp_no_btc* to the number of background work processes you want in the instance profile of each background server.
4. Restart the servers so that the profile parameter change becomes effective.
5. Assign the background processing server(s) to at least one CCMS operation mode.



Background processing jobs can be processed only in operation modes in which at least one background processing server is active. If you are using more than one operation mode, you should assign your background processing server to all of your operation modes, so that background processing is always available.

Verify that your operation modes are active from *CCMS* → *Configuration* → *Operation Mode Calendar* then *Display*.

Job Scheduling Wizard

Definition

The job scheduling wizard is an automated, step-by-step guide through the job scheduling process. It prompts you for information as you go, so there's no need to memorize the process or to spend unnecessary time looking through menus or screens trying to figure out how to schedule jobs.

Use

Start the job scheduling wizard from Transaction SM36 by choosing Job Wizard. Then simply follow the directions and prompts provided.

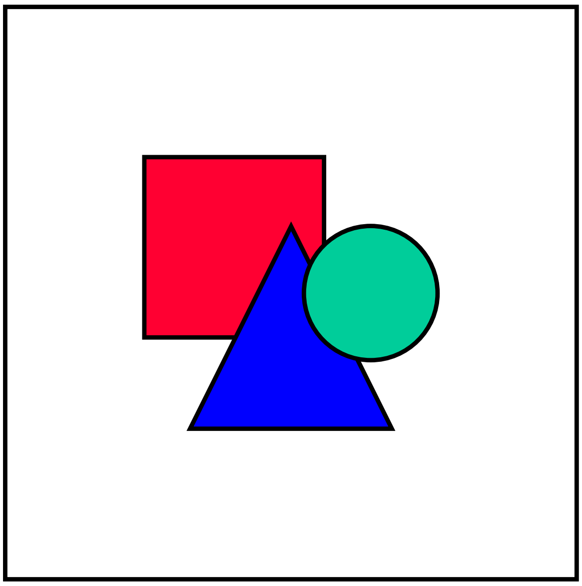
Specifying Job Start Conditions

Use

When scheduling a job from either Define Background Job (Transaction SM36; *CCMS* → *Jobs* → *Definition*), you must specify conditions that will trigger the job to start [Seite 111].

Procedure

1. Choose the *Start condition* button at the top of the *Define Background Job* screen.
2. Choose the button at the top of the *Start Time* screen for the type of start condition you want to use (*Immediate*, *Date/Time*, *After job*, *After event*, or *At operation mode*) and complete the start time definition in the screen that appears.
3. For the job to repeat, check the *Periodic job* box at the bottom of the *Start Time* screen and choose the *Period values* button below it to define the frequency of repetition (hourly, daily, weekly, monthly, or another specific time-related period). Then choose the "Save" button in the *Period values* screen to accept the periodicity and return to the *Start Time* screen. .
4. Once you've completed specifying the job start conditions, choose the "Save" button at the bottom of the *Start Time* screen to return to the *Define Background Job* screen.



No job can be started until it is released, including jobs scheduled to start immediately. Since releasing jobs [Seite 120] can be done only by a system administrator from the job management screen (Transaction SM37) or by other users who have been granted the appropriate Authorizations for Background Processing [Seite 87], no unauthorized user can start a job without explicit permission.

Job Start Management

Definition

Any job can be scheduled to start based either on a certain start time or on any of several events in the background processing system.

Use

Both types of start condition (see below) has its own SAP job scheduler: one is time-driven and one is event-driven. Both schedulers run in dialog work processes. If they find an eligible job, they trigger the job to run in a background work process [Seite 78].

Time-Driven Job Scheduler

This scheduler runs periodically to check for jobs that were scheduled to run based on a certain start time (a particular date, for example). By default, the scheduler runs every 60 seconds on each SAP server that has at least one work process available for background processing. This 60-second interval can be changed in the system profile parameter `rdisp/btctime`.

The time-driven scheduler will also take over the responsibility of starting event-based jobs if such jobs cannot yet be started when the defined start condition occurs (if, for example, no free background work processes are available when the start condition is met). These jobs are managed as jobs whose start time has already arrived and start running as soon as possible.

Time-Driven Job Scheduler

This scheduler starts whenever a background processing event [Seite 102] is triggered. Background processing events have nothing to do with workflow events and have no system-wide meaning. An example of a background processing event is “Job completed.”

The scheduler checks for any jobs that have been scheduled to wait for a certain event or for the event argument and starts these jobs once the event occurs. Event-driven jobs are those that have been scheduled with the one of three start conditions: *After event, At operation mode* or *After job*.

Events can either be triggered within the SAP System or generated externally at the operating-system level. Events are automatically passed by the message server to an active background processing server. In order to respond to external events, the SAP System must be active. An event triggered by an external program will be lost if the SAP System is not running.

Start Eligibility

A job is eligible to start when both of the following are true:

- The start condition specified for the job is met.
- The job has been released to run [Seite 128].

  No job can be run until it has been released, even those scheduled to start immediately. To monitor and control what jobs are submitted to run in background processing, the system can be configured so an administrator can check jobs before releasing them to run.

  The release requirement can also be turned off on a per-user basis. Trusted users can be given a special authorization (authorization object S_BTCH_JOB (*Batch Processing:*

Operations on Batch Jobs), value RELE) which will automatically and immediately release any job scheduled by that user.

Job Start Conditions Explained

Job Start Conditions	Explanation
Immediate	This job will start as soon as possible, subject to job priority and the availability of background work processes. Users granted the appropriate Authorizations for Background Processing [Seite 87] can have their jobs released automatically as soon as they are scheduled, without special attention from a system administrator.
Date/time	The job will start at the date and time specified, subject to job priority and the availability of background work processes. You can further define a timeframe in which the job will run by specifying a time later than which a job will not run. One use of this feature is to prevent periodic jobs from running when they should not. For example, a routine clean-up job scheduled to run only at night is delayed. By defining a start-time window, you will keep this job from being started during the day, when dialog users are active and fewer system resources are available.
After Job	The job will start on the completion of the specified job. You can specify whether the preceding job must complete successfully or not. **Note:** Jobs scheduled to start when a preceding job completes cannot be specified as “Periodic jobs”.
After Event	The job is triggered by the specified event, descriptions of which follow. This start condition lets you define a sequence of individual jobs to model complex activities in your system. For example, a data transfer program that generates a batch-input session can, once the session has been completely generated, use an event to trigger another job that processes the session.
At operation mode	The job will start when the specified operation mode becomes active in the SAP System.

Workday/Time (>>)	The job will start on the specified day of the month (for example, the third to last day of the month) By specifying the appropriate SAP factory calendar, your scheduling can accommodate the occurrence of holidays or other non-workdays. You can further specify how the job should be handled if the start day lands on a non-working day. For example, you can have the job started on the workday before or after the actual scheduled date, or allow the job run in anyway, or have the job be skipped on that day entirely. Verify that the calendar you choose has been correctly defined. If, for example, the calendar ends with the current year and your job is to be started next year, the job will terminate abnormally. If a calendar problem occurs, you can review system's error messages in the job log of the terminated job.

Periodicity: Specifying Automatic Job Repetition

Use

To run your job not only at a particular time, but also to repeat it at regular intervals, you can specify a periodicity, or frequency of repetition. This can be specified in Define Background Job in Transaction SM36 (C*CMS* → *Jobs* → *Definition*).

1. Start the process of specifying when the job will start by choosing the *Start condition* button.
2. Choose the button at the top of the *Start Time* screen for the type of start condition you want to use (*Immediate*, *Date/Time*, *After job*, *After event*, or *At operation mode*) and complete the start time definition.
3. For the job to repeat, check the *Periodic job* box at the bottom of the *Start Time* screen and choose the *Period values* button below it to define the frequency of repetition (hourly, daily, weekly, monthly, or another specific time-related period).
4. Choose the "Save" button in the *Period values* to accept the periodicity and return to the *Start Time* screen.

When a periodic job's start time is *After event*, the system repeats the job every time the relevant event is triggered.

A periodic-job series is not interrupted if one of the jobs terminates abnormally. The background processing alert monitor warns you if any jobs abort, but if the job was not executed at all, there will be no following job. For more information, see Displaying Job Alerts [Seite 133].

Start-Time Windows and Job Repetition

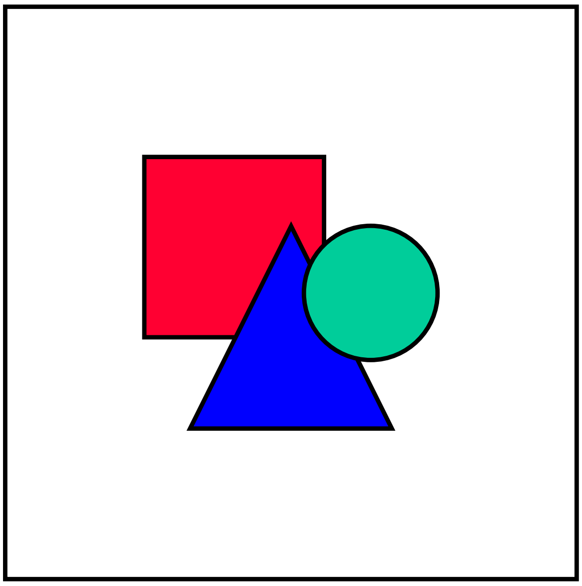
If you schedule a job with a start-time window (that is, a range of time rather than a particular time) and want to have the job repeated, the background processing system adjusts the start-time window so that its length remains constant and is relative to the respective start time. If you specify a start-time window of six hours, then each repetition of the job will also have a six-hour start-time window, starting when the job repetition is scheduled.

You cannot use the start-time window to limit the time of day during which a job is repeated. Scheduling a repeatable job to run between 10:00pm and 6:00am does not guarantee that the job will be repeated only within this time period. Rather, the full start-time window will be applied to all of the repetitions of the job according to the rules above.

For example, you schedule a job to run between 10:00 this evening and 6:00 tomorrow morning and to be repeated every hour. The job will be triggered to repeat only after it has started the first time. At 10:00, the job starts. At the same time, the system schedules it again for 11:00. The start-time window for the job has been adjusted to 11:00 and 7:00.

As the system schedules each repeated job, it adjusts the start-time window to remain the length you specified. The repeat job for 1:00am, for example, will have a start time window of 1:00 to 9:00.

To limit the job to the time period between 10:00 and 6:00, you could define a job that schedules separate jobs to run each hour from 10:00 to 6:00. None of the jobs should be periodic.



To limit an automatically repeated job to a particular time window, you'll need to write a program yourself to schedule the jobs. Your program could schedule a separate job for each repetition that falls within the time period. For more information, please see the R/3 manual Basis Programming Interfaces [Extern].

Prioritizing Class A Jobs

Use

You can reserve any number of background processing work processes to be kept free for high-priority, or class A, jobs. Reserving work processes for class A jobs does not reserve any *particular* work process in a server for class A jobs, but rather makes sure a certain number of work processes is free at all times. To set the number of class A background work processes, you'll need to set the work process distribution [Seite 207] and assign each work process to an operation mode.

Procedure

To set the number of work processes to keep free for high-priority jobs in the definition of an instance (SAP application server):

1. Call Transaction RZ04 or choose *CCMS* → *Configuration* → *Operation Modes / Instances*.
2. To review the current configuration of SAP instances and operation modes and begin defining up background processing servers, choose *Instances/operation modes*.
3. If the server you want to set up as a background processing server doesn't appear in this list, create a new instance. Choose *Instance* → *Create new instance.* Complete the instance definition in this *Maintain Instance Data* window, then save.

   Once the server you want to set up as a background processing server appears in this list, select the mode for that server then click "Choose" (F2). In the *Maintain Work Process Distribution* screen that appears, set the number of work process by selecting the current number and increasing or decreasing it with the "-" or "+" buttons. Save here, return to the previous screen, and save again. These changes will take effect when the operation mode is next activated.

   Numbers of Class A background work processes:

   0
   The default number. Use this value if you do **not** have mission- or system-critical class A jobs that must always start immediately when they become eligible to start. In this case, jobs are simply run according to their priority, or "class". Work processes are not held open for class A jobs.

   1 or more
   Use this value to guarantee that mission- or system-critical jobs start immediately once they become eligible to start. One background processing work process will always be kept free for class A jobs, ensuring no class A job has to wait for a free background work process.

   To keep background work processes available for jobs that aren't class A, be sure that the number of class A work processes is less than the *total number* of background work processes in an application server. Otherwise, only class A jobs will be eligible to run on that server.

   To reserve an entire application server exclusively for class A jobs, set the number of class A work processes to the total number of background work processes. For example, if an SAP server offers 4 background work processes and the class A reservation for the server is set to 4, only class A jobs can start at that server.

The class A setting is in effect whenever the operation mode(s) in which it is defined are active.

Prioritization Strategies for Class A Jobs

Purpose

Being able to assign a priority, or “class” (A, B, or C) to a job and being able to reserve work processes for class A jobs presents a choice between two prioritization strategies for your background jobs:

- **Critical track (class A) and two-level priority (classes B and C):** If you have mission-critical jobs whose immediate execution must always be ensured, you can set up class A as your “critical jobs” category. Reserve a work process for class A jobs and assign class A only to these mission-critical jobs. See the example below.

  Classes B and C provide secondary two-level job prioritization. Class B jobs get free background work processes before class C jobs. This secondary prioritization operates separately from the class A critical track, since class B and C jobs can never block a class A job.

- **Three-level priority for allocation of free work processes:** You can also decide to run with a straight three-level priority scheme for allocating free work processes. In this case, reserve no work processes for class A. See the example below.

  This approach guarantees no open path for class A jobs, which might have to wait if all work processes are busy, even though the work processes may be running class B or C jobs. However, class A jobs that are waiting are guaranteed to start before class B and C jobs that are also waiting. Unlike the “critical track” strategy, no work processes are kept idle for class A jobs.

  If the class A workload is not high or there is rarely a bottleneck in background processing—at least one work process is usually free—there may be no advantage in reserving a work process for class A jobs. Such a reservation in this case might simply result in a work process rarely being used.

Process

Critical track (class A) and two-level priority (classes B and C):

Example 1. A server is set up with three work processes for background processing. The *Class A* field is set to 1, keeping one work process always open for class A jobs:

```
Background work process 1:    Reserved for class A job
Background work process 2:    Free for any job
Background work process 3:    Free for any job
```

When a class A job arrives, it runs in one of the three free background work processes. One of the free background processes remains free, reserved for the time when another class A job arrives. One work process remains for jobs of classes B and C:

```
Background work process 1:    Running class A job
Background work process 2:    Reserved for class A job
Background work process 3:    Free for any job
```

Example 2. All three background processes are free and four jobs become eligible to start: two class A jobs and one each of class B and class C. The two class A jobs start immediately, and the remaining background work process is reserved for the eventual arrival of another class A job. Neither the class B nor class C jobs can start until one class A job finishes.

```
Background work process 1:    Running class A job
Background work process 2:    Running class A job
Background work process 3:    Reserved for class A job
Waiting for free work process:      Class B job

      Class C job
```

When one class A jobs finishes, the class B job can start, but the class C job continues to wait since the remaining work process is reserved for a new class A.

```
Background work process 1:    Running class A job
Background work process 2:    Running class B job
Background work process 3:    Reserved for class A job
Waiting for free work process:      Class C job
```

When the class B job finishes, the class C job can finally run:

```
Background work process 1:    Running class C job
Background work process 2:    Reserved for class A job
Background work process 3:    Free for any job
Waiting for free work process:      No jobs
```

Three-level priority for allocation of free work processes: A server is set up with three work processes for background processing, with none reserved for class A jobs.

Example. If, when all three work processes are running background jobs, a job from each class becomes eligible to run, the class A job could not start because no work process is free. When one becomes free, the class A job would be the first to run:

```
Background work process 1:    Running class x job
Background work process 2:    Running class x job
Background work process 3:    Running class x job
Waiting for free work process:      Class A job

      Class B job

      Class C job
```

When one of the running jobs finishes:

```
Background work process 1:    Running class x job
Background work process 2:    Running class A job
Background work process 3:    Running class x job
Waiting for free work process:      Class B job

      Class C job
```

When another class A job becomes eligible to start, it also must wait, because all background work processes are busy. When a work process becomes free, the class A job starts ahead of the class B and C jobs, even if these have been waiting longer.

Releasing Jobs

Use

Before a scheduled background job can start, it must be released. Only users with appropriate Authorizations for Background Processing [Seite 87] can release jobs.

Procedures

To release a scheduled job:

1. In Transaction SM37 (C*CMS* → *Jobs* → *Maintenance*) select a job or jobs to release.
2. Once the jobs to be released are selected, choose *Release* or *Goto* → *Release*.
3. Enter or change the start condition, if necessary, and choose *Save*. The job is now released.

To revoke released status

To reschedule a job or prevent a released job from being started, revoke the *Released* status. Revoking a job's released status resets the status to *Planned* and prevents the job from being started until after it has been released again.

1. In Transaction SM37 (*CCMS* → *Jobs* → *Maintenance*) select a job or jobs whose released status you want to revoke.
2. Choose *Jobs* → *Release* -> *planned*. The job's status has now reverted to *Planned.*

Deleting Jobs

Use

You delete a job for one of two reasons: the job doesn't need to be processed or the job has already run and does not need to be analyzed or documented further. Until you delete it, a job remains in the job overview. [Seite 129] When a job is deleted, it is removed from both the overview and the associated job log.

Procedure

1. Select a job (or jobs) from the Select Background Jobs screen (Transaction SM37, or choose *CCMS* → *Jobs* → *Maintenance*, complete the description of the jobs you want to delete, then choose *Execute* to get to the *Job Overview*.)
2. In the *Job Overview*, select the job or jobs you want to delete by checking the box to the right of the job name.
3. Choose *Job* → *Delete*.

Deleting Jobs That Have Dependent Jobs

If you delete a job that must be processed before another job can be started, the dependent job can no longer be started. The system will inform you of any such existing dependent, or successor, jobs. You'll then need to either reschedule or delete the dependent job.

If you try to release a job whose predecessor job was deleted, the system sets the status of the job to *Planned*. To start this job, you must release it and specify the start conditions.

Reorganizing Background Jobs

To delete background jobs in bulk, schedule the SAP program RSBTCDEL.

1. Schedule a background job [Seite 105] that has RSBTCDEL as an ABAP program step.
2. Indicate the "variant", or criteria, of the jobs you want to delete, including:
   - job name
   - name of "user", or person who scheduled job
   - job's start and end times or dates
   - "age" of the job (e.g., older than *xx* days)
   - job's status (scheduled, released, finished, cancelled)
   - event ID or parameter for event-driven jobs
3. Run this new background job.

The program RSBTCDEL should be scheduled to run regularly to flush various database tables (TBTCO, TBTCS, BTCEVTJOB, TBTCP, etc.) to keep them from getting unnecessarily large. For more information, see Required Basis Background Jobs [Seite 80].

Displaying a Job Log

Use

All message types issued by a program running in the background are stored in a job log, which you can display either to obtain information on a prematurely terminated program or to complete a detailed investigation of a particular background processing run. A separate log file is created for each job.

Deleting Log Files

You should never delete log files directly. If you want to clear the log directory, you must delete the jobs to which the logs belong. In the process, the logs are also deleted. For more information, see Deleting Jobs [Seite 121].

Job logs are held in the R/3 TemSe temporary sequential objects storage facility. TemSe is configured to always store job logs as operating system files in the shared R/3 directories. If you delete log files directly at operating system level, you create inconsistencies in the TemSe database, which you'll then need to eliminate with *Consistency check* function in Transaction SP12.

Log Messages

Log messages are generally output using ABAP keyword MESSAGE. With the exception of ABAP breakpoint statistics, no other data is normally output to the job log.

ABAP Short Dumps

If an ABAP program generated a dump when it terminated abnormally, you can display the dump by clicking on the abend message.

Batch Input Messages

Messages output during batch input sessions are *not* recorded in the job log. These messages are output only to the batch input system.

Procedure

1. Go to *Select Background Jobs* using Transaction SM37 or by choosing *CCMS* → *Jobs* → *Maintenance*.
2. Select the jobs you want to review in the job log by specifying conditions that identify the job or jobs, including job name, user name, job status, start condition, or contents of a job step. Execute this selection to see a list of jobs that match these criteria.
3. In the *Job overview* screen, select your desired job and choose *Job log*.
4. In the *Job Log Entries for <jobname>* screen, the date and time of each job step is listed along with identification of any system-generated messages that apply to each step. Message types include A (Abend), E (Error), S (Successful processing completed), I (Information), and W (Warning).
5. You can print the job log (*Job log* → *Print list*) or save the log by exporting it to a word processing, spreadsheet or local file (*Job log* → *Print list* → *Export* → *<file type>*).

6. For further information about a particular item in the job log, select the line and choose *Long text*. The resulting help screen can include further information about diagnosing the message, the system's response to the message, and how the user should follow up.

Job Storage Management

Definition

Once jobs are defined, they are stored in the R/3 database, in the following principal tables in the background processing system:

- TBTCO: The job header table (job management data)
- TBTCP: The job step table (individual processing steps in jobs)
- TBTCS: The control table for the time-driven job scheduler
- BTCEVTJOB: Jobs scheduled to wait for an event

Job logs are held in the R/3 TemSe (Temporary-Sequential Objects) database. TemSe is always set to hold job logs as files in your host systems.

Use

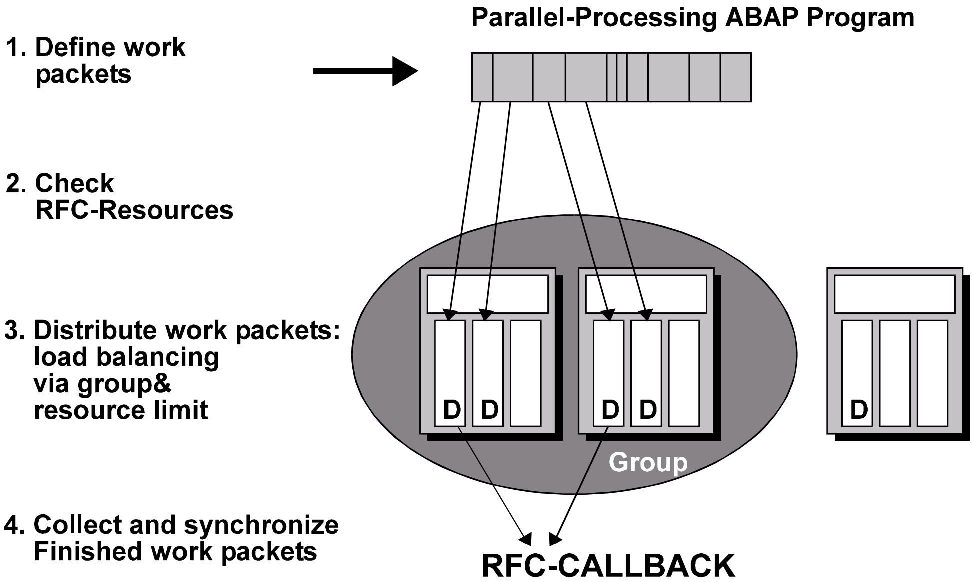
To display or delete job logs, use either Transaction SM37 (Job Overview) or the graphical background monitor [Seite 135]. Old job logs and the corresponding jobs are removed automatically by the ABAP program RSBTCDEL, which should be scheduled to run periodically (see Required Basis Background Jobs [Seite 80]).

Parallel Processing Jobs with Asynchronous RFC

Definition

For some SAP reports, the nights are just getting too short. Especially for customers with large volumes of data, some reports that generally run in the background processing system (such as material planning runs) have run times longer than the “night-time” that is available, especially if dialog users are spread across several time zones.

SAP offers a solution to the “short nights” problem: parallel-processed background jobs. Long-running reports can now implement parallel processing, which lets them parcel out the work to be done to available work processes and then collect and synchronize the results.



In parallel processing, a job step is started as usual in a background processing work process. A program that runs in a job step can be programmed to use a special variant of asynchronous RFC to have portions of the data to be processed run in parallel in other work processes. You can recognize such a program by its use of the CALL FUNCTION STARTING NEW TASK DESTINATION IN GROUP instruction to start the function modules that process the data. For information on programming, see Implementing Parallel Processing [Extern].

While the job itself runs in a background process, the parallel processing tasks that it starts run in dialog work processes. Such dialog work processes may be located on any SAP server.

Parallel processing has been implemented in some SAP applications that have long-running reports. Check the application documentation for information on such reports. The parallel processing interface is also available directly to customers.

Automatic Protection Against Resource Overuse

The asynchronous remote function call (RFC) system contains built-in safeguards against the possibility that a parallel processing job may take too many resources and affect the overall performance of your system.

A parallel-processed job is allowed to use the dialog work processes of an server only if:

- The server in question has at least three free dialog work processes. This requires that at least two dialog work processes remain free for other work on any server used by a parallel-processing job. The job can start a task in a dialog work process only if two other dialog work processes are currently free on the application server in question.
- The dispatching queue of the server is less than 10 percent full. This prevents a parallel processing job from using a server that is already very busy.

Performance testing at SAP suggests that the built-in safeguards allow significantly faster execution of background jobs without unduly affecting system performance as a whole. However, if parallel-processed jobs might compete for the same resources or might share the SAP System with time-critical applications or a full complement of interactive users, you should use RFC groups to control which resources a parallel-processed job uses.

Managing Resources with RFC Groups

By default, the group of servers eligible to be used for parallel processing is all qualifiying servers in your SAP System (CALL FUNCTION STARTING NEW TASK with the argument DESTINATION IN GROUP DEFAULT). However, you can use the “group” specification in a parallel-processing program to control precisely which resources can be used by a parallel processing job.

Monitoring

Parallel processing jobs cannot currently be specially marked in the tools for monitoring background processing. This monitoring capability will be added in a future release.

See also:

Defining RFC Groups for Parallel-Processed Jobs [Seite 127]

Defining RFC Groups for Parallel-Processed Jobs

Use

Asynchronous remote function calls (RFCs) prevent parallel-processed jobs from overusing SAP System resources. For more information, see Parallel Processing of Jobs with Asynchronous RFC [Seite 125].

By default, a parallel-processed job uses all qualified servers in an SAP System according to automatic resource-allocation rules. However, by defining RFC groups, you can control which servers can be used for parallel-processed jobs. An RFC group specifies the set of allowed servers for a particular parallel-processed job. The group used for a particular job step must be specified in the job-step program in the keyword CALL FUNCTION STARTING NEW TASK DESTINATION IN GROUP.

Procedure

1. Choose *Tools* → *Administration* → *Administration* → *Network* → *RFC destinations* and then choose *RFC* → *RFC groups*. Alternatively, call Transaction RZ12.

   This displays a table showing:

   - The names of already-defined RFC groups, or "logon groups" or "server groups"
   - A list of the servers, or "instances," in your SAP System
   - The current status (running/not-running) of each server

2. To define an RFC group, choose *Edit* → *Create assignment* and in the *Server group* field, either choose a name from the list of already defined groups or type a new name. In the *Instance* field, select one of the servers from the list. Repeat this step for each server that should belong to the group or to assign a server to more than one group. When a server belongs to more than one group, jobs that use the group will compete for free work processes on the shared server(s).

   You must ensure that the group you create is also specified in the ABAP program that is to use the group.

3. Fill in the *Determination of resources* section to complete the group definition, then choose *Copy* to save.

For information on how to configure customization for setting up RFC groups for parallel-processed background reports, see application documentation.

Usage examples

RFC groups can allow different parallel-processed jobs to run at the same time without competing for the same servers. In this case, the different groups used by the jobs would specify different sets of servers.

RFC groups can also separate parallel-processed jobs from servers being used by dialog users. In this case, the group used by a job would name servers other than those in the logon groups for users.

Possible Status of Background Jobs

Definition

The status of a background job can signify 6 different conditions, as explained in the table below.

Job Status	Explanation
Planned	Steps that make up the job have already been defined, but the start condition has not yet been defined.
Released	The job has been fully defined, including a start condition. Without a start condition, a job cannot be released. Only an administrator or a user with appropriate Authorizations for Background Processing [Seite 87] can release a job, preventing unauthorized users from running jobs without approval.
Ready	The start condition of a released job has been met. A job scheduler has put the job in line to wait for an available background work process.
Active	The job is currently running. Active jobs can no longer be modified or deleted.
Finished	All steps that make up this job have completed successfully.
Canceled	The job has terminated abnormally. This can happen in two ways: • An administrator intentionally terminates a job with Transaction SM37, *Job → Cancel active job* • A job step contains a program that produces an error, such as: – an E or A error message in an ABAP program – a failure return code from an external SAPXPG program

Only jobs with status *Planned*, *Released*, or *Ready* can still be modified, unreleased, deleted, or prevented from running.

See also:

Analyzing Job Status [Seite 142]

Managing Jobs from the Job Overview

Use

The Job Overview, or Job Maintenance, screen is the single, central area for completing a wide range of tasks related to monitoring and managing jobs, including defining jobs; scheduling, rescheduling, and copying existing jobs; rescheduling and editing jobs and job steps; repeating a job; debugging an active job; reviewing information about a job; canceling a job's release status; canceling and deleting jobs; comparing the specifications of several jobs; checking the status of jobs; reviewing job logs; and releasing a job so it can run.

Procedures

Specifying Jobs

1. To display the *Job Overview* screen, choose *CCMS* → *Jobs* → *Maintenance* or call Transaction SM37. Before entering the *Job Overview* screen, the system first displays the *Select Background Jobs* screen, where you must select the criteria for the jobs you want to manage. These criteria include:
   - Job name (which can contain a wildcard (*) to select jobs with related names or—by using the wildcard alone—any job). **Required.**
   - Name of the user who scheduled the job (the wildcard is allowed here as well). **Required.**
   - Job status
   - Planned or actual start time of the job.
   - Job start condition or event linked to the start condition
   - Job step
   - When and how often the job runs, or "periodicity"
2. You can also choose Extended Job Selection, where you can more precisely define the specific criteria that determine which jobs to select. If you have administrator Authorizations for Background Processing [Seite 87], you can display jobs in all clients. Without this authorization, only jobs in the client that you are logged on to will be displayed.
3. Once you have set the criteria for you job selection, choose Execute.

Managing and Monitoring Jobs

Once you have called the *Job Overview* screen (Transaction SM37 or *CCMS* → *Jobs* → *Maintenance*) and selected which jobs you want to manage as described above, you can choose from a wide range of management tasks.

To **copy a single existing job**, choose *Job* → *Copy*.

To **reschedule or edit job steps or attributes of a single job**, choose *Job* → *Change*. A job step [Seite 90] is an independent unit of work within a background job. Each job step can execute an ABAP or external program. Other variants or authorizations may be used for each job step. The system allows you to display ABAP programs and variants. You can scan a program for syntax errors. You can also display the authorizations for an authorized user of an ABAP job step.

To **repeat a single job**, choose *Job → Repeat scheduling.*

To **debug an active job**, choose *Job → Capture: active job.* Only a single selection is allowed. If an active job seems to be running incorrectly (e.g., running for an excessively long time), you can interrupt and analyze it in debugging mode in a background process, and then either release it again or stop it altogether.

You will be able to capture a background job only if you are logged on to the SAP server on which the job is running. To find server information in the *Job Overview,* select and mark the job, then choose *Job → Job details*.

To **review information about a job**, choose *Job → Job details*. Details displayed can include:

- current job status
- periodicity, or the repetition interval
- other jobs linked to the current job, either as previous or subsequent jobs
- defined job steps
- spool requests generated by the current job

To **cancel a job's "Released" status**, select the job or jobs from the *Job Overview* list and choose *Job → Release -> Scheduled*.

To **cancel a job from running but keep the job definition available**, select the job or jobs from the *Job Overview* list and choose *Job → Cancel active job.*

To **delete a job entirely**, select the job or jobs from the *Job Overview* list and choose *Job → Delete.* Jobs with the status of Ready or Running cannot be deleted.

To **compare the specifications of more than one job**, select the jobs from the *Job Overview* list and choose *Job → Compare jobs.*

To **check the status of jobs**, select the job or jobs from the *Overview Job* list and choose *Job → Check status.* This allows you to either change the job status back to *Planned* or cancel the job altogether. This is especially useful when a job has malfunctioned.

To **review job logs**, select a job or jobs with the status *Completed* or *Canceled* from the *Job Overview* list and choose *Goto → Job log.*

To **release a job so it can run**, select a job from the *Job Overview* list and choose *Goto → Release.*

Monitoring Background Processing Alerts

Use

Monitor the status of background processing in your SAP System with the CCMS alert monitor, Transaction RZ20.

Procedure

1. Start the alert monitor [Seite 837] by calling Transaction RZ20 or by choosing *CCMS* → *Control/Monitoring* → *Alert Monitor.*
2. On the *CCMS Monitor Sets* screen, expand the *SAP CCMS Monitor Templates* monitor set.
3. In the list of monitors, start the *Background Processing* monitor. To do this, position the cursor on the *Background Processing* monitor and choose *Load monitor*.

   The *Background Processing* monitor displays both system-wide indicators of background performance and three kinds of monitoring information for each SAP instance offering background processing.

   The table below provides further information about the contents of the monitor.

Node, or "MTE" (Monitoring Tree Element)	Explanation	App Server-Specific/ System-Wide
Utilization	Percentage of the available background processing capacity currently in use. This value is normalized across the number of background work processes. Alerts are based on average values over time. **Example:** A server with three background processes and 90% utilization has had all three work processes in use an average of 90% of the time.	App Server-Specific

Server-specific queue length	The number of jobs that are released and eligible to run but which have no free background work processes. *Utilization* and *Server-specific queue length* together show whether there is a serious shortfall in background processing capacity. High utilization and low queue length are optimal. High utilization and high queue length show that your system has not been able to satisfy background processing requirements during the alert time period. High utilization can quickly lead to queue length problems. If the monitor shows repeatedly high utilization (> 90%), you should increase background capacity.	App Server-Specific
Aborted jobs	Individual jobs that terminated abnormally. A separate red alert is generated for each aborted job.	App Server-Specific
System-wide queue length	The number of jobs that are ready and eligible to run but which have no free background work processes, averaged across all background servers. System-wide queue length gives you a single overview statistic for background processing in your system. For example, even if one background server is having trouble, a low system-wide queue length shows that the problem is specific to a particular server.	System-Wide

See also:

The Alert Monitor Tutorial [Seite 827].

Displaying Job Alerts

Use

When errors occur while running background processing jobs, such as premature termination, alerts about these errors are triggered and collected in a job log.

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Job Scheduling Monitor*.
2. From the graphical job monitor, choose *Monitor* → *Terminated jobs* to see a list of abnormally terminated jobs.
3. From the graphical job monitor, choose *Jobs* → *Display job log* to see a complete list of of logged data about a particular job.

See also:

Problem Analysis in Background Processing [Seite 137]

Job was Not Started [Seite 158]

Job is Terminated [Seite 162]

Analyzing Terminated Jobs [Seite 136]

Displaying a Job Log [Seite 122]

Graphical Job Scheduling Monitor

Definition

The graphical job scheduling monitor [Seite 135] shows you both the background jobs waiting to be processed and the currently available background processing resources. It also automatically reports malfunctions relating to background processing, such as jobs that have terminated abnormally, via an alert monitor. From here, you can also display job attributes and job logs.

Job Status Legend

You can see an online key to the meaning of the different color and pattern combinations used to show the status of jobs by clicking on the *Legend* button.

Color/Pattern	Meaning
Light gray	Scheduled but not yet released to run.
Light green	Released but the start time has not yet arrived.
Dark green	Ready. The start time has passed and the job is assigned to a work process. Jobs shown in dark green are normally just about to start.
Yellow with <<<< shading	Delayed start. The scheduled start time has been reached, but the job has not yet been started by a background processing scheduler. A lot of yellow in the display indicates that the scheduler is not operating normally. Possible error causes to be investigated are work processes that are blocked (by faulty programs) or deactivated or faulty schedulers.
Blue and white with blue frame	Currently being processed. • The blue part of the box shows the actual runtime to date. • The white part shows the estimated runtime remaining.
Red with >>>> shading	Estimated runtime has been exceeded. If the estimation was very far off, check to see if the scheduled program is running correctly.
White with cross-hatching	Ended normally
Red with cross-hatching	Terminated abnormally. Consult the job log for more information. This is also reported in the alert monitor.
Horizontal shading	Currently being processed, but no runtime statistics exist, making estimation of runtime impossible.

Using the Graphical Job Scheduling Monitor

Use

To display job scheduling information choose *Tools* → *CCMS* → *Control/Monitoring* → *Job Scheduling Monitor*.

To have this display automatically updated every three minutes, select the *Timer on* button.

To schedule and manage background jobs [Seite 106], return to the CCMS *Jobs* menu.

Changing the Monitor's Look

Reconfigure the job scheduling monitor [Seite 134], including the chart's scale, gridlines, color, etc., from the *Settings* menu.

Modify the time units used in the chart from the *Time unit* menu.

Reading the Monitor

In the far left of the monitor window, each available background work processes is shown and identified by the name of the appropriate host system. The color of a job [Seite 134] indicates its status. The length of a job symbol shows the amount of time actually required to process the job. Also shown is which background process handled the job.

From left to right, the monitor shows:

- Left of the “current time” line are jobs that have either already been processed or are currently being processed

  Based on which background processes are running which jobs, you can get a rough idea of the background work processes utilization factor. With automatic load distribution, background processing instances start a number of jobs awaiting processing corresponding to the number of free background processes. If the jobs waiting to be processed are always distributed over all instances and work processes and no work processes are left unused, then the background processing system is being fully utilized.

Right of the "current time" are jobs that are waiting to be processed.

  When jobs have already run at least once, the length of the job symbol indicates the estimated runtime for the job. The background processing system manages a statistics database, based on job names, which is used to generate these estimates. Smaller jobs and jobs for which no runtime statistics exist are shown by the monitor as a symbol with the minimum length.

  In cases where jobs are repeated periodically, runtime estimates can, over time, reach a fairly high degree of precision. Actual runtimes that deviate considerably from expected runtimes may indicate problems or program errors [Seite 137] such as an infinite loop in a program.

  You can move the start time window forward and backwards as required.

Terminated Job Analysis

Definition

If program or system errors occur, a job will terminate abnormally, and the *Cancelled* column in the Job Scheduling Monitor [Seite 134] will be marked.

A job is also shown to have terminated abnormally if it was not started within the specified start time frame. This start time frame generally expires if the system is unavailable for a long time, such as during maintenance. In such cases, jobs that are no longer needed are not executed when the system is restarted.

Error messages issued by the background system and by the scheduled program are recorded in the job log [Seite 122]. Other relevant error messages may be found in the system log, because if the background processing system cannot write a job log for any reason, all of its messages will be recorded instead in the system log.

For more information on analyzing terminated jobs, see Troubleshooting the Background Processing System [Seite 156].

You can check three sources for information on the cause of abnormal termination:

- **The System Log**

  Messages on the processing status are written to the system log while a job is being processed.

  To display messages issued during background processing, search for messages on the processing status. Also, find the SAP number assigned to the background work process with the *Process overview* function. This number stays the same as long as the number of processes in a server does not change.

  To reduce the possibility of a message being issued more than once, the work process does not write a repeated message to the system log until after a specific delay period. When the problem is solved, the system writes an appropriate message for that in the system log as well.

  Assume an error in the database system prevents background processing. During the first hour of the problem situation, a warning message is written in the system log several times at increasing intervals. After this first hour the message is sent hourly. Once the malfunction has been corrected, the system reports that normal operation has been resumed.

- **The Job Log**

  All messages issued by an ABAP program with the MESSAGE statement are recorded in the job log.

- **Output of an ABAP Short Dump**

  If an ABAP program in a job step terminated abnormally, you can access the ABAP short dump from the job log by clicking on the error message.

Problem Analysis in Background Processing

Definition

Problem analysis is supported by background job logs that record all messages issued by programs run in the background. The only exception to this is that batch-input messages are recorded in the batch input log available from Transaction SM35 instead of in the background processing log. Whether or not external program output is logged depends on whether the external program is started synchronously or asynchronously, as well as how the job-step program-output control flags are set. In the case of an asynchronous background job, the background processing system does not wait for an external command or program to complete before starting the next job step or completing the job. Output from this program would not be captured in the job log.

Background job logs also contain job progress and status messages from the background processing system itself. For more information, see Displaying a Job Log [Seite 122].

Use

A variety of tools for analyzing and checking the background processing environment are also available, and you can use standard ABAP debugging to “capture” a job while it is running. The best places to start looking for troubleshooting data are the job log and the SAP system log. Find a problem-solving guide and help in using these tools in Troubleshooting the Background Processing System [Seite 156].

R/3 Analysis Tools: Overview

Definition

Only those general R/3 analysis tools are considered here which are relevant to problems in background processing.

You call these tools from the R/3 initial screen. You can use them for:

- Analyzing the Work Processes and System Log of an Application Server [Seite 140]
- Analyzing ABAP Runtime Problems [Seite 141]

You can find a complete description of the general R/3 analysis tools in the following R/3 documentation:

- BC - System Services
- BC - ABAP Development Workbench Tools

Using the Background Processing Analysis Tools

All the special analysis tools for background processing are available from the CCMS initial screen, which you display by choosing *Tools* → *CCMS.*

You can use these tools for the following purposes:

- Analyzing Job Status [Seite 142]
- Analyzing Resources With the Graphical Job Monitor [Seite 146]
- Analyzing Status, Configuration and Administration Data [Seite 148]
- Analyzing Parts of the Runtime Environment [Seite 151]
- Analyzing Problems With External Commands and Programs [Seite 154]
- Analyzing Problems With Events [Seite 170]

Analyzing the Work Processes and System Log of an Application Server

Use

There are two work process types that are of particular interest for background processing:

- Background work processes

  These are process jobs that are run in the background.

- Dialog work processes

  These process parts of the background processing runtime environment, for example, the time-based job scheduler.

For a list of all the parts of the runtime environment and the work process types in which they run, see Analyzing Status, Configuration and Administration Data [Seite 148].Tools for Monitoring the System

It is often necessary to investigate these work processes using the following criteria:

- Status (waiting, running,...)
- Which program or user ID is running in the work process?
- Trace and error information written to the developer traces (dev files) by the work processes
- System log entries

Procedures

From the initial screen, choose *Tools* → *Administration* → *Monitor* → *System monitoring* → *Servers.*

1. To analyze the system log, choose *Goto* → *System log.*
2. To analyze the work process statuses, choose *Goto* → *Processes*

   The system displays a list of all active work processes on the selected application server. In addition to displaying status information, you can also terminate work processes and programs here and analyze them with the ABAP Debugger.

Sometimes it is necessary to execute an action on a particular application server, such as start a transaction. To do this, you can log onto any application server by choosing *Goto* → *Remote logon*.

See also:

Tools for System Monitoring [Extern]

Analyzing ABAP Runtime Problems

Use

An error in an ABAP program that is running in a background generally causes an immediate abnormal termination of the background job. The only exception to this rule is in the event that the error was returned by a function module called in the program, and the program is set up to trap the error as an exception. In this case, control is returned to the ABAP program, which can attempt to recover from the error and can continue running.

A background job is terminated abnormally if, for example, the program does not handle the error or if a division by zero occurs.

In most cases, an ABAP dump is generated when an ABAP program terminates abnormally. This dump contains a description of the precise cause of the problem.

Procedure

1. To call the analysis tool from the R/3 initial screen, choose *Tools* → *ABAP Workbench* → *Test* → *Dump analysis*.
2. On the initial screen, you must specify whether you want to see today's or yesterday's dumps. If this selection criterion is not precise enough, you can enter more specific criteria by choosing *Goto* → *Select short dump*
3. You can request a list of all ABAP dumps found by choosing *Edit* → *Display list*. You can then display a selected dump and analyze it. To do this, choose *Short dump* → *Dump analysis*. For more information, see "dump analysis" in ABAP Workbench Tools [Extern].

Analyzing Job Status

Procedure

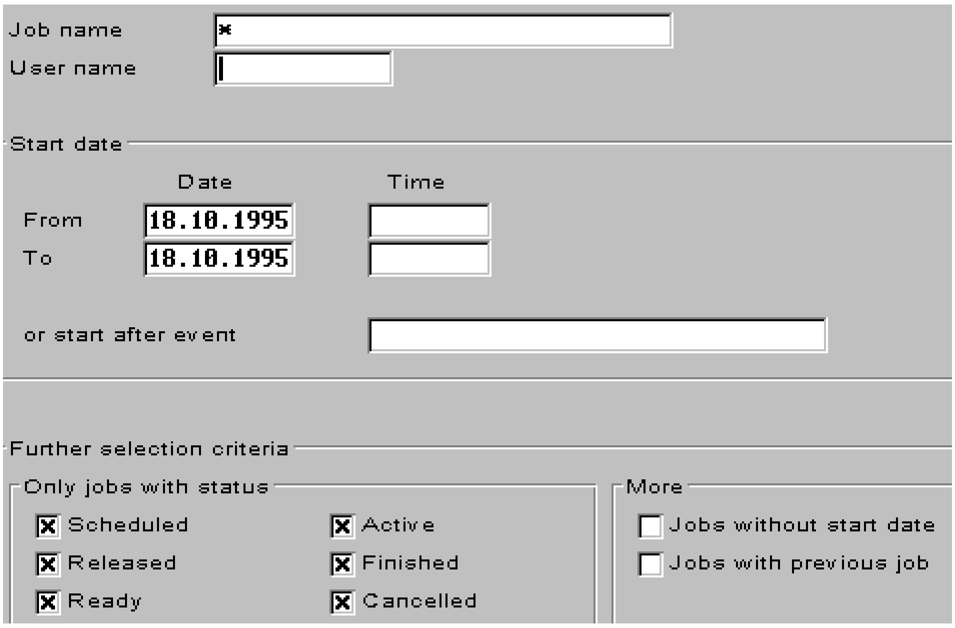
From the CCMS initial screen, choose *Jobs* → *Maintenance.*

All background processing jobs can be subjected to an analysis of job status. This analysis is concerned with the following criteria:

- Existence
- Attributes

  (requested/actual start time, steps, target server, job class and so on)
- Status (scheduled, released, active, nd so on)
- Activity of a running job

You can restrict the number of jobs selected on the request screen:



It is important whether or not the user who makes the selection has an administrator authorization for background processing. If a user has administrator administration, then jobs will be selected across all clients. If not then jobs will only be selected in the client onto which the user has logged on.

The system displays a list of the jobs that were found:

Job name	Scheduled	Released	Ready	Active	Finished	Cancelled
COLLECTOR_FOR_PERFORMANCEMONITOR					X	
COLLECTOR_FOR_PERFORMANCEMONITOR		X				
DD-CHK-A-ECHKALL_ACT		X				
DD-CHK-CHKALL_DB		X				
DD-CHK-CHKALL_NEW		X				
DD-CHK-CHKALL_NT		X				
DD-CHK-CHKALL_TA		X				
DD-CHK-F-FCHKALL_ACT						X
DD-CHK-G-KCHKALL_ACT						X
DD-CHK-L-OCHKALL_ACT						X
DD-CHK-P-RCHKALL_ACT						X
DD-CHK-S-SHCHKALL_ACT						X
DD-CHK-SN-SCHKALL_ACT						X
DD-CHK-T-ZCHKALL_ACT						X
DELETE_ALL_ABORTED_JOBS					X	
DELETE_ALL_FINISHED_JOBS					X	
DK-GROUP						X
DK7					X	

This overview displays information that is of particular importance for problem analysis, that is, job status. The list is usually sorted alphabetically, but you can sort it according to other criteria (by choosing *Edit* → *Sort*). You can sort it chronologically, for example, so that you can easily see the jobs that ran over a given time period.

The following functions are especially important for problem analysis:

- *Job log*:

  This contains information about any problems which occurred in the job at runtime, such as whether an ABAP program was terminated or an external program could not be started.

  This also contains a list of all messages which were output either by ABAP programs with the MESSAGE statement or by external programs.

- *Job* → *Display*:

  Displays all data belonging to a given job, such as the start date, steps, spool lists generated by steps, etc. The job detail information also shows on which host and in which work process the job was executed.

- *Job* → *Change*:

  To change job data such as target server, start date, steps, and so on.

- Spool List :

  You can access the spool list of a job directly from here.

- Steps:

  You can access the step data of a job directly from here.

- *Job* → *Capture*:

  This can be used to pause and analyze an active job which is in the middle of executing an ABAP program. An ABAP debugger window is opened, showing the ABAP program code at the point at which it is being executed. The code can now be analyzed. You exit the debugger by choosing *Debugging* → *Continue*.

The ABAP program (the job) then continues to execute from the point at which it was stopped. You cannot use this function on external programs.

- *Job → Cancel job*:

  This immediately terminates an active job.

- *Job → Check status*:

  This allows you to check whether the status of a job as shown in the list is the same as the real status of the job. There is a database table that contains the status information of the jobs.

  Situations sometimes occur (for example, a problem in the connection to the database) that prevent the background processing runtime system from entering the current status of a job in the appropriate database table. This will cause a discrepancy between the real status of a job and the status that was entered in the database.

Analyzing Resources With the Graphical Job Monitor

Use

You can use the graphical job monitor to display an overview of the resources which are available in an R/3 system for background processing.

The job monitor helps answer the following questions, which are important in problem analysis:

- Which are the job servers?
- How many background work processes are available on each job server and to what extent are these already in use?
- Which jobs are currently running?
- Which jobs are scheduled to run?

Procedure

To analyze resources with the graphical job monitor, from the CCMS initial screen, choose *Control/Monitoring → Job scheduling monitor.*

The vertical broken line indicates the current status of the job server. By choosing *Time unit* you can set the unit of time to be used. Jobs are displayed as rectangles. You can click on a rectangle to obtain information about that job, such as its name, job class, start date, and so on.

Analyzing Status, Configuration and Administration Data

Procedure

To analyze status, configuration and administration date:

From the CCMS initial screen, choose *Jobs* → *Check environment*

Certain requirements have to be met before the background processing can work. For example, the profile parameters and the authorizations have to be set. This analysis transaction can be used to obtain this information and to make the checks.

You have the choice between two types of test:

1. Simple tests (initial screen)

   You can check whether the profile parameters are correctly set and whether the user ID SAPCPIC exists. This check can be made either for a particular background server or for all of them. It is required for starting external programs.

2. Additional tests (*Goto* → *Additional tests*)

   In addition to the fundamental tests mentioned above, you can also run checks on the following:

- User authorization for background processing.
- Names of all job servers
- Functionality and state of the TemSe subsystem, which is required for writing the job logs
- Consistency of database tables

  The background processing stores job data in several database tables. These tables can be checked for consistency. This test is especially important if there have been problems in the database and a check needs to be made of whether all the job data still exists.

  You can also run the consistency check on database tables as a background job. Doing so makes sense if the check runs out of time and is terminated abnormally when you run it directly from the *Check environment* function. (This problem occurs only if you have extremely large background processing tables, that is, a very active background processing system. In this case, the check report may exceed the time limit on reports run in interactive mode, in the foreground. The check report is then terminated with the message `Time limit exceeded`.)

  To run the consistency check in the background, schedule ABAP report RSBTCCNS as a job step in a background job. The report comes with two predefined variants that you can use in the job step. These are:

- Variant SAP&AUTOREPNO: Use this variant to have consistency problems only reported in the RSBTCCNS list output. No automatic repair of the problems is done.
- Variant SAP&AUTOREPYES: Use this variant to have consistency problems logged and also automatically repaired by the report.

The report logs its results in a spool request which you can display or print.

- Current status of the background work processes of either a particular job server or of all of them.
- Number of entries in the dispatcher queue for background work processes

  This queue should always be empty. If, however, there are entries in the queue, then the background work processes in the application server are not currently able to process any more jobs.

  The results of the tests can either be displayed as a list or written to the file ‘BTCSPY’ in the working directory of the application server to which you are logged on.

Analyzing Parts of the Runtime Environment

Use

The background processing runtime system consists of the following components (control objects):

- Time-based job scheduler:

  Responsible for starting jobs which have a start date and time specified.

  If there are jobs with a scheduled start time of *After event, At operation mode* or *After job* that cannot be started at their scheduled start time (for example, if there are no free background work processes), then these jobs will be started at the next possible opportunity. The time-based scheduler is also used to start these jobs.

  The scheduler is started periodically by the dispatcher in a dialog work process, as long as there is at least one background work process on the application server and the periodic value is greater than zero. The number of background work processes is specified in the profile parameter *rdisp/wp_no_btc*. The periodic value is set in the profile parameter *rdisp/btctime*.

  (The default value is 60 seconds)

- Event-based job scheduler:

  If an event is raised in the R/3 system, a check is made using the event based scheduler to see if there are any jobs waiting for this event. If there are, then the event based scheduler makes sure that these jobs are started. The scheduler always runs in a dialog work process on the application server which the profile parameter *rdisp/btcname* points at.

- Job starter:

  A job that is due to be executed will be started by the job starter. This performs all preparatory work such as reading the job data from the database and starting the job steps. The job starter runs in a background work process.

- Switching between operating modes:

  The time-based job scheduler uses a table of times during each run to check whether a switch of operating mode is necessary. If it is, then the scheduler initiates the switch in a dialog process on the application server on which the scheduler is running.

- Zombie cleanup:

  When an R/3 system is started, then a check is made for ‘job corpses’ (zombies). These are jobs which have the status *ready* or *active*. Since it is of course impossible for jobs with one of these status values to exist when the R/3 system is started, they are set to have status *scheduled* or *terminated*. ‘Job corpses’ are created when, for example, an application server on which a job is running is powered down before the job has finished properly. The background processing runtime system can no longer correct the status contained in the database.

- Starting external programs:

  This component allows external programs to be started as part of a job step. An external program is started from the background work process in which the job is running.

  Each of these components (control objects) can be analyzed separately.

Procedure

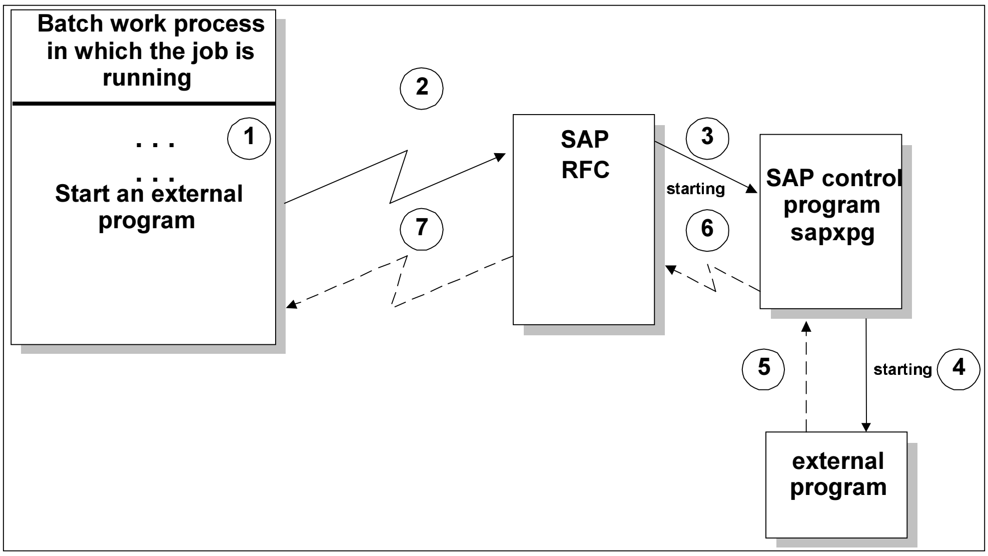
1. To analyze parts of the runtime environment from the CCMS initial screen, choose *Jobs* → *Background objects*
2. Choose *Maintain.* The system displays a list with one entry for each host and control object.
3. Choose *Control object* → Change. The system displays a dialog box in which you can activate a control object for tracing. This object is valid either just for the next run or permanently.
4. Choose *Action log* to find out if the control object ran after the trace was switched on.
5. As long as the control object has run at least once, you can have a look at the trace file on the appropriate application server. To do this, choose *Tools* → *Administration, Monitor* → *System monitoring* → *Servers, Goto* → *Traces (*Transaction SM51)
6. In the case of control objects that run in dialog work processes, it is not possible to predict in which of the possible work processes the object will be processed. However, it is quite likely to be work process 0 or 1. The best approach is to sort the list of trace files according to date and time The trace entries should be found in one of the most recently written trace files. Look through the files for entries that have an 'L' in the first column.

   For control objects that run in a background work process, you can find the work process number in the job detail data.
7. If you cannot evaluate the trace information yourself, you can send the contents of the trace file to SAP for further analysis.

Analyzing Problems With External Commands and Programs

Purpose

The following R/3 system components play a part in the processing of external programs in a job:



This figure illustrates how the control program is started from a background work process via RFC and how the control program calls the external program itself. It also shows how the control program sends a message back to the application server via RFC.

The following problems can arise when processing external programs in a job:

- The external program cannot be started properly
- The external program is not able to send its results back to the background processing runtime system

Process Flow

These problems can be analyzed as follows

1. **Check the system requirements:**

   Check that the target system is set up correctly for running external commands and programs. See Prerequisites for Running External Commands and External Programs [Seite 95].

2. **Check the job log for error messages**

3. **Switch on the trace**

   If the problem has not yet been solved, then you should switch on the trace for the external program. When you are scheduling an external program in the dialog (Transaction SM36 / SM37), you can turn on the SAPXPG trace with for a particular job step with trace option in the job step *Control flags*. You can also turn on the SAPXPG trace in these additional ways, without modifying the job step definition:

– For a particular external command (all job steps that use the command), with the trace option in Transaction SM69

– For all external command processing at a particular host system, by setting the environment variable **`sapxpg_trace`** in the host system in which an external job is to be started. Format and values:

  `sapxpg_trace=<trace level>` where trace level is `1`, `2`, or `3`. `3` is the most detailed tracing. `0` turns tracing off again.

The trace records are written in the trace files listed above.

All of these options turn on SAPXPG's tracing, but have no effect on any tracing facility at the operating-system level or in the external program.

If you use the function module JOB_SUBMIT to schedule a job step, then you must make the following parameter setting: EXTPGM_SET_TRACE_ON = 'X'

This has the effect of creating the following trace files in the home directory of the gateway ID on the host on which the external program is to be executed:

– *dev_cp* Log of the R/3 control program *sapxpg*

– *dev_xpg* Log of all output from the external program

Check both of these trace files for error messages. You may also find error messages in the gateway trace file *dev_rd* and in the system log on the host on which the job was started.

See also:

Analyzing the Work Processes and System Log of an Application Server [Seite 140]

Troubleshooting the Background Processing System

Purpose

Problems in R/3 background processing can be classified as follows:

- Configuration problems
- Administration problems
- Runtime problems

The following sections contain descriptions of the various analysis tools and also case studies illustrating their usage.

Case Studies

Job was not Started [Seite 158]

Job is Terminated [Seite 162]

Job Log Cannot be Displayed [Seite 165]

Job Status Remains “Active” [Seite 168]

Job was Not Started

Procedure

Analysis	Procedure
1) Does the job have a start date?	Display the job's start date (see Analyzing Job Status [Seite 142]) Does the job have a start date? Yes: Continue with step 2. No: Assign the job a start date End of analysis
2) Does the user who scheduled the job have authorization to release jobs?	Display the authorizations that the user has for background processing (see Analyzing Status, Configuration and Administration Data [Seite 148]) Does he have release authorization? Yes: Continue with step 3 No: Either give the user release authorization or have the batch administrator release the job. End of analysis
3) Has the start date been reached?	Yes: Is start date = 'after Event' and was the event raised using the *sapevt* program? Yes: Check whether *sapevt* is working (see Analyzing Problems With Events [Seite 170]) – Is *sapevt* Notifying problems? Yes: End of analysis No: Continue with step 4 No: Either wait until the start date is reached or assign the job a new start date (see Analyzing Job Status [Seite 142]) – End of analysis

4) Are the resources necessary for executing the job available?	– Call the graphical job monitor (see Analyzing ABAP Runtime Problems [Seite 141]) – Does the job have a target server? Yes: Is there a free background work process on the target server for the job class requested? Yes: Continue with step 5. No: Either set up a minimum of one background work process on the target server for the job class (see Analyzing Parts of the Runtime Environment [Seite 151],“Switching between operating modes”) or remove the target server from the job definition (see Analyzing Job Status [Seite 142]) – End of analysis No: Is there at least one host with a free background work process for the job class requested? Yes Continue with step 5 No: Either wait until a background work process becomes free or set up an additional background work process for the job class (see Analyzing Parts of the Runtime Environment [Seite 151],“Switching between operating modes”) – End of analysis
5) Can the time-controlled or the event-controlled job scheduler be started?	– Does the job have a target server? Yes:Check the profile parameters of the background processing on the target server (see Analyzing Status, Configuration and Administration Data [Seite 148]) No: Check the profile parameters of the background processing on all job servers (see Analyzing Status, Configuration and Administration Data [Seite 148]) – Are errors evident in the log? Yes:Fix the problems, e.g. change the profile parameters etc. – End of analysis No: Continue with step 6

6) Are there unusual entries in the system log?	– Does the job have a target server? Yes: Check the system log on the target server for entries (see Analyzing the Work Processes and System Log of an Application Server [Seite 140]) No: Check the system logs of all job servers for entries (see Analyzing the Work Processes and System Log of an Application Server [Seite 140]) – Did you find entries ? Yes: Can you solve the problem yourself? Yes: End of analysis No: Continue with step 7. No: Continue with step 7.
7) Record the background processing	– Record the component of the background processing which is responsible for starting the job (see Analyzing Parts of the Runtime Environment [Seite 151]) – Analyze the trace information – Can you solve the problem yourself? Yes: End of analysis No: Send the trace to SAP

Job is Terminated

Procedure

Analysis	Procedure
1) Are there error messages in the job log?	– Display the job log (see Analyzing Job Status [Seite 142]) – Are there error messages? Yes: – View the error message long texts (choose *Goto* → *Long text*) Due to the large number of different problems which can cause termination of a job, it is not possible here to give a generalized procedure for analyzing the problems. There follows a list of some of the more commonly observed problems: – A variant of an ABAP program cannot be found, because it has been deleted – There is a syntax error in an ABAP program – There is a runtime error in an ABAP program, e.g. divide by zero – An external program cannot be started (see Analyzing Problems With External Commands and Programs [Seite 154]) No : Could the problem be solved ? Yes: End of analysis No: Continue with step 2
2) Are there unusual entries in the system log?	– Does the job have a target server? Yes: Check the system log on the target server for entries (see Analyzing the Work Processes and System Log of an Application Server [Seite 140]) No: Check the system logs of all job servers for entries (see Analyzing the Work Processes and System Log of an Application Server [Seite 140]) – Did you find entries? – Can you solve the problem yourself? Yes: End of analysis No: Notify SAP of the problem

Job Log Cannot be Displayed

Analysis	Procedure
1) Can job logs be created?	– Perform the following test on all job servers: – Schedule the job with the target server = name of host on which the job server is running. You can specify the R/3 ABAP program RSPARAM as a step (for example). Start date of job = 'Immediate' – Log on to job server (see Analyzing the Work Processes and System Log of an Application Server [Seite 140]) – Can the job log be displayed (see Analyzing Job Status [Seite 142])? Yes: End of analysis No: Continue with step 2
2) Are there unusual entries in the system log of the job server?	– Check the system log of the job server (see Analyzing the Work Processes and System Log of an Application Server [Seite 140]) – Are there any 'permission denied' entries, which indicate that the job server is not permitted to create a job log? Yes: Change the access permissions belonging to the directory in which the job log was created, so that the user ID in which the job server is running (generally c11adm) is given read and write permission for the directory – End of analysis No: Check on the job server whether the TemSe subsystem used for creating job logs is working (see Analyzing Status, Configuration and Administration Data [Seite 148]) – Does the test report errors? Yes: You can find entries in the system log which describe the problem in more detail End of analysis No: Continue with step 3

Job Log Cannot be Displayed

3) Is the job log supposed to be displayed on an application server other than the job server?	– Find out which host the job should be executed on (see Analyzing Job Status [Seite 142]) – Is the system attempting to display the job log from an application server which is running on a different host than the host on which the job is being executed? Yes: Do the application server and the job server each have their own R/3 global directory ? Yes: The job log cannot be displayed because the application server cannot access the global directory of the job server. Make the necessary changes so that both servers have the same global directory. End of analysis No: Report the problem to SAP

Job Status Remains “Active”

Procedure

Analysis	**Procedure**
1) Is the job really still running?	– Check the status of the job explicitly (see Analyzing Job Status [Seite 142]) – Is the job still active? – Yes: To find out what the job is currently doing, you can ‘capture’ it (see Analyzing Job Status [Seite 142]). However, this only works for ABAP programs that are executed as job steps. It is not possible to analyze external programs in this way. For an external program, the check if it is still active must be made at the operating system level. – Is the job waiting for an external program that is being executed in the job to be terminated? Yes: Check whether the user SAPCPIC exists in the client in which the job is running (see Analyzing Status, Configuration and Administration Data [Seite 148]) No: continue with step 2
2) Was the job server restarted while the job was still running?	– Find out the start time and the execution host of the job (see Analyzing Job Status [Seite 142]) – Find out from the system log of the job server when the job server was last started. – Is the start time of the job server after the start time of the job? – Yes: – The Job Server was restarted while the job was running. The runtime environment of the background processing system was not able to update the job status in the database. – You can manually set the job status to aborted (see Analyzing Job Status [Seite 142]) – End of analysis – No: Contact your SAP consultant.

Analyzing Problems With Events

Purpose

Problems can occur in jobs with the start time *After event*.

The analysis of these problems depends on whether the event was raised from within the R/3 system or from outside.

Process Flow

If the event was raised from within the R/3 System (for example, through the function module BP_EVENT_RAISE), you must check the name of the application server.

1. For processing events, an application server is specified in the profile parameter *rdisp/btcname*.
2. An event-based job scheduler is started on this server. This checks whether there is a job which is waiting for this event which has occurred (see Analyzing Parts of the Runtime Environment [Seite 151]). It is important that the parameter *rdisp/btcname* contains the name of an active application server.
3. Execute the ABAP program RSPARAM on the server on which the event was raised. This program displays the values of all active profile parameters on the server.
4. Check whether the name specified in *rdisp/btcname* is found in the list on your application server.
5. Another possible cause of errors is an invalid event name. Check the system log for the server on which the event was raised for entries indicating this kind of an error (see Analyzing the Work Processes and System Log of an Application Server [Seite 140]).

If the event was raised from outside the R/3 System (via the *sapevt* program at the operating system level), record a trace and check the trace file:

6. If you want to raise events from outside the R/3 system, that is, from the operating system level, you can do this with the R/3 program *sapevt* (see also Starting External Commands and Programs [Seite 100]).
7. If you specify the *-t* option when calling *sapevt*, the system creates a trace file with the name *dev_evt*. If *sapevt* discovers any problems, these are displayed in the trace file.

Control Object Types

Definition

You can test and analyze the following objects:

Job starter	The part of the background processing system that starts a job and executes the various job steps.
Time-driven scheduler	The part of the background processing system that prepares jobs for background processing according to their scheduled start times. It runs at regular intervals and determines whether the scheduled start times of released jobs have been reached. Any such jobs found are started by the job starter in the local application server as soon as background work processes become available. The time interval at which the scheduler is started is laid down in parameter `rdisp/btctime`.
Event-driven scheduler	The part of the background processing system that prepares jobs for background processing dependent on defined events. It is triggered each time an event occurs and checks whether any released jobs are linked to this event. Affected jobs are started in the local application server by the job starter as soon as background work processes become available.
External program	The SAP part of the control program that starts an external program from within the background processing system If required, the **external program** can store the standard output or standard error output of the external program and can receive the return code from the program, which also goes to the job log.
Zombie clean-up	Removes jobs interrupted by a system crash. Although they have been abnormally terminated, these jobs would have status **active** because normal processing of the termination could not take place. When the zombie clean-up is triggered, it determines which jobs have status **active** but are not currently being processed and then changes the status of these jobs to **terminated**, allowing you to remove or reschedule the jobs. In a multi-server environment, another server handles the clean-up for the crashed server. On the other hand, in a single server environment, the zombie clean-up is done when the crashed system is restarted.

Configuration

Purpose

This component enables you to configure your SAP System using the Computing Center Management System (CCMS). With CCMS, you get extensive support in system configuration and can perform many functions from within the SAP System.

Integration

The CCMS is a standard part of the SAP System. Therefore you can use it without problem as part of your overall system configuration activities.

Features

The main functions that you can use the CCMS to configure are:

- Logon load balancing
- Operation modes
- Instances
- Profiles

Logon Load Balancing

Purpose

Logon load balancing is used to distribute SAP users dynamically across application server instances. You can maximize the efficiency of every defined workgroup by setting up various logon workgroups that consist of one or more application instances,.

You can assign one or more application servers to specific workgroups or applications. When the users log on to the system, they are automatically logged on to the server that currently has the best performance statistics and/or the lowest number of logged on users.

Workgroups are configured and maintained centrally from within the SAP System. Each application server group is assigned maximum threshold values for response time and the number of users that can log on to a specific group server. Especially important workgroups with time-critical transactions can be assigned to application servers with improved response time behavior. Setting up workgroups makes it easier to predict the behavior of each server, thereby enabling you to set up each server according to the working applications.

To log on to an SAP System, the user only needs to know the name of the SAP System and the logon group. Host name and system numbers are no longer needed once you have logged on.

See also:

The SAP Logon [Seite 182]

Configuring Logon Groups [Seite 187]

Questions and Answers: Logon Load Balancing [Seite 177]

Recommendations for Logon Load Balancing and Logon Groups

Definition

Logon Load Balancing

Logon load balancing increases the efficiency of various workgroups that have been defined, in terms of performance and system resource consumption, by distributing users across available application servers based on requirements for workgroup service and load sensitivity.

In a system landscape where there are several application server instances, specific servers are best assigned to specific application workgroups, whereby the available resources and buffers of that server are set specifically to the application and not shared with other applications. This is illustrated below:

Recommended:

With logon load balancing and servers assigned to specific applications:

Logon Group FI/CO	Logon Group SD		
Server A FI/CO	Server B FI/CO	Server C SD	Server D SD

With logon load balancing and shared, or homogeneous, server properties across logon groups:

Logon Group FI/CO	Logon Group SD				
Server A FI/CO	Server B FI/CO	Server C FI/CO	Server C SD	Server D SD	Server E SD

Not recommended:

With logon load balancing and servers available to all applications

Logon Group PUBLIC			
Server A FI/CO + SD	Server B FI/CO + SD	Server C FI/CO + SD	Server D FI/CO + SD

With only two servers with logon load balancing and servers assigned to specific groups:

Logon Group FI/CO	Logon Group SD
Server A	Server B

Logon Groups

Each SAP application has different resource requirements. Certain applications may therefore require more servers and logon groups. For example, you should assign separate servers for the R/3 application component PP.

Recommendations for Logon Load Balancing and Logon Groups

In general, each logon group should have two servers. If one server is not available, users are automatically connected to the second server. Servers can be added or removed while the R/3 System is running.

If it is not practical for you to assign separate servers to closely integrated R/3 applications such as the application components SD-MM and FI-CO, you should assign common logon groups to these applications.

See also:

SAP Logon Menu [Seite 182]

Configuring Logon Groups [Seite 187]

Questions and Answers: Logon Load Balancing [Seite 177]

Questions and Answers: Logon Load Balancing

Process

Q: How can I define a default configuration for different PCs?

A: First create or check the files described in PC Installation [Seite 180]. From the SAP Logon, create the desired entries. Copy the files `sapmsg.ini`, `saproute.ini`, `saplogon.ini` and `services` to each of the PCs. The SAP Logon then displays the configured entries on all these PCs.

Q: Can I connect to a system that is not included in the file `sapmsg.ini`?

A: Yes. If you create an entry as described in Editing Entries Manually [Seite 190], you do not need corresponding entries in the configuration files.

Q: Can I use the SAPgui program in the same way as in previous R/3 Releases?

A: Yes, `sapgui` behaves as it did in previous R/3 releases.

Checking the Logon Load

Use

You can check the logon load by displaying a list of the servers that are available and an overview of load distribution in your system.

Procedure

1. Choose *CCMS* → *Configuration* → *Logon groups* → *Goto* → *Load distribution.*

   This list also shows the current performance status of the application servers that are both assigned to logon groups and currently running. Every five minutes, or after every five logons, each application server writes its own performance statistics data to a memory-resident table on the message server. The current logon server is refreshed for each group.

2. To refresh the performance status of an application server, double-click an application server line in the *Instance* column.

To display which users are logged on:

3. Choose *Goto* → *Back* and then choose *Goto* → *User list*.

   The list shows how users are distributed over the different servers, and whether a particular server is full. You can sort this list by:

   - user
   - instance
   - terminal
   - time of the last user action.

PC Installation

Use

When you upgrade an SAP System from Release 2.x to Release 3.x, copy the file saplogon.exe to the directory in which the sapgui.exe file was installed. The saplogon.ini file should be saved locally in the Windows directory. You should save the sapmsg.ini and saproute.ini files, however, in the frontend installation directory.

Procedures

1. In your Windows installation directory, check the sapmsg.ini file.

   With Windows 3.1/3.11, the files are normally in the directory c:\windows. Wiith Windows NT they are normally in the directory c:\winnt.

   The first line must contain the entry [Message Server]. For each SAP System that is available, the following entry should exist:

   <SID> = <host name on which the message server is running>.

   ```
   [Message Server]
   C11 = hostname.yourdomain
   O01 = oss001
   ```

   saproute.ini

   The first line must contain the item [Router]

   For each possible network connection, the following entry should exist:

   <route name>=<route>

   Here <route> is formulated as:

   /H/<hostname on which the SAProuter is running>/S/<service>

   You can concatenate several of these strings if the connection uses multiple SAProuters.

   ```
   [Router]
   SAP Walldorf=/H/hostname.your.saprouter/S/sapdp99/H/147.204.64.1/S/sapdp99
   SAP Foster City=/H/hostname.your.saprouter/S/sapdp99/H/147.204.69.1/S/sapdp99
   ```

2. Ensure that the services file (with Windows 3.1/3.11 normally in the directory c:\windows) for each system contains the following entry in the sapms.ini file:

sapms<SID> <nr.>/tcp

sapmsC11 3670/tcp

sapms001 3616/tcp

The configuration data is stored in the saplogon.ini file in the Windows installation directory. You can configure the file names and the path specifications of the saplogon.ini file as you like. You should also save this individually-configured file locally.

If you are the administrator and want to preset a particular selection for the user, you can either deactivate:

- selection options for Group Selection and Server Selection
- unrestricted user logon

To deactivate the selection options for *Group Selection* and *Server Selection*:

To log on to the system, you only need the saplogon.ini file. If sapmsg.ini and saproute.ini are not available locally, the selection options are not available to the user.

Alternatively, you can call the saplogon.ini file and insert the entry Restricted Mode=1 in the *Configuration* section. To be on the safe side, save saplogon.ini as a write-protected file. You could also set the environment variable to slg_restriction=1 for the entire system.

To deactivate unrestricted logon for the user:

You can use the saplgpad program so that users cannot edit or log on freely to the SAP Logon.

To log on to the system, you only need the preconfigured file saplogon.ini.

sapmsg.ini and saproute.ini are not required at runtime.

saplgpad and saplogon use .ini files from the installation directory. If no .ini files are available in the installation directory, the programs access the Windows directory. If the system still cannot find any .ini files, saplogon creates saplogon.ini in the Windows directory as well as sapmsg.ini (if this is required).

When the user logs on, the system displays the SAP Logon. This only contains the selection list.

The SAP Logon

Definition

The SAP Logon is the Windows program that you use to log on to SAP Systems on Windows PCs. It mediates between the SAP System and the SAPgui user interface. The SAP Logon displays a list of available SAP Systems and automatically selects servers with the best current response times. You can add available systems or servers to this menu.

Use

When you log on to the SAP System, you can:

- Log on to a specific application server.
- Log on to a group. In this case, the application server with the best response time is selected automatically.

From Release 4.6A, when you log on to the SAP Logon, the SAP Logon icon is displayed in the system tray of the system taskbar (in the bottom right-hand corner of the screen). You can maximize or minimize the SAP Logon by clicking the icon using the left mouse button.

If you click the icon using the right mouse button, you can display a list of connections to SAP Systems that are already open.

From 4.6A you can also use the mouse to increase the size of the SAP Logon. This enables you to see all of the following information:

- Description of the SAP System and its system ID
- The group or server
- The system number
- The message server
- SAP routers.

See also:

Adding a New Logon User Group [Seite 186]

Adding an Application Server [Seite 189]

Configuring the SAP Logon [Seite 184]

Questions and Answers: Logon Load Balancing [Seite 177]

Logging on to an SAP System

Use

You use the SAP Logon to log on to an SAP System.

Procedure

1. Display the SAP Logon by choosing *Start* → *SAP Frontend* → *SAPlogon.*
2. Position the cursor on the SAP System that you require and choose *Logon.* Alternatively, you can start the system by double-clicking on it in the list of available systems.

   You can also start an SAP System using a command line parameter. For example, if you enter the line `saplogon BIN [public]`, an icon is displayed in the selection. The system automatically uses the server with the least load.

See also:

Configuring the SAP Logon [Seite 184]

Configuring the SAP Logon

Use

You can change the following settings in the SAP Logon:

Language

You can display the SAP Logon in the language that you select. To use this option, the SAP Logon language file must be installed by the system administrator.

Message Server Timeout: ____ secs

Specifies how long the SAP Logon waits for a response from the R/3 Message Server. The default value, ten seconds, is normally sufficient, even with slow wide-area network connections.

If you experience repeated timeout connection errors, increase this value. If the error persists, there is probably a network installation problem.

Confirmation of listbox entry delete

Check this box if you want to display a warning before you delete a system or logon group from the SAP Logon.

Disable editing functionality

Check this box if you want to prevent logon entries from being changed. If editing functionality is disabled, you cannot use the options *Properties, Groups, Server, New* and *Delete* in the SAP Logon.

Activate SAPgui trace level

Check this box if you want to define and activate a network trace (SAPGUI trace). For reasons of security and performance, you should only activate the trace options to diagnose the system.

When you choose this option, you can select the trace level that is used. If you select level 2 or 3, an additional log file is generated that records all incoming data in an encrypted binary code.

Additional data hexdump in trace

Check this box if you want to list additional memory areas in the SAPgui trace. This option is only available if you choose trace level 2 or 3, since these trace the data that must be checked against the hexdump when errors occur.

This option can result in both considerable losses in performance and very large trace files.

Additional command line arguments

You can enter any additional command lines arguments in this input field.

The additional information that appears here can help solve particular frontend problems. Only perform the network trace options if the SAP Hotlines requests you to. You should cancel the network trace options as quickly as possible and remove the trace files.

Procedure

1. Display the SAP Logon by choosing *Start* → *SAP Frontend* → *SAPlogon*.
2. Click on the SAP Logon icon in the top left-hand corner of the window and choose *Options*. You can then change the settings that are described above in the *SAP Logon configuration* dialog box.
3. Choose *OK* to return to the initial screen of the SAP Logon.

When you change the SAP Logon language, the system asks you to restart the SAP Logon that is affected by the new language setting. You should then close the SAP Logon and restart it.

Adding a New Logon User Group

Use

Most users only have one group in their selection. You can add additional groups to the SAP Logon.

Procedure

1. Display the SAP Logon by choosing *Start → SAP Frontend → SAPlogon.*
2. Choose *Groups…* to display the *Group Selection* dialog box.
3. In the *System* ID field, enter the system that you want to log on to.

   If a SAP Router is used to connect to the message server, select the SAP router in the *SAP Router for* field.
4. Choose *Generate list*. The system displays the logon groups that are active.
5. Select a logon group and then choose:

- *Logon* to log on without adding the logon group to the list
- *Add* to add the logon group to the list without logging on
- *Add and Logon* to add a logon group to the list and log on immediately.

See also:

Adding an Application Server [Seite 189]

Configuring Logon Groups

Use

You can create and delete logon group entries, remove groups from instances, and delete complete logon groups.

Procedure

To create or change a logon group:

1. Choose C*CMS* → *Configuration* → *Logon Groups.* Alternatively, call Transaction SMLG.
2. The system displays logon group names, instances and their status. To sort this list choose *Group list*→ *Sort* and then choose either *Sort by group* or *Sort by instance*.
3. From here you can:
   - Create entries
   - Sort by instance
   - Delete entries
   - Delete groups
   - Delete group assignments

Maintaining logon groups

1. To display and edit the response time thresholds for each logon group, in Transaction SMLG choose *Group list* → *Format* → *Extended*.
2. If you want to display additional information on a logon group, double-click on an entry in the table or choose F2.
3. In the *Create/Change Entry* dialog box, you can configure the following options:
   - *Group assignment:* The name of the workgroup that is or will be defined. SAP reserves the logon group called "SPACE", so do not use that name.
   - *Instance (application server):* Specifies the name of the instance that supports the class name that was defined above.
   - *Response time:* Once a response time has been defined for one group in an instance, you must define a response time for all groups using the same instance. The system automatically sets a homogeneous value for all groups that are associated with that instance.
   - *Users:* The maximum number of configured users who can be logged on to an instance. The system automatically sets a homogeneous value for all groups that are associated with that instance.

     The values for *Response Time* and *Users* are thresholds rather than absolute limits. Even if the current value for response time or number of users is higher than the threshold value, you can still log on to another instance. The threshold values only influence the calculation of the current logon server of the logon groups.

See also:

Recommendations for Logon Groups [Seite 175]

Questions and Answers: Logon Load Balancing [Seite 177]

Adding an Application Server

Procedure

1. Display the SAP Logon by choosing *Start → SAP Frontend → SAPlogon.*
2. Choose *Server…* to display the *Server Selection* dialog box.
3. In the *System ID* field, enter the system that you want to log on to.

   If a SAP router is used to connect to the message server, select the SAP Router in the *SAP Router for* field.
4. Choose *Generate list* to display the active servers.
5. Select a server and choose:

- *Logon* to log on without adding the server to the list
- *Add* to add the server to the list without logging on
- *Add and Logon* to add a server to the list and log on immediately

See also:

Adding a New Logon User Group [Seite 186]

Adding and Editing Entries Manually [Seite 190] [Seite 190]

Configuring Logon Groups [Seite 187]

Adding and Editing Entries Manually

Procedure

To add a new entry:

1. Display the SAP Logon by choosing *Start* → *SAP Frontend* → *SAPlogon*
2. Choose *New*... to display the *New Entry* dialog box.
3. Enter the following information:

   Description: A short description of the system

   Application Server: The name of the host that you want to connect to

   SAP Router String: A routing entry (for example, `saproute.ini`.)

   SAP System: Specify whether the new SAP System is an R/2 or an R/3 System

   System Number: Specify the system number of the SAP System that you want to connect to

To change an existing entry:

From the SAP Logon, select a system and choose *Properties*.

For server entries, you can change all data. For logon group entries, you cannot change the *Application Server,* the *SAP System* or the *System Number*.

To remove an entry:

From the SAP Logon, select a system and choose *Delete*.

To exit:

To exit the SAP Logon, choose *Close*.

See also:

Configuring Logon Groups [Seite 187]

Recommendations for Logon Groups [Seite 175]

Setting Low Speed Connection

Use

To connect to SAP Systems in a wide area network, you should use a low speed connection. This minimizes data traffic over the network.

Procedure

1. Display the SAP Logon by choosing *Start* → *SAP Frontend* → *SAPlogon*.
2. Choose *Properties* and then *Advanced...*.
3. Flag the *Low Speed Connection* box.

Operation Modes

Purpose

Operation modes let you flexibly adapt R/3 System configuration to varying requirements, maximizing the use of available system resources.

An operation mode defines a resource configuration for the instances in your R/3 System. It can be used to determine which instances are started and stopped, and how the individual services are allocated for each instance in the configuration.

You can define operation modes to suit specific system requirements, for example, to provide additional dialog or background processing resources during a particular period of time without having to restart the R/3 System.

Operation modes define:

- The number of work processes used for each service in the instance
- The times that the services are available

Operation modes support:

- 24 hour uninterrupted system operation
- Automatic switching of work process types

An R/3 instance is **not created** through an operation mode. Instances are created when the R/3 System is installed, and can be assigned to operation modes.

See also:

Example: Day and Night Operation [Seite 214]

Instances [Seite 229]

Consistency Check [Seite 227]

Switching Operation Modes [Seite 194]

Switching Operation Modes Automatically [Seite 198]

Switching Operation Modes Manually [Seite 200]

Checking the Operation Mode with the Control Panel [Seite 225]

Monitoring Servers and Work Processes [Seite 223]

Operation Mode Errors [Seite 209]

Defining Operation Modes [Seite 211]

Operation Mode Switches

Use

When operation modes are switched, the R/3 System work processes are redistributed automatically without stopping and restarting the instances.

Only the work process types are changed. For example, a work process used as a dialog process can be switched for use as a background process.

The new process type is not activated until the process is free. This means that a process may not be switched immediately. Instead, a process is set to be switched when next possible. For example, if all background processes to be switched to dialog processes still have jobs, the processes are switched one by one when the jobs are completed.

Processing is not interrupted. Normal system operation continues uninterrupted during the operation mode switch.

Operation mode switches are recorded in the system log. The old process type and the new process type are recorded for each work process that is switched.

Additional Information on Operation Mode Switches

System Startup

When a server is started, it first runs without an operation mode. The number of work processes and their distribution is defined in the instance profile used to start the server.

The status of the operation mode is checked at regular time intervals and the operation mode defined as active in the timetable is activated automatically. If there is no operation mode for the current time in the timetable, an operation mode is not activated.

Initial Operation Mode Switch

The first time an application server is switched to an operation mode, this is done by the ABAP program `SAPMSSY6`. This program runs cyclically, collects the alert values, performs profile checks and creates DUMMY operation modes. The profile parameter `rdisp/autoabaptime` controls what intervals the program runs at. The default cycle is 300 seconds.

`SAPMSSY6` only switches operation modes if the application server is **not** running in an operation mode. Normally, the system is switched within 5 minutes of system startup to the operation mode defined in the timetable.

`SAPMSSY6` does **not** perform a continuous switch. This means that a manual operation mode switch will not be switched back after 5 minutes.

Timed Operation Mode Switch

The exact times for operation mode switches are defined in the timetable.

A timed operation mode switch is started by the job scheduler (`SAPMSSY2`). The operation mode switch can therefore only be performed automatically if at least one job scheduler is running in the R/3 System. This means that at least one background work process and parameter `rdisp/btctime > 0` must be available.

The switch is always done for all servers defined in the operation mode.

If more than one job scheduler is running, only one job scheduler can activate an operation mode switch.

See also:

Switching Operation Modes [Seite 194]

Switching Operation Modes Automatically [Seite 198]

Normal Operation [Seite 219]

Exception Operation [Seite 221]

Switching Operation Modes Manually [Seite 200]

Switching Operation Modes Automatically

Use

You can:

- Switch operation modes **automatically**.
- Switch operation modes **manually**.

Procedure

There are two ways to switch operation modes automatically:

Normal operation	The time intervals defined for the operation modes are repeated in a 24 hour cycle
Exception operation	The time intervals defined for the operation modes are activated **once** for a specified time period

Only productive operation modes can be switched automatically. Test operation modes cannot be entered in the timetable.

See also:

Normal Operation [Seite 219]

Exception Operation [Seite 221]

Switching Operation Modes Manually

Use

You can switch an operation mode manually at any time.

However, you must always ensure that a manual operation switch does not disrupt system operation, for example, by providing too few dialog processes.

Before you switch the operation mode in the system manually you can carry out a simulation of this switch [Seite 198].

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Control panel.* Alternatively, call Transaction RZ03.

   The system displays information on operation modes and server statuses.

2. Choose *Choose operation mode*.

   The system displays a list of the defined operation modes.

3. Choose the operation mode that you want to switch to.

   You return to the list of operation modes and server statuses.

4. To switch the operation mode on all servers, choose *Control* → *Switch operation mode* → *All servers*.

   The servers remain in the manually activated operation mode until the next switch time.

   You can also switch the operation mode on individual servers. However, you should only do this in exceptional cases, for example, if automatic switching has not worked correctly.

The system displays an error message if the operation mode switch cannot be performed on all of the chosen servers. The test log for operation mode switching (*Log*) helps you decide whether or not to go ahead with the switch.

See also:

Platform Specificity [Seite 65]

Switching Operation Modes Automatically [Seite 198]

Simulating Operation Mode Switching

Use

Before you switch an operation mode in the system, you can simulate this switch.

Procedure

To start the simulation from the R/3 main menu, choose *Tools* → *CCMS* → *Control/Monitoring* → *Control panel* → *Control* → *Switch operation mode* → *Simulation*.

The test log describes the possible switches and the errors that may occur, which you can then avoid for the “real” operation mode switch.

Also see:

Operation Mode Switches [Seite 194]

Rules for Work Process Distribution

Definition

When you set work process distribution, observe the following points:

Number of work processes

- The total number of work processes cannot be changed because it is defined in the instance profile. When the system is started, the defined number of processes is generated at operating system level, and cannot therefore be changed during an operation mode switch.

Dialog processes

- You cannot change the number of dialog processes directly. The number of dialog processes is calculated automatically from the total number of work processes minus all the other work processes.
- There must always be at least two dialog processes. The number of other work process types can only be increased if there are more than two dialog processes.

Background processes

- The number of background processes can be changed as required.
- The number of background processes can be set to 0 for each server, but there must be at least 1 background process available in the system.

  Class A jobs

  The number of work processes reserved for job class A is a subset of the number of background processes.

  You should only reserve work processes for job class A if it makes sense within your system organization. Work processes reserved for class A jobs are no longer available for job classes B or C. For more information, see Prioritization Strategies: Keeping Work Processes Open for Class A Jobs [Seite 116].

Update processes

- The number of update and V2 update processes can be increased as required, but cannot be reduced to 0. If no update process were available, the update queue could not be processed during an operation mode switch.

  If an instance is running without an update process, you **cannot** increase the number to 1 during an operation mode switch.

Enqueue processes

- The number of enqueue processes can only be changed within certain limitations. An instance is defined as an enqueue server in the instance profile. You can only change the number of enqueue processes as follows:
  - 1 to n
  - n to 1

- You can increase the number of enqueue processes to more than 1. However, most of the time there is no advantage in doing so. You should only increase the number of enqueue processes to more than 1 after consulting EarlyWatch or your SAP consultant.

Spool processes

- The number of spool processes cannot be changed.

Setting the Work Process Distribution

Use

When you create a new instance definition, you must set the work process distribution and assign the work process distribution to an operation mode.

Procedure

To change the work process distribution:

1. Choose *CCMS* → *Configuration* → *Operation mode/instances.* Alternatively, call Transaction RZ04.
2. Choose *Instances/operation modes.* In the list of productive instances, position the cursor on an operation mode name and choose *Instance* → *Maintain instance* → *WP distribution*. The system displays the box *CCMS: Maintain Work Process Distribution* dialog box.
3. To change the work process distribution, position the cursor on the number of work processes, then choose '+' or '-'. The work processes are redistributed automatically.

   You can only change the assignment of work processes to services in the *Number of work processes* group box. You cannot change the total number of work processes, or the start or system profiles.
4. Save your changes. The changes will take effect when the operation mode is next activated.

To assign the work process distribution to another operation mode:

1. Choose → *CCMS* → *Configuration* → *Operation mode/instances.* Alternatively, call Transaction RZ04.
2. Choose *Instances/operation modes.* In the list of productive instances, position the cursor on an operation mode name and choose *Instance* → *Maintain instance* → *WP distribution*. The system displays the box *CCMS: Maintain Work Process Distribution* dialog box.
3. Choose *Other operation mode* and enter the name of an operation mode in the *Operation mode* field. Repeat this for all the operation modes whose work process distribution you want to change.
4. Change the work process distribution for the new operation mode as required.
5. Save your changes.The changes will take effect when the operation mode is next activated.

   To set the same work process distribution for all operation modes, enter '*' for all operation modes. This definition does not depend on the operating system.

See also:

Rules for Work Process Distribution [Seite 204]

Operation Mode Errors

Purpose

If an operation mode switch fails, the most frequent cause is inconsistency in the instance definition. Two examples of this are:

- A different number of profiles or a different total number of work processes are defined for day operation and for night operation.
- A server was started with a different configuration than the configuration described in the instance definition.

If these problems occur, the Control Panel displays the message *Operation mode non-standard* or *Work processes non-standard*.

Process Flow

To solve this problem:

- Make the necessary changes to the instance definition

or

- Copy the current instance status to the instance definition.

Other Servers Cannot Be Started Remotely (UNIX systems only)

If you cannot start servers from the control panel, most likely the remote command `rexec` could not be executed. This is, in turn, often because the `.netrc` file required for a remote logon on a UNIX server has not been correctly maintained.

See also:

Prerequisites for Starting Remote Instances [Seite 236]

Defining Operation Modes

Use

You can modify an existing operation mode or define a new operation mode.

Procedure

1. Choose *CCMS* → *Configuration* → *OP modes/instances*. Alternatively, call Transaction RZ04.

   The system displays a list of productive instances.

   If you have not yet defined any operation modes, the test operation mode DUMMY will be displayed. This operation mode is configured automatically so that system functions such as the Control Panel and background job scheduling can be used.

   If a DUMMY operation mode is already available, it is updated whenever a server is booted. In this way, it is adjusted to the changed system conditions.

   The operation mode DUMMY cannot be used for operation mode switching.

2. Create a new operation mode (that can be switched automatically)

   To do this, choose *Operation mode* → *Create*.

3. Enter data in the following fields:

Operation mode:	Enter a name for the operation mode, for example `DAY_OPERATION`, and a short description. You can enter any name for the operation mode, but it is better to choose a name that is meaningful and that makes the task of the operation mode clear.
Short description	Type a short description of the operation mode, for example, “Day operation with 6 dialog processes.”
Monitoring properties variant	Enter the name of the monitoring properties variant [Seite 903] in the monitoring architecture that is assigned to the operation mode. When the operation mode is started, the settings that the monitoring properties variant contains are activated automatically in the alert monitor.

4. Save the data by choosing *Operation mode* → *Save*.
5. Repeat this procedure for all other operation modes that you require.

   When you have defined and saved your operation modes, you can define the instances and servers.

   You can also call the operation mode maintenance tool from the control panel.

See also:

Defining Operation Modes

Instances [Seite 229]

Requirements for Maintaining Operation Modes [Seite 217]

Switching Operation Modes Automatically [Seite 198]

Switching Operation Modes [Seite 194]

Setting Up the CCMS [Seite 26]

Using the Alert Monitor [Seite 812]

Defining Day and Night Operation

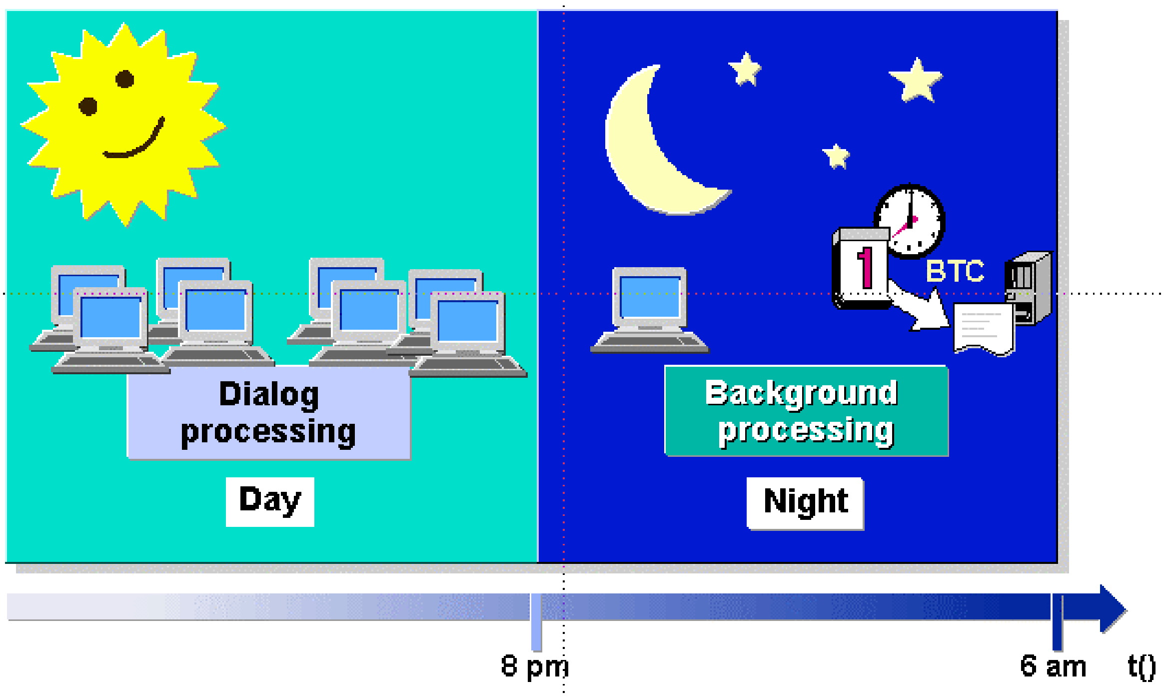
Use

You can define separate operation modes for day and night operation. This means you can guarantee response times for important data entry transactions during the day, and use more work processes at night for job processing.

You are most likely to use the R/3 System for dialog processing between 6:00 and 20:00. Outside that time, system resources are normally required primarily for non-dialog processing (background jobs).

You can reconfigure your R/3 System dynamically at set times by switching operation modes and thus avoid the disadvantages of a system restart.

The diagram below illustrates the effect of an operation mode switch:



Procedure

You can display the status of the operation modes and instances using the control panel. To do this, choose *CCMS* → *Control/Monitoring* → *Control panel.*

Assume that your R/3 System has a total of ten work processes available. Define an operation mode [Seite 211] for night operation to allow more efficient data processing at night.

In the timetable for automatic operation mode switching, enter the times for the two operation modes:

- 6 a.m. to 8 p.m. for day operation
- 8 p.m. to 6 a.m. for night operation

Table: Operation mode switching: Day/night operation

Work process distribution	**Day operation 6 a.m. to 8 p.m.**	**Night operation 8 p.m. to 6 a.m.**
Dialog	5	2
Background	1	4
Update	2	2
Enqueue	1	1
Spool	1	1

Most R/3 resources will then be available for dialog processing during the day. At 8 p.m. the operation mode will be switched automatically, making more system resources available for background jobs and data transfer from other systems. At 6 a.m. the next day, the day operation mode will automatically be activated, making more system resources available for dialog processing.

See also:

Instances [Seite 229]

Scheduling an Operation Mode Switch [Seite 198]

Switching Operation Modes [Seite 194]

Requirements for Maintaining Operation Modes

Definition

Before you can maintain operation modes and instances in the CCMS, the SAP System must be installed correctly using the R/3 installation program `R3INST`. The installation procedure for distributed instances ensures that the following actions are carried out:

- The shared file systems on the SAP System are automatically mounted on the host system of the distributed instance.
- The instance directories are created.
- The R/3 users are defined in the host system.
- The operating system kernel for the instance is generated in accordance with the host system attributes.
- Entries in host system configuration files (for example, `hosts` and `services`) are maintained.
- Start and system profiles for the instance are generated.

You can use the program `sapstart` and the installed profiles to start the R/3 System.

When the SAP System has started, you can begin configuration.

See also:

Setting Up the CCMS [Seite 26]

Profile Files [Seite 253]

Defining Operation Modes [Seite 211]

Defining Normal Operation

Use

For normal operation, you must maintain the operation mode timetable.

Procedure

1. Choose *CCMS* → *Configuration* → *Operation mode timetable.* Alternatively, call Transaction SM63.
2. Choose *Normal operation (24 hr),* and *Display.*

   The system displays the timetable for the 24 hour cycle.
3. To assign other times to the operation mode, choose *Normal operation (24 hr)* and *Change*.
4. Double-click on the start time and end time that you require.
5. Choose *Assign* to assign these times to the operation mode.
6. Enter the name of the operation mode to be active for the time period that you defined.

   You can only select productive operation modes.
7. Save the changes.
8. Finally, save the table.

You must always define a full 24 hour cycle, that is, the operation modes must be set up for 24 hour system operation. If the timetable is **not** defined:

- The start configuration according to the profile will remain active
- No unattended automatic operation mode switch is possible
- The job scheduling monitor will not function properly
- When a job is created or released with start date/time, it will not be possible to check whether background processes are available at the start time
- Jobs that are scheduled to start when the operation mode is switched are **not** started.

See also:

Exception Operation [Seite 221]

Switching Operation Modes [Seite 194]

Defining Exception Operation

Procedure

1. Choose *CCMS* → *Configuration* → *Operation mode timetable.*

   Alternatively, call Transaction SM63.

2. Choose *Exception operation,* and *Display.*

   The system displays the timetable for the 24 hour cycle.

3. Go back and then choose *Exception operation*, and *Change*.

4. You can now change the date on which the operation mode is to be activated. Choose *Day+1*, *Day-1*, *Specify a day* accordingly.

5. Choose *Assign* to set the time at which the operation mode is to be activated.

   You do not need to define a full 24 hour cycle for exception operation. You must only define the period in which you want the operation mode to be active.

6. Save the changes.

See also:

Normal Operation [Seite 219]

Switching Operation Modes [Seite 194]

Monitoring Servers and Work Processes

Use

When an operation mode has been switched you can see in the process overview whether the work process types have been changed correctly.

Procedure

To monitor the work processes on your current application server:

From the R/3 initial screen, choose *Tools* → *Administration* → *Monitor* → *System monitoring* → *Process overview*. Alternatively, call Transaction SM50.

The system displays an overview of the work processes for the server you have logged onto.

The information for each work process is displayed in a line. The *Ty.* column shows the work process type: for example, DIA is a dialog process, BTC is a background process.

To monitor the work processes on a different application server:

From the R/3 initial screen, choose *Tools* → *Administration* → *Monitor* → *System monitoring* → *Servers*. Alternatively, call Transaction SM51.

The system displays a list of the R/3 servers. To display the processes for a particular server, position the cursor on the line containing the server name, then choose *Processes*.

See also:

Work Process Load Monitor [Seite 1042]

Checking the Operation Mode with the Control Panel

Use

The primary function of the control panel is to. start or stop instances [Seite 69]. However, you can also use it to check whether instances were started correctly, and whether instances are running in the correct operation mode, that is, whether automatic operation mode switching is possible.

Procedures

To display an overview of operation modes and instances:

1. Start the control panel [Seite 958] by choosing *CCMS* → *Control/monitoring* →.*Control panel*.

   The system displays a list of the host systems and the instances in your R/3 System. The list contains:

   – Instance names

   – Status

   – Services running on your application server

   – Active operation mode

2. Select a server.

To find out the cause of a non-standard operation mode:

1. Choose *Monitoring* → *Status details*.

   The system displays a dialog box showing the details of the server.

   The most frequent reason for an operation mode switch failing is inconsistencies in the total number of work processes or in the profiles.

2. When you have rectified the problem, you can switching the operation mode manually [Seite 200] from the control panel.

See also:

Starting and Stopping the Database System: UNIX [Seite 50]

Controlling System Startup [Seite 61]

Consistency Check

Use

You can run an error check of the R/3 profile definitions to avoid any system problems. This consistency check determines whether changes made to profiles are valid, and recognizes any settings that will prevent the R/3 System from starting or operating properly.

Prerequisite

Consistency checks are only possible for profiles that were maintained using the R/3 profile maintenance tool. You cannot perform a consistency check if you have maintained a profile in any other way, for example with an operating system editor.

See also:

Switching Operation Modes Using Start and Instance Profiles [Seite 273]

Procedure

From the list of productive instances, choose *Instance* → *Consistency check*.

All the start and instance profiles for the defined instances are checked. The profiles are compared with the current work process distribution. The check determines, for example, whether the enqueue processes are started on more than one server, or whether the message server has been properly configured.

See also:

Rules for Work Process Distribution [Seite 204]

Instances

Definition

An R/3 instance is defined when the R/3 System is installed.

In the R/3 System, the term "instance" is often used as a synonym for "server" or "application server". This depends on the chosen installation environment. It is possible to install several instances on a single application server.

An R/3 instance defines a group of resources such as memory, work processes and so on, usually in support of a single application server or database server within an R/3 client/server environment. Application servers share a common set of buffers, and are controlled by the same dispatcher process.

An instance is the mechanism to which users connect to work in the R/3 system, and is managed by the CCMS as a single entity.

An R/3 System can consist of one or more instances. For example, an R/3 System with a single instance containing only a central server, or a client/server system containing two or more separate instances.

For each R/3 instance:

- Separate directories are defined on the UNIX, AS/400, or Windows NT server on which the instance is to run
- Shared file systems can be used
- Entries in operating system configuration files are created (`/etc/services`, `/etc/sapconfig`...)
- Communication entries in the host are created
- Start and system profiles are created
- R/3 operating system users are installed

An R/3 instance is **not** assigned automatically to an operation mode.

See also:

Creating an Instance Definition for All Servers [Seite 234]

Prerequisites for Starting Remote Instances [Seite 236]

Creating an Instance Definition for One Server [Seite 232]

Consistency Check [Seite 227]

Starting and Stopping Instances [Seite 69]

Setting Up the CCMS [Seite 26]

Monitoring Servers and Work Processes [Seite 223]

Creating an Instance Definition for One Server

Use

If you only want to create a new instance definition for one application server, it is useful to copy the instance definition for the server. Assigning an instance definition to an operation mode makes it part of the operation mode. As a result, you can switch the instance, see if it is inactive and start and stop it using the control panel [Seite 956] (Transaction RZ03).

Prerequisites

You must define an operation mode [Seite 211] before you can assign an instance definition to it.

Procedure

1. Choose *CCMS* → *Configuration* → *Operation modes/instances.* Alternatively, call Transaction RZ04.
2. Choose *Instances/operation modes* and *then Instance* → *Create new instance*. The system displays the *CCMS: Maintain Instance Data* screen.
3. Enter the host name and choose *Current settings*. If the application server is already running, the system will display the current settings for that instance. However, if the application server is not yet running, you should use the input help to display the possible entries and fill in the following fields:

Host name	Enter the name of the host running the instance.
SAP System Number	Enter the SAP System number that was specified when your SAP System was installed.
Start Profile: Profile Name	The name of the start profile from Profiles [Seite 240] to start the instance with. Choose the input help to display a list of available start profiles. You can choose a profile from the list.
Instance Profile: Profile Name	The name of the instance profile from Profiles [Seite 240] to run the instance with. Choose the input help to display a list of available instance profiles. You can choose a profile from the list.

4. Choose *Save.* The system checks your configuration automatically when you save and informs you of any inconsistencies. Choose *Continue.* The result of the check is displayed in the *Display Check Log of Profile* window.

   UNIX systems only: So that you can start and stop instances from the control panel, you must enable a remote logon on the target UNIX server. See Prerequisites for Starting Remote Instances [Seite 236]

5. In the *CCMS: Maintain distribution of work processes* dialog box, assign the work processes to at least one operation mode.

In the *Operation mode* field, enter the name of the operation mode or select an operation mode from the list of possible entries. To set the same work process distribution for all operation modes, enter `*`.

The system proposes the current work process distribution in the instance profile. However you can change the work process distribution (see Setting the Work Process Distribution [Seite 207]).

6. Save your changes. To assign the work process distribution to additional operation modes, choose *Yes* when the system prompts you. Choose *No* if you do not want to do this.
7. When you have finished, you can perform a consistency check [Seite 227]. Save your changes. All the instance definitions for productive operation modes are then saved in the database.

   To change the assignment of work processes to the instance, see Setting the Work Process Distribution [Seite 207]

See also:

Creating an Instance Definition for All Servers [Seite 234]

Maintaining Instance Definitions [Seite 238]

Creating an Instance Definition for All Servers

Use

You can generate the current instance definition for all the active servers. This procedure is helpful if you want to generate all instance descriptions for the first time.

Procedure

1. Choose *CCMS* → *Configuration* → *Operation modes/instances* → *Instances/operation modes*. Alternatively, call Transaction RZ04.
2. The system displays the *CCMS: Maintain Operation Modes and Instances* screen. Choose *Settings* → *Based on act. status* → *New instances* → *Generate*. The system now queries the actual status of every application server that is active, and an instance definition is created.

   For every configured operation mode, the system creates a work process distribution in accordance with the distribution of the respective instance profile.
3. Save the new instance.
4. Now define the work process distribution for switching operation modes [Seite 194]

   If you want to start and stop servers from the control panel [Seite 225], you should maintain the instance startup user.

   See also: Prerequisites for Starting Remote Instances [Seite 236]
5. When you are finished, you can perform a consistency check [Seite 227].

   It only makes sense to do this if you are maintaining profiles using the CCMS profile maintenance tool.

   See also: Switching Operation Modes Using Start and Instance Profiles [Seite 273]
6. Save the changes. All the instance definitions for productive operation modes are then saved in the database.

See also:

Creating an Instance Definition for One Server [Seite 232]

Maintaining Instance Definitions [Seite 238]

Prerequisites for Starting Remote Instances

Starting an R/3 Instance: Remote UNIX Server

To start an R/3 instance on a remote UNIX server, the CCMS must log onto the remote UNIX server as the UNIX administrative user for R/3. To enable this remote logon, you must maintain the `.netrc` file on the target UNIX system, the one on which an R/3 instance is to be started or stopped.

The `.netrc` file contains the user and password for the remote logon. The file must therefore be protected with read permission only for the R/3 administrator in UNIX, no permissions for any other user. It should reside in the home directory of the administrative user for R/3 on the target host.

See the documentation of your UNIX systems for exact information on maintaining `.netrc`.

Prior to Release 4.5A, you could use Transaction RZ04 (*Tools* → *CCMS* → *Configuration* → *Operation modes/instances*) to store a UNIX user and password for use in starting R/3 instances on remote servers. The remote start-up capability (and the user and password) were used in the CCMS control panel (Transaction RZ03) and in the CCMS system monitor (Transaction RZ02), for example.

The R/3 System needs a stored user and password only for starting R/3 instances at remote UNIX servers. Remote start-up of instances at AS/400 and Windows NT servers uses other mechanisms and requires no user and password.

It is no longer possible to store a user and password in R/3. You should therefore maintain the UNIX files listed above to enable remote startup and shutdown of R/3 instances.

Further, whenever Transaction RZ04 is started, the system checks whether a user and password have been stored. If so, the user (the system administrator) is warned about the problem. If the user so wishes, the user and password are deleted automatically by the system.

Starting an R/3 Instance: Remote Windows NT Server

An R/3 instance on a Windows NT host is started with a message to the SAP Service Manager on the remote Windows NT. No remote user is required. The Service Manager is set up for the communication that is necessary for remote operations during the installation of R/3 or of an R/3 instance. No further maintenance should be necessary to enable remote start-up and shutdown from the CCMS control panel or system monitor.

Starting an R/3 Instance: Remote AS/400 Server

An R/3 instance on an AS/400 host is started through an AS/400 mechanism that is transparent to the system administrator. This mechanism is set up for the communication that is necessary for remote operations during the installation of R/3 or of an R/3 instance. No further maintenance should be necessary to enable remote start-up and shutdown from the CCMS control panel or system monitor.

Maintaining Instance Definitions

Use

Once you have defined the operation modes, you must maintain the instance definition.

It is also necessary to maintain instance definitions if, for example, server names or profiles were changed, or if new servers have been added.

Procedures

To change instance definitions:

1. Choose *CCMS* → *Configuration* → *Operation modes/instances.* Alternatively, call Transaction RZ04.
2. Choose *Instances/operation mode.* Select an entry from the list of productive instances and choose *Choose*.
3. You can then change the data for that instance.

To delete a server definition:

1. From the list of productive instances, position the cursor on a server.
2. Choose *Instance* → *Delete entry*. All data for that server is then deleted.

To delete an assignment to a work process:

1. From the list of productive instances, position the cursor on the line with the operation mode or work process.
2. Choose *Instance* → *Delete entry*. Only the assignment to that work process is deleted.

To delete an operation mode:

1. From the list of productive operation modes, position the cursor on a line with the operation mode.
2. Choose *Operation mode* → *Delete*. The operation mode, and all the instances and work process definitions assigned to it, are deleted.

The automatically configured DUMMY operation mode is no longer needed once you have defined your operation modes. Delete the DUMMY operation mode as described above.

See also:

Creating an Instance Definition for All Servers [Seite 234]

Creating an Instance Definition for One Server [Seite 232]

Profiles

Purpose

SAP profiles are operating system files that contain instance setup information. SAP Systems can consist of one or more instances. Individual setup parameters can be customized to the requirements of each instance. These individual parameters allow you to configure:

- The runtime environment of the instance (resources such as main memory size, shared memory, roll size)
- Which services the instance has available (work processes)
- Where other services can be found (database host).

Features

This profile file is structured as follows:

```
# This is a comment in a SAP profile:
Parametername1 = Value1
Parametername2 = Value2
```

Parameter names that logically belong together have a common root. For example, the root of parameters that control the dispatcher within an application server is: `rdisp/`.

The parameter `rdisp/wp_non_dia` specifies how many dialog work processes are started by the dispatcher.

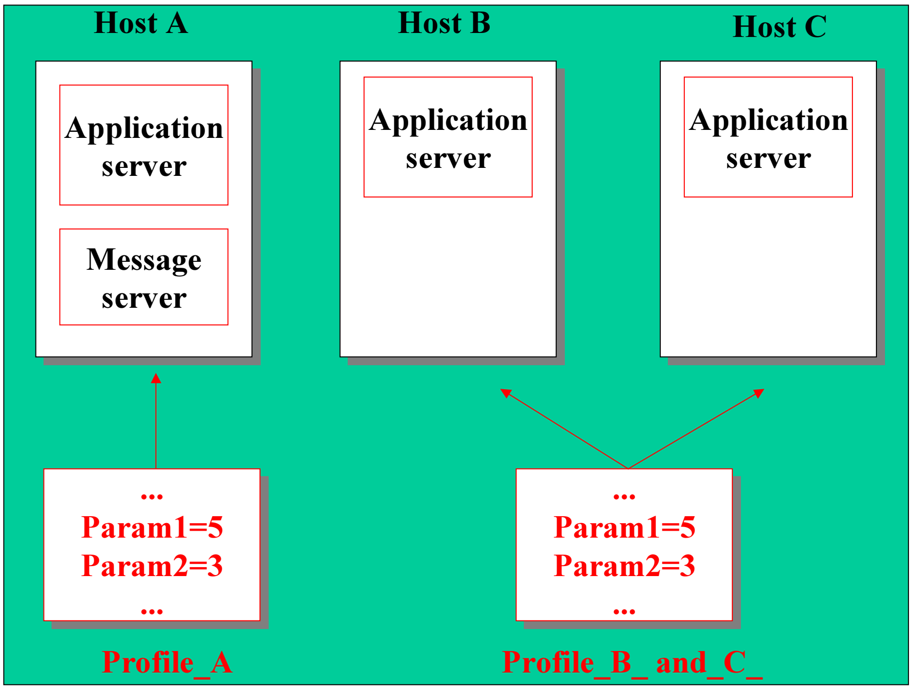
SAP profiles are stored in a special file directory. This directory can be made accessible from all hosts depending on current needs:

UNIX systems: `/usr/sap/<SID>/SYS/profile`

Windows NT systems: `\<SAPGLOBALHOST>\sapmnt\<SID>\sys\profile\`

(`<SID>` = SAP system name and `<SAPGLOBALHOST>` = name of the NT machine on which the global profile directory is physically located)

All hosts in an SAP System can access these profiles. It is possible for several R/3 instances to use a single profile simultaneously. Separate profiles are not required for each R/3 instance.



You should maintain setup profiles using the Computing Center Management System (CCMS). You should therefore not edit the active profiles directly at operating system level.

See also:

[Substitute Variables in Profile Values [Seite 242]]

[Profile Maintenances [Seite 278]]

[Checking, Saving and Activating Profiles [Seite 287]]

[Start Profiles [Seite 255]]

[Default Profiles [Seite 261]]

[Instance Profiles [Seite 266]]

Substituting Variables in Profile Values

Process Flow

Parameter values in instance profiles [Seite 266] can contain the following variables:

`$(parameter name)` is replaced by the value of the parameter name specified in brackets.

```
Param1 = '/usr/sap/C11/D53/dbg'
Param2 = $(Param1)/stats
```

Therefore `Param2` = `/usr/sap/C11/D53/dbg/stats`

$$ is replaced by the SAP System number.

```
Param3 = logfile$$.
SAPSYSTEM = 29
```

Therefore `Param3` = `logfile29`

You can also define **local substitute variables**. These values are not used by the SAP programs, their importance lies within a profile. They contain values used for setting up or filling parameter values. The names of these local substitute variables begin with an underscore ('_').

```
_SAP_PROFILE_DEFAULT = /usr/sap/C11/SYS/profile/DEFAULT.PFL
Param4 = _SAP_PROFILE_DEFAULT. Therefore
Param4 = /usr/sap/C11/SYS/profile/DEFAULT.PFL
```

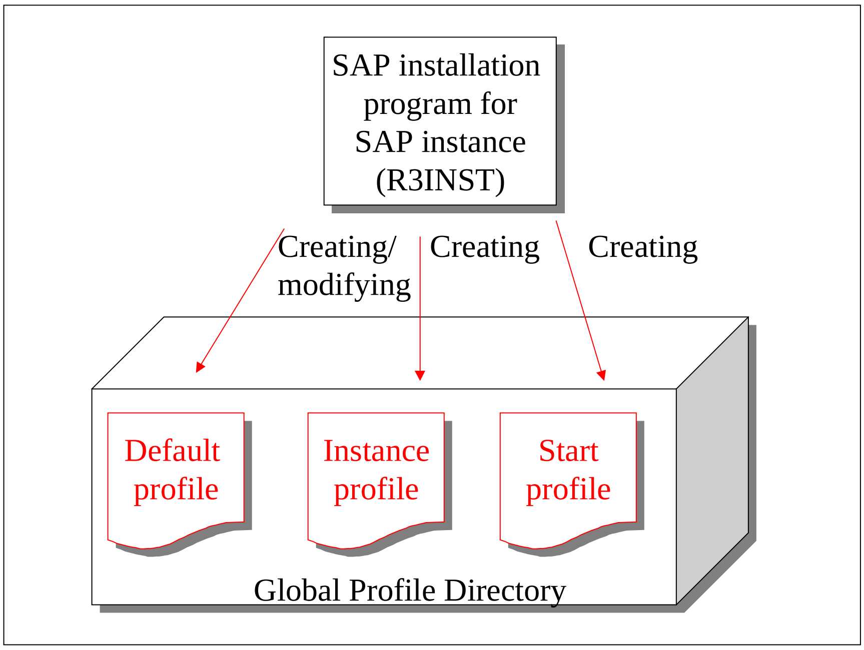
Where Do Profiles Come From?

Definition

When an SAP instance is installed on a host using the SAP installation program **R3INST**, a start profile and an instance profile are automatically generated. If it is the first instance of an SAP System, the system also creates a default profile. Otherwise the existing default profile is simply updated.

The SAP installation program ascertains various profile settings. For example, appropriate buffer sizes, important data directories, and the name of the database host.

The profiles are placed in a global file directory so as to have access from every client in the system.



SAP start and stop utilities are also generated during the installation process. Your new installation may not require these utilities, depending on whether two or more instances share the same profile. If so, these redundant utilities can be removed. However, before you delete the utilities, complete the entire installation process and verify its operability.

In R/3 Releases 2.1x / 2.2x, the profiles were generated with the names: **START_<instancename>** and **<SID>_<instancename>**. From Release 3.0A, the profiles are generated with the names: **START_<instancename>_<hostname>** and **<SID>_<instancename>_<hostname>**. The '**startsap**' script first looks for profiles that end with **_<hostname>**. If these profiles do not exist, the names from the Releases 2.1x / 2.2x are used.

See also:

Profile Maintenance [Seite 278]

What to do After Installing an SAP Instance [Seite 274]

Reference Server of a Profile

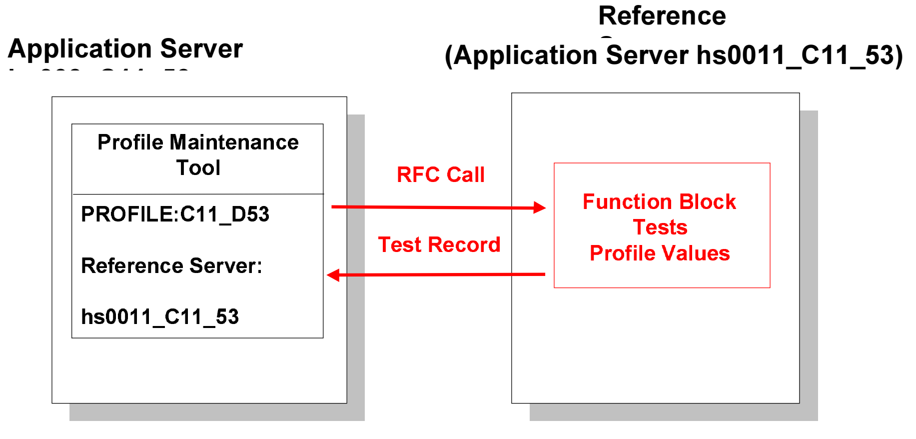
Definition

A “reference server” (= SAP application server) is assigned to each profile by the CCMS profile maintenance tool. The profile is tested on this reference server. There are several reasons for this:

Many profile parameters are platform-dependent, for example the names of file directories. In heterogeneous operating system environments, it is important that the names of file directories are tested on the operating system platform on which a particular application server is going to run.

Many profile parameters are dependent on the resources allocated to their computer, such as the number and size of shared memory segments made available on the computer. Therefore, SAP shared memory pools inside the instance profiles must be adjusted for SAP pools on the host on which a particular application server should run.

The test is performed using an RFC module on the relevant application server:



The assignment of a reference server to a profile does **not** mean that you have to create a separate profile for every R/3 instance. You can use a profile representing several R/3 instances simultaneously. In this case, simply ensure that a particular profile is only assigned to R/3 instances with the same type of operating system.

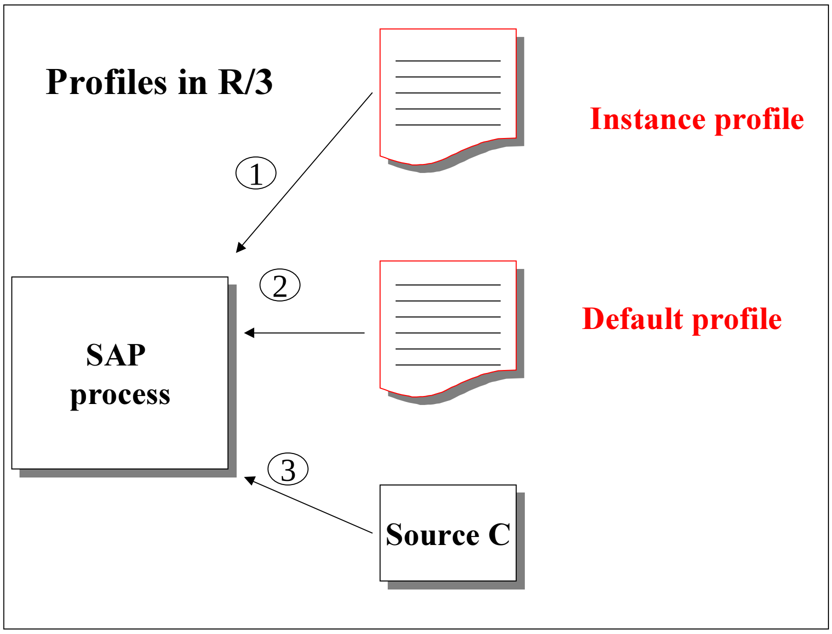
See also:

Maintaining Profiles [Seite 278]

Profile Parameters

Definition

Parameters can be in default profiles [Seite 261] or instance profiles [Seite 266]. Parameter values are configured as shown below:



When an SAP startup process wants to read a profile parameter, it checks:

- Whether the parameter is contained in the instance profile
- The default profile, if the parameter cannot be found in the instance profile.

If neither profile contains the parameter, the default value is taken out of the startup program coding.

If possible, ensure that a particular parameter appears only in the default or instance profile.

Important Profile Parameters

Definition

The recommended SAP profile parameter settings are listed below for the various application modules being used on any SAP instance. For large SAP client/server installations containing many application instances, the settings for one application instance may vary significantly compared to another. This is expected as the various application work groups will most likely be segregated from one application instance/server to another. There are three different categories of parameter settings depending on the mix of application modules being utilized:

Category I	Category II	Category III
FI and GL	CO, FI and GL	CO, AM and HR
CO	AM, FI and GL	CO, AM, HR, FI and GL
AM	HR, FI and GL	SD and other applications
HR	HR and AM	MM and other applications
	CO and AM	
	SD	
	MM	

The profile parameter settings for each category are specified below for both basic and extended maintenance. For basic maintenance, enter the parameter name in the right screen. For extended maintenance, enter the actual parameter name in the profile. The following table labels parameters which cannot be set using basic maintenance as N/A (not applicable).

For system profile parameters, you should use the standard values proposed by the system. The preset values are valid for almost all cases. You should only change the parameter values with the prior agreement of SAP or a SAP partner.

Technical information for a parameter is provided by the system. Choose F1 to display documentation for the parameters that appear in the current profile.

Dictionary Buffers

Basic maintenance

Buffer name	Field	I	II	III
Nametab	`Nametab`	6.000	8.000	10.000

Extended maintenance

Buffer name	Parameter name	I	II	III

Important Profile Parameters

Table descr.	`rsdb/ntab/entrycount`	4.096	6.144	7.168
Field catalog	`rsdb/ntab/ftabsize`	2.000	3.000	3.500
Short nametab	`rsdb/ntab/sntabsize`	1.000	1.000	1.500
Initial record	`rsdb/ntab/irbdsize`	1.500	2.000	2.500

Program and CUA Buffers

Basic maintenance

Buffer name	Dynpro field	I	II	III
Program	ABAP/4 Programs	35.000	50.000	70.000
CUA	CUA Information	2.200	2.200	2.200
Dynpro	Dynpros	2.200	2.500	3.000
Directory Entries Dynpro	N.A.	N.A.	N.A.	N.A.

Extended maintenance

Buffer name	Parameter name	I	II	III
Program	`abap/buffersize`	35.000	50.000	70.000
CUA	`rsdb/cua/buffersize`	2.200	2.200	2.200
Dynpro	`zcsa/presentation_buffer_area`	2.200.000	2.500.000	3.000.000
Directory Entries Dynpro	`sap/bufdir_entries`	2.200	2.500	3.000

Table Buffers

Basic maintenance

Buffer name	Dynpro field	I	II	III
Generic Key	Generic key tables	7.000	9.000	11.000
Directory Entries Generic Key	N.A.	N.A.	N.A.	N.A.
Single Key	Single key tables	5.000	5.000	7.000
Directory Entries Single Key	N.A.	N.A.	N.A.	N.A.

Extended maintenance

Buffer name	Parameter name	I	II	III
Generic Key	zcsa/table_buffer_area	7.000.000	9.000.000	11.000.000
Directory Entries Generic Key	zcsa/db_max_buftab	1.200	1.500	2.000
Single Key	rtbb/buffer_length	5.000	5.000	7.000
Directory Entries Single Key	rtbb/max_tables	100	100	100

Roll and Paging Files

Basic maintenance

N.A.

Extended maintenance

File	Parameter name	Up to 30 users	As from 30 users
Paging	rdisp/PG_MAXFS	8.000	16.000
Roll	rdisp/ROLL_MAXFS	4.000	8.000

Roll and Paging Areas in the Shared Memory

Area	Parameter name	Value
Roll	rdisp/ROLL_SHM	1000 or 0
Paging	rdisp/PG_SHM	1000 or 0

See also:

Buffer Monitor [Seite 1066]

Profile Files

Definition

During R/3 installation, three profiles are created in the file system:

- `DEFAULT.PFL`
- `START_<Instance>`
- `<SID>_<Instance>`

These profiles contain the parameters that determine the operating system resource usage of the R/3 instance, for example, the number of work processes, the buffer sizes.

You import and activate these profiles using the R/3 profile maintenance tool.

After that, you should simply maintain and activate the profiles using the profile maintenance [Seite 278].

The profile values can only be checked if the profiles have been maintained using the profile maintenance tool.

See also:

Profiles [Seite 240]

Start Profiles

Definition

When you start an SAP instance on a host, the start profile defines which SAP services are started (message server, dialog, gateway or enqueue process. for example). The `startsap` program is responsible for starting these service processes, and it uses a start profile to begin the startup process.

The processes that can be started include:

- Application server
- Message server
- SNA Gateway
- System log send demon
- System log receive demon

Apart from the general profile parameters, such as the name of the SAP System (`SAPSYSTEMNAME`), instance number (`SAPSYSTEM`) and name of the SAP instance (`INSTANCE_NAME`), the only parameter names that are permitted in a start profile are:

- `Execute_xx (xx = 00-99)`: To start operating system commands, which prepare the SAP System start. For example, you can use this parameter to start the SAP-related database or to set up links to executables on UNIX platforms.
- `Start_Program_xx (xx = 00-99)`: To start an SAP instance, for example, on an application server.
- `Stop_Program_xx (xx = 00-99)`: To start an operating system command or SAP program after the SAP instance was stopped. For example, the halting or removal of shared memory areas that were used by the SAP System.

The number `xx` defines the execution sequence. The programs specified in `Execute_ parameters` are executed before the programs listed in the `Start_Program` parameters. After the SAP instance has been stopped, the programs specified in the `Stop_Program` parameters are started. Here is an example of a start profile used to start a message server, an application server and an SNA gateway:

```
#.*****************************************************************
#.*       Start profile START_DVEBMG47
#.*
#.*       Version                                = 000003
#.*       last changed by                    = WATT
#.*       last changed on   = 21.03.1995, 15:05:19
#.*****************************************************************
SAPSYSTEMNAME = K11
```

```
INSTANCE_NAME = DVEBMG47

#-----------------------------------------------------------------

# start message server

#-----------------------------------------------------------------


_MS = ms.sapK11_DVEBMG47

Execute_01 = local ln -s -f $(DIR_EXECUTABLE)/msg_server $(_MS)

Start_Program_01 = local $(_MS) pf=$(DIR_PROFILE)/K11_DVEBMG47


#-----------------------------------------------------------------

# start application server

#-----------------------------------------------------------------


_DW = dw.sapK11_DVEBMG47

Execute_02 = local ln -s -f $(DIR_EXECUTABLE)/disp+work $(_DW)

Start_Program_02 = local $(_DW) pf=$(DIR_PROFILE)/K11_DVEBMG47


#-----------------------------------------------------------------

# start SNA-Gateway

#-----------------------------------------------------------------
```

Use

To run a program on the local host, place the word 'local' in front of the relevant parameter value:

```
Execute_00 =  local sapmsesa 53 remove
```

To execute a program on a remote host, place the host name in front of the parameter value.

```
Execute_00 = hs0011 sapmsesa 53 remove
```

You can choose any name for a start profile. The start profile files generated by SAP are structured as follows: START_<instance name> or START_<instancename>_<hostname>.

```
START_DVEBMGS53, START_DVEBMGS53_hs0311
```

To start the same SAP service processes on several hosts, you can use a single start profile. Each SAP instance does not have to have its own start profile.

See also:

Profiles [Seite 240]

Default Profiles [Seite 261]

Where Do Profiles Come From? [Seite 244]

Start Profile: Changing Instance Services

Purpose

To start an additional SAP service process on a computer (or delete an SAP service process), you must adapt the start profile accordingly.

Process Flow

You do not want the message server to run on Host A, but on Host B. You must adapt the start profiles for the instances on both hosts accordingly. In this case, the name of the start profile file should end with `_<hostname>`.

If you are using a start profile to start several SAP instances, the change will effect all these instances. In this case, the name of the start profile should **NOT** end with `_<hostname>`.

See also:

Changing Profile Parameters [Seite 283]

Default Profiles

Definition

If you want to assign the same parameter value for all application servers (such as the name of the database host, or the host on which the message server is running), enter it in the default profile. In general, you can list any parameter you like here.

SAP recommends that you use the following values:

Parameter Definition	Parameter Name in Profile
Name of the database host	SAPDBHOST
Name of the update server	rdisp/vbname
Name of the enqueue server	rdisp/enqname
Name of the server for handling background processing events	rdisp/btcname
Name of the computer on which the message server is running	rdisp/msname
Name of the TCP service under which the message server can be reached	rdisp/msserv
Name of the computer on which the SNA Gateway is running	rdisp/sna_gateway
Name of the TCP Service under which the SNA Gateway can be reached	rdisp/sna_gw_service

You **cannot** choose a name for the default profile. It is always called DEFAULT.PFL. The default profile, like all other profiles, is located in the global profile directory of the SAP System. For example, under UNIX it is located in the directory /usr/sap/<SID>/SYS/profile (<SID> = SAP System name). There is always one active default profile.

Default profiles are also called **system profiles**.

(Section from a default profile):

```
SAPSYSTEMNAME          = BIN
SAPDBHOST                   = hs0011
DIR_ORAHOME                  = /oracle/BIN
rdisp/mshost           = hs0311
rdisp/vbname           = hs0311_BIN_53
rdisp/enqname          = hs0311_BIN_53
rdisp/btcname          = hs0002_BIN_53
rdisp/sna_gateway      = is0001
```

Default Profiles

```
rdisp/sna_gw_service  = sapgw00
```

See also:

Profiles [Seite 240]

Default Profile: Changing Global Instance Parameters

Process Flow

To change a global system parameter, which is valid for all participating instances of an R/3 System, you must adapt the default profile.

Instance Profiles

Definition

Instance profiles provide an application server with additional configuration parameters to complement the settings values from the default profile [Seite 261]. Typically, these parameter settings adapt the instance according to the desired resources. They also define the available instance resources (main memory, shared memory, roll memory and so on), and how to allocate memory to the SAP application buffers.

Below is a typical instance profile:

```
#.***********************************************
#.*   Instance profile BIN_DVEBMG53            *
#.*   Version        = 000005                  *
#.*   Generated by user = BLOCHING             *
#.*   Date generated    = 04.08.1995, 11:10:35  *
#.***********************************************

INSTANCE_NAME = DVEBMG53
SAPSYSTEM = 53
SAPSYSTEMNAME = BIN
abap/buffersize = 40000
abap/programs = 600
ipc/shm_psize_10 = 15000000
ipc/shm_psize_14 = 0
ipc/shm_psize_17 = 0
ipc/shm_psize_18 = 0
ipc/shm_psize_19 = 0
ipc/shm_psize_40 = 17000000
rdisp/PG_MAXFS = 4096
rdisp/PG_SHM = 1000
rdisp/ROLL_MAXFS = 16384
rdisp/ROLL_SHM = 200
rdisp/btctime = 60
rdisp/wp_no_dia = 5
rdisp/wp_no_enq = 1
```

```
rdisp/wp_no_vb = 1
rdisp/wp_no_vb2 = 1
```

You can choose any name for an instance profile. The SAP naming convention is as follows: `<SID>_<instancename>` or `<SID>_<instancename>_<hostname>`.

```
C11_DVEBMGS12.
C11_DVEBMGS12_hs0011
```

To start application servers on several computers using identical parameter settings, you can use a single instance profile. It is generally not necessary for each application server to have its own instance profile. Instance profiles are also called **system profiles**.

See also:

Profiles [Seite 240]

Where Do Profiles Come From? [Seite 244]

Instance Profile: Automatic Adjustment of SAP Shared Memory Pools

Definition

The SAP pool buffers help interprocess communications of all SAP work processes from within an R/3 instance. These buffers are distributed among the shared memory segments of the operating system. These pools are dependent on the size and number of the shared memory segments provided by the operating system. As it can be somewhat laborious to set the profile parameters in the instance profile manually, this procedure has been automated.

The R/3 System detects the set of parameters, the shared memory pool requests. This set is compared to the parameters the system offers. In case of divergence, you can choose the set of parameters that the system will use.

Instance Profile: Changing Instance Parameters

Purpose

To change a parameter value for an R/3 instance, you must identify the instance profile used at startup time.

Process Flow

If you want to increase the number of work processes provided for background processing, the name of the start profile file should end with `_<hostname>`.

If an instance profile is used by several instances, profile changes will have an effect on all these instances. In this case, the name of the start profile file should **NOT** end with `_<hostname>`.

Switching Operation Modes Using Start and Instance Profiles

Prerequisites

To switch from one operation mode to another in the R/3 System, all application server instance definitions must be recognized by the operation mode definition.

Each individual operation mode should contain configuration specifications for each application instance involved. That is, how many work processes and which services (for example, dialog, background processing, spool) each application instance will provide.

Process Flow

You can get this information from the relevant instance profile, or from within the operation modes tool set. You must also verify that you have a corresponding start profile so that any application instance can be started from within the R/3 System.

When you switch operation modes, the new instance settings should be based on the existing profile definitions on the operating system. That is, the number of work processes and important directory path names should be consistent.

You are strongly advised to follow the recommendations above. The benefits of this procedure are:

- The profiles and operation mode data are always consistent.
- You can change profile data, for example the number of work processes, using the profile maintenance tool, and the corresponding operation modes can then be adjusted automatically.
- You no longer have to work with the operating system profile files.
- When **all** servers are assigned profiles from the profile maintenance tool, the whole system is checked thoroughly for any consistencies.

See also:

Operation Modes [Seite 192]

What to do After Installing an SAP Instance

Use

Once you have set up a new instance with the SAP installation program `R3INST`, you can decide whether you want to keep the associated instance profiles or delete them. You can either:

- Keep the profiles to represent one or more instances.

  To use the profiles generated or modified by `R3INST` (start, default and instance profile), they should be imported directly into the SAP profile maintenance database.

- Remove the profiles.

  There are already other profiles in the system that support or are shared by more than one instance. To use the existing start and instance profiles instead of the profile files generated by `R3INST`, identify and delete the files generated by `R3INST` in the global profile directory.

  Once the profiles have been removed, the startup/shutdown scripts of the new instance must be modified to enable instance startup/shutdown from operating system level. The scripts to modify are called `startsap` or `stopsap`.

Procedure

To modify `startsap` or `stopsap`:

1. Fill the variable `START_PROFILE` (in the scripts `startsap` and `stopsap`) with the name of the start profile you require. These two scripts are located in the home directory of the SAP administrator (for example, `c11adm`).
2. Use `<hostname>` for a new Release 3.0 installation of the start and instance profiles. An upgrade from Release 2.2x must be performed without `<hostname>`.
3. In the start profile file, you may have to adapt the name of the instance profile used to start the SAP instance programs. To do this, use the CCMS profile maintenance tool.

See also:

Changing Profile Parameters [Seite 283]

Importing Profiles [Seite 276]

Importing Profiles

Use

When you first install an SAP System, upgrade to a new SAP release or add a new application server, the system automatically generates or updates SAP instance profiles [Seite 266] at operating system level. Unfortunately, the installation program cannot save these profiles directly to the database. You must therefore import the SAP instance profiles before you can then edited them.

You can import profiles at any time, not only when you change profiles [Seite 283] for the first time. You can also use the CCMS profile maintenance tool to generate a new operating system copy of all the active profiles that are currently in the profile maintenance database.

Ensure that all application servers (instances) are active before you import profiles.

Procedures

To import SAP profiles from all active application servers (mass import):

1. Call the CCMS profile maintenance tool by choosing *CCMS* → *Configuration* → *Profile Maintenance*. Alternatively, call Transaction RZ10.
2. Choose *Utilities* → *Import profiles* → *of active servers.* The system imports the default profile and all start and instance profiles that are used by the SAP instances. The system checks the profiles and displays a log. The names of the profiles in the database are taken from the corresponding file names on the operating system.

   If you import the instance profile `/usr/sap/C11/SYS/profile/C11_D53`, then the profile in the database will be called `C11_D53`.
3. Once you have finished importing the profiles, position the cursor on the *Profile* field and press `F4`. The system displays the names of the profiles that were imported.

To import individual profiles:

You should use this function if you have installed a new application server, or if a profile was modified at the operating system level. You must first create a new profile [Seite 281] using the profile maintenance function.

1. On the initial screen of Transaction RZ10, enter the profile name (the version number is generated automatically).

Profile type and status now appear on the screen for your information.

2. Choose *Profile* → *Create*.
3. Maintain the administration data: Short description, the file name in which the profile should be activated (you must specify a reference server and profile type).

4. Once the administration data has been transferred, choose *Profile* → *Import* on the basic profile maintenance screen.
5. The system displays a dialog box in which you should specify the operating system file into which the profile should be imported. You can display all the profile files which are in the global profile directory using the F4 key.

   The system checks the imported profile for errors. You can now edit and/or import the profile into the database as described above. Once the import process is complete, you can decide whether you want to activate [Seite 287] the profile.

See also:

Start Profiles [Seite 255]

Default Profiles [Seite 261]

Instance Profiles [Seite 266]

Profile Maintenance

Use

You can frequently make changes to system profiles [Seite 240] for non-productive systems. You will rarely need to make changes to system profiles for productive SAP Systems. In either case, you should use the CCMS profile maintenance tool to maintain profiles.

The profile maintenance tool has the following advantages:

- Easy to use
- Extensive profile checks
- You can test the consistency of (several) individual profiles
- You no longer need to edit profiles at operating system level
- Changes to profiles are logged
- Profile data is maintained in the SAP database (ensuring security and consistency)
- Provision of basic data for operation mode switching

You should no longer edit profiles at operating system level using an editor. It only makes sense to do this if the SAP System cannot be started in any other way.

Integration

You can maintain profiles using either basic or extended maintenance.

The basic maintenance option displays general data on the profile and the parameters for the buffer sizes, number of work processes and the swap requirement.

Extended profile maintenance displays all individual parameter values for the profile.

Detailed documentation is available for each parameter contained in the profile. To display this, position the cursor on the parameter and choose F1.

Prerequisites

To use the profile maintenance tool, assign authorization values to authorization object `S_RZL_ADM` (Computing Center Management System Administration), according to the editing mode that you require:

Editing Mode	Value of Activity Field of Authorization Object `S_RZL_ADM`
Edit and display profiles	01, 03
Display profiles	03

Profiles on disk must then be read into the SAP System so that standard maintenance procedures can be executed.

See also:

Where Do Profiles Come From? [Seite 244]

Creating and Copying Individual Profiles

Use

You can use the CCMS profile maintenance [Seite 278] tool to create and copy individual profiles [Seite 240].

Procedures

To create a new profile manually:

1. Call the CCMS profile maintenance tool by choosing *CCMS* → *Configuration* → *Profile maintenance*.
2. Specify the name of the profile that you want to create and choose *Profile* → *Create.* The system creates the version number automatically.
3. Enter a short description of the profile, the filename into which the profile should be activated, a reference server and the profile type. Then choose *Copy.*
4. You can maintain parameters and their values by flagging either *Basic* or *Extended maintenance* and choosing *Maintain*.

   It is best if you start with *Basic maintenance* as this is where the parameters and their active values in the reference server are allocated.
5. Once you have made the changes, choose *Copy* to return to the initial screen of Transaction RZ10.
6. You must now check, save and activate [Seite 287] the profile that you have just created.

To copy an existing profile:

1. If necessary, change the profile parameters [Seite 283] of the existing profile in Transaction RZ10.

   Ensure that the servers for the source profile and the new profile have similar configurations. You should also align the host's operating system platforms on which the servers are running. This is important for the file names.
2. Choose *Profile* → *Copy*.

   You should also enter the new instance profile in the SAP instance start profile where the instance profile will be activated.
3. You must now check, save and activate [Seite 287] the profile that you have just created.

Changing and Switching Profile Parameters

Use

You can use the CCMS profile maintenance [Seite 278] tool to change and switch profile parameters [Seite 247].

Procedures

1. Call the CCMS profile maintenance tool by choosing *CCMS* → *Configuration* → *Profile maintenance.* Alternatively, call Transaction RZ10.
2. Specify the name of the profile whose parameters you want to change in the *Profile* field.
3. If you want to change the name of the file in which the profile should be activated, check *Administration data* and choose *Change*. Once you have made the changes, choose *Copy*.

    If SAP System profiles are imported automatically, the system creates administration data automatically.

4. Maintain the most important profile parameters by checking *Basic maintenance* and choosing *Change*. Once you have made the changes, choose *Copy.*
5. You can create, change or delete all parameters in a profile by choosing *Extended maintenance*. Once you have made the changes, choose *Copy.*

    You can switch between *Basic* and *Extended maintenance* at any time. Changes to profiles are adopted when you save them to the database [Seite 287].

    You can subsequently change and switch SAP profile parameters for instance profiles without having to restart the system.

To switch profile parameters:

1. Enter the profile name in the input field of Transaction RZ10 and choose *Profile* → *Dyn. switching* → *Display parameters.*

The system displays a list of memory management parameters that can be switched.

2. Change the relevant profile parameters using basic maintenance (as described above) and save them.
3. On the profile maintenance main screen, choose *Profile* → *Dyn. switching* → *Execute.*
4. From the list of active application servers, choose an application server and confirm.

The profile is used immediately after the change; you do not have to restart the system. At the end, the system displays a log for parameter switching.

See also:

Profiles [Seite 240]

Profile Checks

Definition

You can execute extensive checks of one or more profiles [Seite 240] once they have been changed. You can check the semantics, syntax and parameter names of the profile.

The system displays the results of the profile checks as a log that contains either warnings or error messages.

Single Profile Checks

The parameters in a single profile are divided into classes. For each class, there is a separate check rule. The table below shows these rules and includes an example for each:

Parameter Class	Check Rule	Example
Integer value	- Value less than default value → Error - Value greater than *Default* by a factor of 10 → Warning	`rdisp/TRACE`
Time value	- Value less than *0* → Error - Permissible characters: 0-9	`rdisp/btctime`
Boolean value	- Valid value set: 0, 1 ON, OFF YES, NO TRUE, FALSE - Other values → Error	`gw/keep_process`
File directory	- Directory does not exist → Warning - No *write authorization* → Warning	`rdisp/workdir`
File name	File does *not* exist → Error	`abap/rsyn`
TCP computer name	- Host is not known to TCP → Error	`rdisp/mshost`
TCP service name	- Service name is not known to TCP → Error	`rdisp/msserv`
SAP server name	- Server name is not known to R/3 System → Error	`rdisp/vbname`
File mask	- Check is not possible	`enque/log_file`
Strings	- Check is not possible	`abap/locale_ctype`
Cannot be changed by customer	- If particular parameters are changed by the customer → Error	`transport/systemtype`
Special parameters	- If value does not contain a particular character string → Error	`rdisp/bufrefmode`

Start profile	- If one program at least is not started → Error	

Multiple Profile Checks

In addition to the single profile check, the system checks all profiles (of a profile type) that are used in an SAP System to ensure that they are consistent with each other. For example, all start profiles are checked to see whether exactly one message server is started. With instance profiles, the system checks that enqueue work process was configured.

You can check profile sets for:

- All profiles of an active SAP System
- All profiles used for operation mode switching.

See also:

Reference Server of a Profile [Seite 246]

Profile Versions [Seite 290]

Instance Profile: Automatic Adjustment of SAP Shared Memory Pools [Seite 269]

Finding Manual Changes to a Profile [Seite 292]

Checking, Saving and Activating Profiles [Seite 287]

Instance Profile: Automatic Adjustment of SAP Shared Memory Pools [Seite 269]

Saving, Checking and Activating Profiles

Use

Once you have finished maintaining [Seite 278] a profile, you can check it for errors or inconsistencies, save it in the database and then activate it.

The CCMS profile maintenance tool archives profiles as operating system files and also stores a reference copy in the database. The database copy is used to create the profiles at database level. This process is called activating a profile.

The profile maintenance tool ensures that changes to profiles are activated when the corresponding SAP instance is restarted. You cannot make changes to an SAP instance during active operation.

Procedures

Saving profiles

1. Call the CCMS profile maintenance tool by choosing *Administration* → *CCMS* → *Configuration* → *Profile Maintenance*.
2. Specify a profile name in the *Profile* field and then choose *Profile* → *Save.*

Checking single profiles:

Enter the name and version of the profile on the initial screen of Transaction RZ10. Then choose *Profile* → *Check*.

Checking all profiles for the active server:

On the initial screen of Transaction RZ10, choose *Utilities* → *Check all profiles* → *on the active server*.

Checking all profiles used in operation modes:

On the initial screen of Transaction RZ10, choose *Utilities* → *Check all profiles* → *in operation modes*.

Activating profiles:

1. On the initial screen of Transaction RZ10, enter the name of the profile you want to activate. Use the possible entries pushbutton to display the profiles that are available.
2. Choose *Profile* → *Activate*.
3. Stop and restart the SAP instance(s) in which the profile changes should take effect. You can do this using the CCMS control panel [Seite 956].

The system creates automatically creates a backup file with the extension `.bak` for active profiles that already exist in an SAP System. You can copy and display this backup file even if the SAP System is not available.

See also:

Checking Active Parameters [Seite 294]

Creating and Copying Individual Profiles [Seite 281]

Profile Versions

Definition

When you modify [Seite 283] and save [Seite 287] a profile, the old status in the database is not overwritten. Instead, a separate version with the changed values is created. The SAP System allocates an individual number for each profile version. For example, when you create a new profile, it has the version number 1. Additional versions are numbered 2, 3, 4,... etc.

The associated profile file at operating system level is only overwritten if this is explicitly specified within the profile maintenance transaction (the system prompts you accordingly when you save a profile). You can only activate the most recent version of a profile. However, you can reactivate old versions by copying [Seite 281] them.

For each individual profile parameter that you change, an appropriate comment is stored in that same profile. For example, if user BATCHMAN changes the number of update work processes in an instance profile from 0 to 1. The following entry is made:

```
# old value: 0     changed by: BATCHMAN, 9.8.1995, 16:45:38
```

```
rdisp/wp_no_vb = 1
```

Using the versions of the profile and the associated modification comments, you should be able to trace the history of all changes.

See also:

Saving, Checking and Activating Profiles [Seite 287]

Start Profiles: Changing Instance Services [Seite 259]

Default Profiles: Changing Global Instance Parameters [Seite 264]

Finding Manual Changes to a Profile

Use

Once you have started an instance, you can confirm whether a profile [Seite 240] has been changed manually at operating system level by comparing the profile information in the database with the active [Seite 287] profile in the operating system.

If the profile data does not match, the operating system files have been changed manually. The system displays a log of any discrepancies that are found and triggers an alert in the alert monitor [Seite 812].

Procedure

1. Call the CCMS profile maintenance tool by choosing *CCMS* → *Configuration* → *Profile maintenance*.
2. Specify the profile name and version that you want to examine and choose *Profile* → *Compare* → *Profile in database* → *with active profile.* If system displays a log if it finds any discrepancies.

Checking Active Parameters

Use

You can use the ABAP program RSPARAM to find out which parameters are active for a particular SAP instance.

Procedure

1. Call program RSPARAM.
2. Start this report on the SAP instance which has the parameter values you are interested in.

The system displays a list that consists of two parts:

- The first part shows the parameter values in “unsubstituted” form, that is, before the substitute variables were replaced.
- The second part displays the actual values of the individual parameters.

See also:

Substitute Variables in Profile Values [Seite 242]

Deleting Profiles

Use

You can delete either single profiles [Seite 240] or all versions of a profile.

Procedure

1. Call the CCMS profile maintenance tool [Seite 278] by choosing *CCMS* → *Configuration* → *Profile maintenance.* Alternatively, call Transaction RZ10-
2. You delete a single profile by choosing *Profile* → *Delete* → *Single profile*.
3. Delete all versions of a profile by choosing *Profile* → *Delete* → *All versions of a profile*.

Initially, the profiles are deleted from the database. However, you can also delete the corresponding profile files at operating system level.

External Operating System Command: Contents

Executing External Commands

Procedure

To call an external command:

Choose *CCMS, Jobs* → *External Commands*.

Alternatively, call Transaction SM49.

The system displays the *Operating system commands* dialog box, which lists all of the commands that are defined in the system.

To execute a command:

Select the line containing the command and choose *Execute*.

The command is then displayed along with its parameters. Before you execute the command from here, you can specify additional parameters (if this is allowed) as well as the name of a target computer. The default entry for target computer is the name of the application server (`SY-HOST`).

You can currently only execute commands synchronously in dialog mode.

To maintain an external command

You can also maintain commands from this window.

Choose *Display* → *Change*.

However, you may not execute changed commands until their changes have been saved.

See also:

What Information is Displayed? [Seite 309]

Additional Parameters [Seite 314]

Maintaining External Commands [Seite 302]

Maintaining External Commands

Use

You can create, display, change, delete, rename, and copy external commands. SAP commands cannot, however, be changed in customer systems.

External commands can only be processed individually. For example, you cannot delete several external commands at the same time. External commands can only be processed by one user at a time.

Procedure

1. Choose *CCMS, Configuration* → *External Commands*. Alternatively, call Transaction SM69.
2. You can then display or change the external commands. To display a list of all external commands defined in the system, choose *Display* ↔ *Change*.

    External commands are uniquely identified by a user-definable logical name and an operating system.

3. You can specify the name of an R/3 function module which, when executed, decides whether the specific command should be executed or not. The interface to this check module must be the same as for the check module '`SXPG_DUMMY_COMMAND_CHECK`' delivered by SAP.

    SAP recommends that you copy '`SXPG_DUMMY_COMMAND_CHECK`' before using it. Do not change this function module in your system.

    Command names beginning with 'X' or 'Z' are reserved for customer commands.

    Before Release 3.0C, external commands must be confirmed in the target system after a transport (between customer systems). Otherwise they are considered invalid. When you call the list of external commands, the system informs you of the need to confirm the commands.

See also:

What Information is Displayed? [Seite 309]

Additional Parameters [Seite 314]

Executing External Commands [Seite 300]

External Commands: Overview

Definition

An external command is a predefined operating system command that can be executed within the R/3 System.

Both the maintenance and execution of external commands is protected with R/3 authorizations. You can maintain and execute external commands online (from the CCMS menu) or in ABAP programs, using special function modules. External commands can also be executed as a step in a background job.

As the R/3 CCMS administrator, you can modify commands and activate additional security mechanisms without having to change the program.

You can add your own commands and parameters to the variety of (predefined) commands delivered by SAP.

See also:

Authorizations for External Commands [Seite 306]

Executing External Commands [Seite 300]

Maintaining External Commands [Seite 302]

Security Checks [Seite 318]

Authorizations for External Commands

Use

To create or change external commands, you require appropriate authorization.

Procedure

Define the authorizations for executing external commands in the R/3 System and assign them to users through authorization profiles. Authorizations can be given for all external commands, for groups of external commands, or for individual commands. They can also be restricted to specific computers.

The R/3 System does not differentiate between capital and lowercase letters in authorizations.

Table: Meaning of R/3 Authorizations

R/3 Authorization	Meaning	
S_RZL_ADM with activity '01'	Maintenance of external commands The authorization should be queried in the ABAP program with the function module 'SXPG_MAINTENANCE_PERMISSION', instead of with AUTHORITY_CHECK.	
S_LOG_COM	Execution of external commands has three fields:	
	COMMAND	Name of the external command
	OPSYSTEM	Operating systems for which the command was defined (does not have to be identical to the operating system of the target computer)
	HOST	Symbolic computer name of the target system
	The fields *COMMAND* and *OPSYSTEM* are used to uniquely identify the external command, while *HOST* defines the authorizations for executing commands on certain target computers.	

The authorization S_LOGCOM_ALL, which enables the execution of all external commands, is delivered as standard in the profiles S_A.SYSTEM and S_A.ADMIN.

See also:

Security Checks [Seite 318]

Which Information is Displayed?

Definition

When you execute or maintain external commands, a list of the commands is displayed. Each line in the list contains a brief summary of the most important information for an individual command.

The first two columns contain the name of the external command and an operating system on which the external command should be executed.

Operating system

An external command is uniquely identified by its name and operating system. These two key fields therefore also have to be passed when an external command is executed. (No difference is made between capital and lowercase letters.)

The operating system does not have to be the same as the specific operating system. However, we recommend that you adapt the type to the ABAP runtime variable **SY-OPSYS** where possible, as the default for operating system is **SY-OPSYS** when an external command is executed.

You can also enter the name of the syntax group belonging to **SY-OPSYS** or the name **ANYOS**, provided the corresponding command can be used universally.

See also: Documentation on **SXPG_CALL_SYSTEM**.

Type

In the column *Type* you find two different identifiers: *SAP* and *Customer*.

External commands of the type *SAP* are delivered by SAP and cannot be changed in customer systems. Customers can create commands of the type *Customer*. These commands are not changed during R/3 release upgrades.

We recommend that you only use commands of the type *Customer* for in-house developments (that is, in your own ABAP sources).

Commands are automatically created with the type *Customer* in customer systems.

OS command

In the fourth and fifth columns, you can see the specific operating system command and the predefined parameters with which it should be executed. These two values are not shown in their entirety in this list, but can be defined with up to 128 characters each.

See also:

Detailed Information [Seite 312]

Additional Parameters [Seite 314]

Which Information is Displayed?

R/3 Profile Parameters in External Commands [Seite 316]

Displaying Detailed Information

Procedure

You can display detailed information on a command by double-clicking on it.

The system displays the operating system command and the predefined parameters in full, as well as whether or not additional parameters can be included when the command is executed.

For security reasons, information relevant to system security is **only** displayed in 'Display mode' if the user has CCMS administrator authorization. This information includes, for example, the name of the check module and the information whether an entry is made in the system log when the command is executed.

See also:

Additional Parameters [Seite 314]

R/3 Profile Parameters in External Commands [Seite 316]

Additional Parameters

Definition

When you execute an external command, you can include a character string of up to 128 characters containing additional parameters.

Prerequisites:

- The use of additional parameters must be allowed for this command (standard).

  Maintaining External Commands [Seite 302]

- The total length of all parameters (predefined and passed at runtime) cannot exceed the maximum length of 128 characters.

Placeholders

These additional parameters are generally added to the end of predefined parameters following a blank space. You can, however, also define placeholders for these additional parameters within the predefined parameters.

Table: Placeholders for Parameters

Placeholder	Meaning
'?'	used for required parameters
'&'	used for optional parameters

If no parameter values (or a blank character string) is passed at runtime for a required parameter, then the exception `PARAMETER_EXPECTED` is triggered. The placeholder '&' for optional parameters is removed in this case. Otherwise, the characters '?' and '&' are replaced with the parameter character string passed.

See also:

Detailed Information [Seite 312]

R/3 Profile Parameters in External Commands [Seite 316]

R/3 Profile Parameters in External Commands

Definition

You can decide whether the value of an R/3 profile parameter is entered in the command text at runtime (for operating system commands and the pre-defined parameters). A placeholder for an R/3 profile parameter must have the following syntax:

`$-SAPSYSTEMNAME-`

The first place must always contain a '`$`', followed by a separator which does not appear in the names of any R/3 profile parameters. The subsequent text is interpreted as the name of an R/3 profile parameter until the separator appears again. The complete character string from '`$`' up to and including the second separator is replaced by a value defined for the R/3 profile parameter before the operating system command is executed. The value of the profile parameter is used as defined in the target system, provided an R/3 application server was started there.

`/usr/sap/$-SAPSYSTEMNAME-/SYS/exe/run/sapgui`

If the R/3 profile parameter `SAPSYSTEMNAME` had the value '`ABC`', then the character string above would look like this:

`/usr/sap/ABC/SYS/exe/run/sapgui`

Special Characters

Use the following special characters:

$	To identify R/3 profile parameters
?	Placeholder for required parameters
&	Placeholder for optional parameters
#	Escape character

If you want to use '`$`' as a regular character, then you need to precede it with '`#`' as escape character. You must enter the escape character '`#`' twice if you want to use it in the text.

See also:

Detailed Information [Seite 312]

Additional Parameters [Seite 314]

Security Checks

Use

Before external commands are executed, the additional parameters passed are checked. If "illegal" characters are found in the process, the command is not executed and the exception `SECURITY_RISK` is triggered. These illegal characters have been defined specific to operating systems, as displayed below:

Table: Illegal Characters for Parameters

Operating System	Illegal Characters
AIX	| & ; ^ \ < > ‘
HP-UX	| & ; ^ \ < > ‘
Windows NT	| & < > ()
VMS	
other	| & ; ^ \ < > ‘

If you want to prohibit the use of other characters, SAP recommends that you use check modules. If you want to avoid these restrictions, you should use shell scripts or in-house C programs.

See also:

Check Modules [Seite 320]

Illegal Changes to External Commands [Seite 322]

Syslog Trace and System Alert Monitor [Seite 325]

Check Modules

Definition

When maintaining an external command, you can specify the name of a check module. A check module is a function module that is executed immediately before the command, and can decide whether the command will be executed or not.

Your R/3 System already contains the prototype of such a check module:

`SXPG_DUMMY_COMMAND_CHECK`

This check module contains an ABAP comment with an example of how it is used.

Do not change the check module `SXPG_DUMMY_COMMAND_CHECK`, but instead work with a copy. You can freely define the name of the check module. Only the interface has to be the same as for the check module `SXPG_DUMMY_COMMAND_CHECK`. The check module can only stop the execution of the command by triggering the exception `NO_PERMISSION`.

See also:

Illegal Changes to External Commands [Seite 322]

Syslog Trace and System Alert Monitor [Seite 325]

Processing Illegal Changes to External Commands

Use

When a CCMS administrator calls the command maintenance transaction, he/she is informed of any changes that were made to external commands by means other than the maintenance transaction.

The administrator can then decide whether or not to accept these changes or to restore the old status. If a user tries to execute a command entered or changed illegally, the command will not be executed, and this is recorded in the system log.

Procedure

To accept or reset changes:

Illegal changes are displayed in two lists. You can switch between these two lists by selecting *Show original values* and *Show changed values*.

1. Select (in the appropriate list) the command that you want to accept or reset.
2. Choose *Reset to orig. value* **or** *Accept ill. changes*.

   Note the list in which you are currently working:

   - If you are in the list of original commands, the changes will be reset in all of the commands selected.
   - If you are in the list of changed commands, the changes will be accepted.

   In both cases, the corresponding commands are then removed from the lists of changed commands.

Comparing Old and New Commands

As a CCMS administrator, you can use the function for comparing "old" and "new" external commands. You can also see a note that states the nature of each change in the second column immediately after the checkbox the "old" and "new" commands:

C	Command was newly created
D	Command was deleted
M	Command was modified

As soon as you exit the list, the system displays the list of all external commands defined in this R/3 System.

See also:

Check Modules [Seite 320]

Syslog Trace and System Alert Monitor [Seite 325]

Syslog Trace and System Alert Monitor

Use

If a user tries to execute a command that was created or changed illegally, an entry is automatically made in the system log.

The syslog area 'LC' is reserved for external commands.

You can decide whether these events are displayed in the alert monitor.

To maintain the alert thresholds:

From the main R/3 menu, choose *Tools* → *CCMS, Configuration* → *Alert monitor* → *Threshold values (3.x)*.

Alternatively, call Transaction RZ06.

See also:

Check Modules [Seite 320]

Illegal Changes to External Commands [Seite 322]

CCMS Monitoring

Purpose

The CCMS provides a variety of monitors to support the SAP environment and its surrounding components. These monitors are critical for understanding and evaluating the behavior of the SAP processing environment. If your SAP System is operating at unacceptable performance levels, they provide the proper tuning insight for making sure your SAP installation is operating at peak efficiency.

Features

The CCMS analysis monitors provide functions for:

- Checking system status and operation modes
- Locating and eliminating potential problems as quickly as possible
- Early diagnosis of potential problems, for example resource problems in the host or database system, that could adversely affect the SAP System.
- Analyzing and tuning the SAP System and its environment (host and database systems) in order to optimize the throughput of the SAP System.

The monitoring architecture [Seite 814] has replaced the previous monitoring and alert system in the CCMS.

The new monitor offers all of the functionality that was previously available, as well as new, more reliable alerts and more advanced and powerful features.

Use

You can use the following monitors independently, or execute them as analysis methods [Seite 896] from the alert monitor:

Global Work Process Load Monitor [Seite 1042]

Workload Monitor [Seite 965]

Operating System Monitor [Seite 750]

Operating System Collector [Seite 769]

SAP Buffers [Seite 1047]

Database Monitor [Seite 328]

Database Monitor

Purpose

The database monitor checks important performance indicators in the database system, such as database size, database buffer quality, and database indexes.

The database monitor works with each database system supported by SAP, and uses the statistics supplied by the database system that you are working with. You can access most of these database system statistics from the CCMS Performance Monitoring menu.

You can use the database monitor to:

- Monitor the database of an active production SAP System
- Analyze certain types of problems
- Get information that you need to tune the database system

Integration

Although the database monitor accesses and evaluates database-specific statistics, it has essentially the same appearance regardless of which database system you are working with.

You can call the database monitor from every application server in your SAP System. The data displayed by the database monitor is identical on each application server.

Features

The CCMS alert monitor [Seite 812] allows you to display the database statistics graphically, and check critical database functions more effectively.

SAP/Oracle Database Monitor

SAP/Oracle Database Monitor: Introduction

SAP has implemented its own database monitor and does not use the monitoring tools supplied by the database providers for the following reasons:

- Monitoring and administration are not always clearly separable.

  SAP requires that the database is monitored in write protected mode.

- SAP wants to provide the support team with a standard interface for monitoring database activity.

- Because of its three-tier client / server architecture, the R/3 System makes special demands on database monitoring software. It is extremely important that you have information from both the database and the R/3 System so that you can determine the database resources that a user or program occupies.

Much of this information in the database monitor comes from Oracle-specific monitoring views and tables. Oracle provides a wide range of information on the status of the database in virtual tables. These tables are stored in the working memory and are identified as dynamic performance tables or as Oracle V$ tables.

The SAP / Oracle database uses these and other Oracle administration tables to enter, analyze and present its information. To compile the information that is required, the system runs special ABAP reports and also accesses the Oracle data directly (Data for the Oracle Database Monitor [Seite 417]).

This help does not replace the Oracle Tuning Manuals, but is intended as an R/3 System-specific enhancement to the Oracle documentation, in particular the manuals *Server Concepts*, *Server Reference* and *Server Tuning.*

See also:

SAP/Oracle Database Monitor: Main Screen [Seite 331]

Detailed Analysis (Oracle) [Seite 339]

SAP/Oracle Database Monitor: Data Status [Seite 368]

SAP/Oracle Performance Monitoring Strategies [Seite 385]

Diagnosing SAP/Oracle Performance Problems [Seite 396]

SAP/Oracle Database Monitor: Main Screen

To display the main screen of the SAP/Oracle Database monitor:

Choose *CCMS* → *Control* → *Performance Menu* → *Database* → *Activity*.

Alternatively, use transaction `ST04`.

The main screen of the SAP/Oracle Database Monitor shows the most important indicators of Oracle database performance.

See also:

Database Monitor [Seite 328]

The information on the main screen *Database Performance Analysis* is subdivided into various sections. The most important sections are explained below:

Data Buffer (Oracle) [Seite 337]

Shared Pool (Oracle) [Seite 338]

Redo Log Buffer (Oracle) [Seite 335]

Calls (Oracle) [Seite 336]

Time Statistics (Oracle) [Seite 333]

Table Scans/Table Fetch (Oracle) [Seite 334]

Sorts (Oracle) [Seite 332]

See also:

Detailed Analysis (Oracle) [Seite 339]

SAP/Oracle Database Monitor: Status of the Data [Seite 368]

SAP/Oracle Performance Monitoring Strategies [Seite 385]

Sorts (Oracle)

This section shows the total number of sort operations along with the number of sort operations performed in memory or on disk.

Sort operations take place if you use `ORDER BY`, `GROUP BY` or `SORT MERGE JOIN` SQL statements. Sorting is also done during index creation. Sorting can be a very expensive process and should be avoided whenever possible. It is generally better for performance if sorting is done in memory than on disk.

See also:

Monitoring Sorting (Oracle) [Seite 395]

SORT_AREA_SIZE (Oracle) [Seite 415]

Time Statistics (Oracle)

An Oracle shadow process either runs actively on the PC (it uses *CPU time*) or else it waits. Wait situations can be divided into cases where the Oracle process is waiting because there is currently nothing for it to do ('idle waits') or where the Oracle process wants to run but first has to wait for a resource that is not yet available (*'busy waits*'). 'Total waits time' describes the sum of 'idle wait time' and 'busy waits time'.

Sessions busy is defined as *(CPU time + busy wait time) / (CPU time + total wait time). CPU usage* is defined as *CPU time / Elapsed time*. *Time/ User call* is defined as *(CPU time + busy wait time) / User calls*.

Note that the three ratios show mean values since database startup. If you want to determine the actual load at its peak, you should use the monitor's `Reset`-function.

See also:

Wait Events (Oracle) [Seite 344]

Table Scans/Table Fetch (Oracle)

This section of the monitor shows how database data is accessed.

A full table scan occurs when Oracle must read all data blocks of a table from disk. When the amount of data being read is small (short tables), this type of access is preferable. When the amount of data being read is large (long tables), index access may be preferable.

When data from a table is accessed via an index, Oracle performs the actual lookup using the rowID of the block holding the data. Access via this kind of index is normally very fast.

If a data record does not fit in one Oracle data block (whose size is determined by `db_block_size` (DB_BLOCK_SIZE (Oracle) [Seite 408])), it must be continued in another block (data chaining).

See also:

Table Scans: Problem Analysis (Oracle) [Seite 357]

Monitoring Table Access Methods (Oracle) [Seite 393]

Redo Log Buffer (Oracle)

This buffer (memory area in the SGA) holds information about changes made to data and objects in the database. The Oracle background process LGWR writes entries from the redo log buffer to the on-line redo log files on the disk.

Allocation fault rate shows the ratio of times Oracle attempted to find space available space in the redo log buffer and was unsuccessful. When this happens, the user process must wait until space in the buffer is free.

See also:

Monitoring the Redo Log Buffer (Oracle) [Seite 390]

LOG_BUFFER (Oracle) [Seite 411]

Calls (Oracle)

This section of the monitor shows the type and number of database accesses made on behalf of Oracle processes. The value for rollbacks indicates the number of times an Oracle process failed to complete the commit of an operation.

By monitoring database accesses, you can control the system load, separated by both user and internal operations.

See also:

Monitoring Calls (Oracle) [Seite 391]

Data Buffer (Oracle)

Data buffers have the following functions:

Table: Data buffer and their functions

Buffer	Function
Data buffer	holds Oracle blocks in shared memory (System Global Area or SGA)
Data buffer quality cache hit ratio (CHR)	measures the number of times that a data block requested by an Oracle process is found to be already in memory.
Physical reads and *physical writes*	Information is also provided on the number of physical input/output (I/O) operations performed on behalf of Oracle
Busy waits	the number of times a process had to wait to acquire a data buffer in a compatible state

See also:

Monitoring the Data Buffer (Oracle) [Seite 386]

Buffer Busy Waits (Oracle) [Seite 343]

DB_BLOCK_BUFFERS (Oracle) [Seite 407]

Shared Pool (Oracle)

The *shared pool* (shared memory area in the SGA) is used by Oracle to hold several key memory structures. Most important among these are the data dictionary cache and the shared SQL area.

The data dictionary cache contains information on Oracle objects e.g.

- naming
- definition
- access

It is regularly referenced by Oracle itself, as well as some application programs and database users.

The shared SQL area, also known as a "shared cursor cache", is a memory area which contains the parsed representation of SQL statements. Since a certain amount of system overhead is needed to parse an SQL statement, the ability to reuse statements already in memory can add a significant performance advantage.

See also:

Dictionary Buffer (Oracle) [Seite 346]

Monitoring the Shared Pool (Oracle) [Seite 388]

Monitoring the Shared SQL Area (Oracle) [Seite 398]

SHARED_POOL_SIZE (Oracle) [Seite 414]

Detailed Analysis (Oracle)

If the Alert Monitor indicates that the database has potential performance problems, you can analyze the database in more detail.

Use the analysis functions available from the main screen of the Database Monitor *Database Performance Analysis* (SAP/Oracle Database Monitor: Main Screen [Seite 331]).

All these functions provide numeric information on the database. A lot of the information can also be displayed graphically. You can access the most important analysis functions by *Detail analysis menu*. You will find a list below of some of the diverse analysis options - these are options which, from SAP's experience, are frequently used by customers.

Not all the analyses offered by the monitors are included in the list. Functions that you do not normally need are not listed. These functions are mainly used by the SAP Service & Support to analyze your R/3 and database system.

Table: Analysis options

Analysis	For detailed information, see...
File activity statistics	File System Requests (Oracle) [Seite 342]
Overview of wait situations	Wait Events (Oracle) [Seite 344]
Wait situations in the data buffer (buffer busy waits)	Buffer Busy Waits (Oracle) [Seite 343]
Data dictionary cache statistics	Dictionary Buffer (Oracle) [Seite 346]
Performance statistics per application server	SAP Client (Oracle) [Seite 347]
Resource consumption per Oracle shadow process	Oracle Sessions [Seite 348]
Resource consumption by SQL statements	Monitoring the Shared SQL Area (Oracle) [Seite 398]
Exclusive lockwaits	Exclusive Lockwaits (Oracle) [Seite 351]
ALERT file	Database Message Log (Oracle) [Seite 352]
Oracle statistics tables	Display V$ Tables (Oracle) [Seite 353]
Overview of database performance	Performance Database (Oracle) [Seite 354]
Monitoring datasets	State on Disk (Oracle) [Seite 355]
Changes to init.ora parameters	Parameter Changes (Oracle) [Seite 356]
Consistency checks	Consistency Checks [Seite 370]
Missing indexes	Missing Indexes [Seite 384]
Type of table scan	Table Scans: Problem Analysis (Oracle) [Seite 357]

Checking tablespaces	Checking for Full Tablespaces (Oracle) [Seite 359]
Monitoring storage space	Storage Management Errors (Oracle) [Seite 361]
Free space analysis	Checking for Freespace Problems (Oracle) [Seite 363]
Displaying storage parameters	Checking Storage Parameters (Oracle) [Seite 364]
Extent analysis	Extent Analysis (Oracle) [Seite 379]
MAXEXTENTS values	Problems with Maximum Number of Extents (Oracle) [Seite 366]
Display of Oracle table statistics for the cost-based optimizer	Display of Oracle table statistics

Detail Analysis Menu (Oracle)

To display more detailed Oracle performance statistics:

1. From the R/3 main screen, choose: *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Activity*

   Alternatively, use transaction code `ST04`.

2. Choose *Detail analysis menu*.

The analysis options displayed here are needed for advanced database monitoring operations. Some of the options are explained below:

Buffer Busy Waits (Oracle) [Seite 343]

File System Requests (Oracle) [Seite 342]

Wait Events (Oracle) [Seite 344]

Dictionary Buffer (Oracle) [Seite 346]

SAP Client (Oracle) [Seite 347]

Oracle Sessions [Seite 348]

Exclusive Lockwaits (Oracle) [Seite 351]

Database Message Log (Oracle) [Seite 352]

Display V$ Tables (Oracle) [Seite 353]

Performance Database (Oracle) [Seite 354]

State on Disk (Oracle) [Seite 355]

Parameter Changes (Oracle) [Seite 356]

File System Requests (Oracle)

To display the File System Requests:

1. From the R/3 main screen, choose: *Tools → Administration → Computing Center → Management System → Control → Performance Menu → Database → Activity*

   Alternatively, use transaction code **ST04.**

2. Choose *Detail analysis menu → Filesystem requests*.

This screen displays statistics on physical accesses to database files. The display includes duration for block reads and writes measured in milliseconds. To display these time values, the **init<SID>.ora** parameter **timed_statistics**(TIMED_STATISTICS (Oracle) [Seite 416]) must be set to **true**.

You can minimize the time needed for reading from or writing to a data file:

1. Identify the frequently used data files
2. Ensure that frequently used files are on separate disks so that I/O requests for objects do not directly compete with each other.

   Data file activity has an important effect on performance if your database is very large and is used intensively.

Buffer Busy Waits (Oracle)

To display *Buffer busy waits*:

1. From the R/3 main screen, choose *Tool* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Activity*

   Alternatively, use transaction code `ST04`.

2. Choose *Detail analysis menu* → *Buffer busy waits*.

The statistics presented on this screen tell you which types of Oracle block classes processes were waiting for. The number of waits for the four most common classes (data block, segment header, undo header and undo block) are usually shown with the cumulative time in milliseconds that processes were waiting.

These wait situations indicate that an Oracle process had to wait for the indicated block class which was in an inconsistent state.

See also:

Monitoring the Data Buffer (Oracle) [Seite 386]

Wait Events (Oracle)

To display *Wait events*:

1. From the R/3 main screen, choose *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Activity*

   Alternatively, use transaction code `ST04`.

2. Choose *Detail analysis menu* → *Wait events*.

This screen formats the data in the Oracle system tables V$System_Event and V$Session_Event. To display the current value, parameter `init<SID>.ora timed_statistics` (TIMED_STATISTICS (Oracle) [Seite 416]) must be set to `true`.

This screen displays the wait situation of the Oracle processes. There is a difference between cases where the Oracle process is waiting because there is currently nothing for it to do ('idle waits'), and where the Oracle process wants to run but first has to wait for a resource that is not yet available (*'busy waits*'). 'Total waits time' describes the sum of 'idle wait time' and 'busy waits time'.

'Idle wait' situations are:

- 'SQL*Net message from client' (the process is waiting for an SQL statement from the client, for example, the R/3 work process),
- 'rdbms ipc message' (the process is waiting for a statement from the RDBMS),
- 'dispatcher timer', 'virtual circuit status', 'pmon timer', 'smon timer', 'WMON goes to sleep', 'Null event'.

'Busy wait' situations are:

- Wait situation for physical I/O: 'db file sequential read', 'db file parallel write', 'log file sequential read', etc.:
- 'enqueue': wait situations due to exclusive database locks that can be examined using the 'Exclusive Lockwaits' screen.
- 'buffer busy waits': wait situations in the Oracle buffers: you can find details on this on the screen under 'Buffer busy waits'.
- 'log file switch (archiving needed)': problems with the log switch or with checkpointing. Look in the database message log for entries such as 'Cannot allocate log, archival required' or 'All online logs needed archiving'.
- 'SQL*Net more data to client' and 'SQL*Net more data from client': The Oracle process is waiting because data cannot be transferred quickly enough from, for example, the client. This wait situation indicates problems with the SQL*Net-Installation or with the network.

See also:

Oracle Sessions [Seite 348]

Buffer Busy Waits (Oracle) [Seite 343]

Exclusive Lockwaits (Oracle) [Seite 351]

Database Message Log (Oracle) [Seite 352]

Dictionary Buffer (Oracle)

To display Dictionary Buffer:

1. From the R/3 main screen, choose: *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Activity*

   Alternatively, use transaction code **ST04.**

2. Choose *Goto* → *Statistics* → *Row cache.*

This screen displays in-depth statistics on the quality of the Oracle data dictionary buffer cache (row cache). Entries for all cache objects are displayed here.

This information is read into memory from the dictionary tables stored on disk. When an Oracle instance is first started, this cache is necessarily empty and must be loaded as dictionary information is accessed. For this reason, hit ratios are generally low at database instance startup time and stabilize over time.

In version 6 of Oracle, there were **init<SID>.ora** parameters that could be changed if the values in *Used* were close to those in *Total*. This no longer the case with version 7. Instead, setting the data dictionary cache is an automatic process executed by the database itself. The only parameter you can use to control the data dictionary is **shared_pool_size** (SHARED_POOL_SIZE (Oracle) [Seite 414]).

You can allocate more room in the shared pool by increasing the value of this parameter. As the data dictionary cache is part of the shared pool, if necessary it can dynamically extend itself within the shared pool area, as long as there is sufficient free memory space available.

See also:

SHARED_POOL_SIZE (Oracle) [Seite 414]

SAP Client (Oracle)

To display SAP Client:

1. From the R/3 main screen, choose: *Tools → Administration → Computing Center → Management System → Control → Performance Menu → Database → Activity*

   Alternatively, use transaction code `ST04`.

2. Choose *Detail analysis menu → SAP Client*.

You can analyze the workload of the database for each application server in the R/3 System. This helps you find out which application servers put the highest workload on the database.

The fields on the statistics screen contain the following information:

Table: Information on the statistics screen

Field	Information
Current Open Cursors	Number of open cursors or context areas occupied on the application server by user processes
User calls	Number of database queries from users
Recursive calls	Number of internal data dictionary queries
User commits	Number of user transactions performed
User rollbacks	Number of user transactions terminated
Parse count	Number of "parsed" SQL statements
Database block gets	Number of logical read operations required to call the current version of the required data
Consistent gets	Number of logical read operations required to call a consistent version of the required data
Physical reads/writes	Number of physical read and write operations performed in the database
Redo blocks written	Number of blocks written by the log writer in the redo log
Long table scans, rows gotten	Number of full table scans of tables larger than four blocks and number of data records read sequentially
Table fetch by row ID	Number of table data records accessed directly
Table fetch by continued row	Number of chained data records that were read

Oracle Sessions

To display *Oracle sessions*:

1. From the R/3 main screen, choose *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Activity*.

   Alternatively, use transaction code **ST04**.

2. Choose *Detailed analysis menu* → *Oracle session*.

This screen displays information about the Oracle shadow processes from the Oracle system tables *V$Session, V$Process, V$Session_Wait* and *V$Sess_IO*.

The most important fields mean:

SID: Oracle session ID

ORA proc: Prozess ID (at operating system level) of the Oracle shadow process

Clnt proc: Prozess ID (at operating system level) of the R/3 work process

Client-Host: Name of the host running the R/3 work process

Status: 'ACTIVE' or 'INACTIVE'

Event: Event for which the process is waiting (only valid if the process is set to 'INACTIVE')

The *'Wait events'* screen displays statistics about the events.

You can use the `Filter` function to display only active sessions, or sessions that are not in the event 'SQL*Net message from client'.

You can display information - if available - on the R/3 work processes using the function `R/3 WPs`.

See also:

Wait Events (Oracle) [Seite 344]

Exclusive Lockwaits (Oracle) [Seite 351]

SQL Request (Shared SQL Area)

Now and again, executing a single SQL statement can have a negative effect on system performance for all users. This is possible, for example, if the scanned dataset is very large or if the the data returned must be processed (sorted) in large amounts. Statements of this type use CPU time ineffectively and database buffer and disk I/O operations reduce system performance for all users. It is the database administrator's task to monitor the shared cursor cache (also Shared SQL-Area), to identify uneconomical statements and to determine how to increase their performance.

To check the shared cursor cache:

1. From the main screen, choose *Tools → Administration → Computing Center → Management System → Control → Performance Menu → Database → Activity*.

   Alternatively, use transaction code `ST04`.

2. Choose *Detail analysis menu* and then *SQL Request*. You can also change the default selection criteria (Buffer gets >= 100.000, Disk Reads >= 10.000).

The most interesting entries for the database administrator are:

- *Total Executions* - The frequency a statement is executed
- *Disk reads* - Number of Oracle blocks read for the statement by the hard disk
- *Buffer gets* - Number of Oracle buffer blocks read for the statement from the data buffer
- *Records processed* - Number of table lines for the statement returned to the R/3 work process

You can see the SQL statements in the column *SQL Text* and display them in full by double-clicking on the line.

For an overview of the statement types frequently executed in the cursor cache, you can use *Sort* to arrange the display according to different areas.

In any case, do not worry if the value of *Total Executions* is high, as some statements must be regularly executed. If, on the other hand, a repeatedly executed SQL statement has a high number of *Reads* or *Gets* each time it is executed, you should analyze the system in detail. Check whether any indexes are missing or whether existing indexes are fragmented. Uneconomical SQL statements often access tables which would benefit from a new, secondary index. It is possible that indexes exist for the table, but that the SQL statement is written in such a way that it cannot use these indexes correctly.

Hinweis

From this screen you cannot tell which user or which ABAP program is responsible for the uneconomical statement. From time to time it can be a laborious process from realizing that a table contains uneconomical statements, to actually finding the program containing these statements. You can use the dictionary info system to find a description of a specified table. You can also determine where this table is used. This information should help you to restrict your search.

See also:

Monitoring the Shared SQL Area (Oracle) [Seite 398]

Missing Indexes [Seite 384]

Exclusive Lockwaits (Oracle)

To display Exclusive Lockwaits:

1. From the R/3 main screen, choose: *Tools → Administration → Computing Center → Management System → Control → Performance Menu → Database → Activity*

   Alternatively, use transaction code `ST04`.

2. Choose *Detail analysis menu → Exclusive lockwaits*.

Exclusive lockwaits are displayed here. A lockwait means that at least one process is locked through a lock held by another process. A request waits for a resource which is locked exclusively by another user. The process holding the lock and the waiting process(es) are displayed.

Database Message Log (Oracle)

To display Database Message Log:

1. From the R/3 main screen, choose: *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Activity*

   Alternatively, use transaction code **`ST04`**.

2. Choose *Detail analysis menu* → *Database message log*.

This function lists information from the ALERT file (an error and message file provided by Oracle). This file includes important information about error situations and the general status of the database. A good DBA will check this log regularly.

Display V$ Tables (Oracle)

To display Display V$ Tables:

1. From the R/3 main screen, choose: *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Activity*

   Alternatively, use transaction code **ST04.**

2. Choose *Detail analysis menu* → *Display V$ tables*.

You will get a list of V$ tables (dynamic performance tables), which are provided by Oracle for displaying statistics on the database system. For more information on these tables, refer to the Oracle documentation.

Performance Database (Oracle)

To display Performance Database:

1. From the R/3 main screen, choose: *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Activity*

   Alternatively, use transaction code `ST04`.

2. Choose *Detail analysis menu* → *Performance Database*.

A detailed database history is displayed. There is a variety of display options available:

You can display:

- activity peaks (*Peaks*)
- display delta values (click on a day and then choose *Intervals*)
- display peaks graphically (*Graph by column*)

When you choose a day, extracts of database activity are displayed at two-hour intervals. This is useful if you want to perform a trend analysis. The header line of this overview screen contains two additional dates. You can immediately select one of these days for further analysis.

State on Disk (Oracle)

To display Performance Database:

1. From the R/3 main screen, choose: *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Activity*

   Alternatively, use transaction code `ST04`.

2. Choose *Detail analysis menu* → *State on disk*.

The screen *Database Performance: Tables and Indexes* is displayed. From this screen, you can analyze the dataset of the SAP system. You can check the state of the data in the database and its correspondence to SAP data.

See also:

SAP/Oracle Database Monitor: Status of the Data [Seite 368]

Parameter Changes (Oracle)

To display Parameter Changes:

1. From the R/3 main screen, choose: *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Activity*

   Alternatively, use transaction code **ST04**.

2. Choose *Detail analysis menu* → *Parameter changes*.

This section lets you display the current and historical settings of the **init<SID>.ora** parameters, as well as the date they were changed. If you compare the changes in the database history (Performance Database (Oracle) [Seite 354]) with the parameter changes performed, you can roughly work out what effect the parameter change has had.

Note that changes to parameters do not take effect until the database instance is restarted. Refer to the relevant Oracle documentation for more information.

See also:

Important init.ora Parameters (Oracle) [Seite 406]

Table Scans: Problem Analysis (Oracle)

The *Table Scans* entry that appears in the Database Alert Monitor and the Database Monitor shows the number of sequential read operations on tables per day. If the number of sequential read operations per day is very high, you should perform further analyses. Sequential data access is generally not very efficient, which is why you should try to minimize the number of full table scans.

Causes

- Full table scans are often caused by missing table indexes. You can display tables with missing indexes by choosing *Database indexes* in the Database Alert Monitor.

  See also: Missing Indexes [Seite 384]

- Incorrect coding of **SELECT** SQL statements may also result in too many full table scans.

To identify tables affected by sequential read operations:

1. From the R/3 main screen, choose *Tools → Administration → Computing Center → Management System → Control → Performance Menu → Database → Activity.*

   Alternatively, use transaction code **ST04**.

2. Choose *Goto → Exceptions → Alert Monitor.*

   You reach the Database Alert Monitor.

3. Choose *Table Scans long tables* to display the applications servers and processes responsible for the table scans.

   If the processes belong to an Oracle user (for example, **SYS**), the table scans are actually caused by the database. Processes belonging to the SAP user **SAPR3** are important for further analysis.

4. Log on to the application server that is causing the table scans.

5. Use the Process Monitor to identify the user and report causing the table scans.

   From the R/3 main screen, choose *Tools → Administration → Monitor → System monitoring → Process overview*.

   Alternatively, use transaction code **SM50** (Work Process Load Monitor: Overview [Seite 1043]).

6. Find out which tables are used by this report.

   There are two ways of doing this:

- Start the report with an activated SQL trace
- Analyze the program.

7. Compare theses tables with those that appear in the list of tables with missing indexes. Choose *Database indexes* in the Alert Monitor.

   See also: Missing Indexes [Seite 384]

   If none of these tables have indexes missing, the table scans are probably caused by an SQL statement in the report that has not been optimized.

See also:

Table Scans/Table Fetch (Oracle) [Seite 334]

Monitoring Table Access Methods (Oracle) [Seite 393]

Monitoring the Shared SQL Area (Oracle) [Seite 398]

Checking for Full Tablespaces (Oracle)

In a production system you should check for full tablespaces on a regular basis in order to recognize storage problems early and avoid them. In particular, you should check whether there is sufficient space in all tablespaces before transmitting mass data (after system installation or release upgrade, for example).

If you find that the tablespaces are full, you should extend them. Additional storage space is then available. In exceptional situations, it may make sense to reorganize the tablespaces concerned.

To check tablespaces:

1. From the R/3 main screen, choose *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Tables/Indexes*

   Alternatively, use transaction code `DB02`.

2. In the *Tablespaces* section, choose *Current sizes*. You now reach the *Memory Management: Tablespaces* screen.

   The values in the *Used* column tell you which tablespaces are almost full.

   You can estimate the degree of fragmentation from the ratio of the values under *Tab/Ind* and *Extents*.

   See also: Monitoring Table and Index Fragmentation (Oracle) [Seite 400]

   To display tablespace information in graphical format:

   From the *Memory Management: Tablespaces* screen, choose *Graphics by columns*.

   The following information is displayed:

   – Size of the tablespace

   – Free storage space

   – Utilization

   To display an overview of the storage parameters of the tablespaces:

   From the *Memory Management: Tablespaces* screen, choose *Storage parameter* to display the storage parameters of the individual tablespaces. You can also display the storage parameters of the individual objects of a tablespace from the *Memory Management: Tablespaces* screen by clicking on the tablespace and performing a detailed analysis of the tables/indexes of this tablespace.

3. Choose *Analysis* for a closer check of tablespaces with storage space problems (that is, those with a high value in the *Used* column). You can analyze the tables/indexes of the tablespace, the associated files, the development history (statistics) and the freespace situation.

   See also: Tablespace Analysis (Oracle) [Seite 380]

Storage management problems can occur more frequently in some tablespaces than in others. You should always monitor these tablespaces closely, especially during data transfer after R/3 installation. To do this, you can use the SAP utility program SAPDBA with the `check` and/or `analyze` option.

See also: SAP Database Management: Oracle [Extern]

See also:

Storage Management Errors (Oracle) [Seite 361]

Checking for Freespace Problems (Oracle) [Seite 363]

Checking Storage Parameters (Oracle) [Seite 364]

Storage Management Errors (Oracle)

Storage management problems generally develop so slowly that you can recognize them and eliminate them before they seriously affect system operation. However, such problems can also arise very quickly during the data transfer phase when implementing your R/3 System if you transfer mass data from an old system into a new one.

If a problem becomes acute, you probably get one or both of the following database error messages. These are critical errors since they at least partially interrupt the operation of the database and the R/3 System.

- Tablespace overflow

  Problem: The database could not allocate another extent for a table or another index since the tablespace is full. The corresponding Oracle error is displayed.

  Solution: Extend the tablespace by creating another data file using the SAP tool SAPDBA.

  The file you create should be large enough to cover expected use of the tablespace in order to avoid recurrence of this problem in the long-term. Note that the Oracle database supports only a limited number of files (1024 files max.). SAP generally configures the database so that you can create up to 254 files on each platform. It is quite possible for you to reach this limit if you continually create small files when extending tablespaces.

  Precautionary measure: Check the storage space situation in the tablespaces on a regular basis. Alongside the options provided by the CCMS, you can use the analysis options of the SAP utility program SAPDBA. Tablespaces whose capacity is almost exhausted should be extended in good time. Use the appropriate SAPDBA function (Reference [Extern], Reference [Extern]).

  Analyses with the CCMS Database Monitor

  Tablespace Analysis (Oracle) [Seite 380]

  Extent Analysis (Oracle) [Seite 379]

- Extent overflow

  Problem: The database system could not allocate a new extent to a table or an index in a tablespace since the upper limit for the number of extents (MAXEXTENTS parameter) for this object was reached. The corresponding Oracle error is displayed.

  Solution:

  – Short-term: If the soft limit defined by SAP for MAXEXTENTS (usually 100) was reached, you can change the value for MAXEXTENTS. Use the SAP utility program SAPDBA (menu options *Reorganization* → *Alter table or index storage parameters*).

  – Long-term: If the method described above does not solve the problem since the hard limit for MAXEXTENTS (set to 505 extents for a block size of 8 KB by Oracle) will soon be reached, you must reorganize the object. Use the SAP utility program SAPDBA.

  Precautionary measure: Check for objects that are close to the MAXEXTENTS limit (or objects that are growing rapidly) on a regularly basis. If you find tables or indexes of this kind, carry out the following actions: Make sure that the value for the next extent (NEXT parameter) is increased. To do this, use the SAP utility program SAPDBA with the `next` option. If necessary, you can increase the value for MAXEXTENTS as described above (SAP Database Management: Oracle [Extern]).

Analyses with the CCMS Database Monitor

Tables/Index Analysis (Oracle) [Seite 382]

Extent Analysis (Oracle) [Seite 379]

See also:

Checking for Full Tablespaces (Oracle) [Seite 359]

Checking for Freespace Problems (Oracle) [Seite 363]

Checking Storage Parameters (Oracle) [Seite 364]

Problems with Maximum Number of Extents (Oracle) [Seite 366]

Checking for Freespace Problems (Oracle)

You can analyze the storage space situation using the Database Alert Monitor, the Database Monitor (Tablespace Analysis (Oracle) [Seite 380]) or program SAPDBA (SAP Database Management: Oracle [Extern]).

You will find below an example of freespace analysis using the Database Alert Monitor. Call this monitor and display freespace problems using the *Freespace management* display screen.

green	Indicates that no tablespace is in danger of running out of space at the time of the last database check This means that at least one additional extent can be assigned.
yellow or **red**	Indicates that one or more tablespaces have freespace problems.

Displaying Tablespaces by Available Freespace

To display the 20 tablespaces with the most urgent freespace problems, click the *Freespace management* field.

A graphic is displayed. The graphic shows the size of every tablespace, available freespace and the average growth of the tablespace per day. Tablespaces with space problems are shown in **yellow** or **red**. These tablespaces probably need additional freespace.

To display a forecast for available freespace, click a tablespace in the overview of the freespace problems. A graphic is displayed, which shows when, according to current trends, the freespace will be exhausted. If the entry *Extent allocation* appears in the *Alert* field, it means that there are objects in this tablespace which run the risk of an extent overflow. You can display the critical objects by clicking on this field.

See also:

Checking for Full Tablespaces (Oracle) [Seite 359]

Storage Management Errors (Oracle) [Seite 361]

Checking Storage Parameters (Oracle) [Seite 364]

Problems with Maximum Number of Extents (Oracle) [Seite 366]

Checking Storage Parameters (Oracle)

You should check whether there is enough freespace in the tablespaces in particular before transmitting mass data to the R/3 System. You can use the Database Monitors to find out whether additional extents - if required for the amount of data to be transferred - can be allocated. When you perform a storage space analysis, you should also check the storage parameters.

To analyze the storage parameters:

From the R/3 main screen, choose *Tools → Administration → Computing Center → Management System → Control → Performance Menu → Database → Tables / Indexes.*

Alternatively, use transaction code `DB02`.

Choose *Detail analysis menu → Parameter changes.*

You can now analyze the storage parameters for tablespaces or individual tables.

- Storage parameters for each tablespace
  - Tablespaces section: *Current sizes*, *Storage parameter*
- Storage parameters for tables of a tablespace
  - *Tables and Indexes* section: *Detailed analysis* (specify the tablespace), select the table, *Analyze → Extents*
  - *Tables and Indexes* section: *Detailed analysis* (specify the tablespace), select the table, *Analyze → Tables and its indexes*, *Storage parameter*
- Storage parameters for a table
  - *Tables and Indexes* section: *Detailed analysis* (specify the table), *Analyze → Extents*
  - *Tables and Indexes* section: *Detailed analysis* (specify the table), *Analyze → Tables and its indexes*, *Storage parameter*

Table: Storage parameters

Field	SQL parameters	Meaning
Pct. fre	`PCTFREE`	Percentage of memory of a data block that is kept free for possible changes to existing lines (default value is usually 10%).
Pct. use	`PCTUSED`	When a data block is full (except for the space for PCTFREE), no more new lines are inserted in it. Lines can only be inserted in this block again when the percentage of used memory falls below the value of PCTUSED (default value is usually 40%).
Init ext	`INITIAL`	Size of the first extent with which a table or index was created.
Next ext	`NEXT`	Size of the next extent that is assigned should a new extent be required.

Min Ext	`MINEXTENTS`	Initial number of extents when a table or index is created.
Max Ext	`MAXEXTENTS`	Upper limit for the number of extents of a table or index (default value is usually 100). MAXEXTENTS is a "soft limit " for the number of extents that can be allocated in a tablespace.
Pct. inc	`PCTINCREASE`	Percentage by which the size of the next extent is increased when each additional extent is assigned. SAP tables and indexes usually show a factor of zero. The size of the next extent remains constant, and as a result, problems with freespace are avoided.

If you are importing new data to the system, you should in particular monitor the number of extents allocated for tables that are growing rapidly to avoid reaching the value for the MAXEXTENTS storage parameter.

New master records are to be imported to the system.

To estimate the amount of data to be transferred, determine the average size of the master records to be transferred and multiply it by the approximate number of the master records. Generally, the master records are in tablespace `PSAPSTABD` and the indexes are in `PSAPSTABI`. Check whether the tablespace still has enough freespace.

Then determine the total number of additional extents necessary for the data. To calculate this, divide the quantity of the data by the expected value in the column *Next extent* of a representative table.

You can use this information to estimate whether the tablespace must be extended, the `NEXT` parameter changed or the `MAXEXTENTS` parameter increased for the tablespace.(SAP Database Management: Oracle [Extern]).

See also:

Checking for Full Tablespaces (Oracle) [Seite 359]

Checking for Freespace Problems (Oracle) [Seite 363]

Storage Management Errors (Oracle) [Seite 361]

Problems with Maximum Number of Extents (Oracle) [Seite 366]

Extent Analysis (Oracle) [Seite 379]

Problems with Maximum Number of Extents (Oracle)

You should check on a regular basis whether there are tables or indexes that are close to reaching their maximum number of extents (MAXEXTENTS parameter). This applies in particular when you are transferring mass data to the SAP system.

To display extents:

1. From the R/3 main screen, choose *Tools → Administration → Computing Center → Management System → Control → Performance Menu → Database → Tables / Indexes.*

   Alternatively, use transaction code `DB02`.

2. Choose *Checks* on the *Database Performance: Tables and Indexes* screen.
3. In the *Check for reorganizations* section, choose *Extents of tables and indexes*. All database objects with more than 10 extents are displayed.
4. Sort the tables and indexes on the display screen by the *Extents* column.

This quickly provides you with an overview of objects with a large number of extents. As the MAXEXTENTS parameter is also displayed, you can very quickly find out which objects are close to reaching the limit.

See also: Extent Analysis (Oracle) [Seite 379].

The MAXEXTENTS value for SAP objects is usually set to 100. This is a soft limit, which is sufficiently below the maximum value allowed by Oracle for MAXEXTENTS (usually 505 for a block size of 8 KB).Every table or index whose number of extents comes close to this limit may actually reach it during further database operation, resulting in a terminated transaction.

From Oracle release 7.3 there is no longer a hard limit for the number of events of a database object. For performance reasons, SAP recommends that you do not allow the number of extents to become too high.

If you find objects with a high number of extents, you can increase the MAXEXTENTS value for the table or index. You should also adjust the storage parameters for the size of the next extent (NEXT). If an object is close to reaching the maximum value allowed by Oracle for MAXEXTENTS, you must reorganize this object. For these operations, use the SAP utility program SAPDBA.

You should check the number of extents filled by tables and indexes particularly when you have completed a data transfer to the R/3 System. If it is not possible to use the Database Monitor to do this (for example, the R/3 System is not available after a new installation), you can use the analysis options provided by SAPDBA (Reference [Extern], Reference [Extern]).

See also:

Checking for Full Tablespaces (Oracle) [Seite 359]

Checking for Freespace Problems (Oracle) [Seite 363]

Checking Storage Parameters (Oracle) [Seite 364]

Storage Management Errors (Oracle) [Seite 361]

Extent Analysis (Oracle) [Seite 379]

Displaying the Oracle Table Statistics

If you use Oracle with the cost-based Optimizer, you should create statistics for the database tables on a regular basis. You should analyze tables using the auxiliary program SAPDBA and schedule them in the CCMS DBA scheduling calendar (transaction code DB13). Ensure regularly that tables or indexes were correctly analyzed.

To display data from the last table analysis:

1. From the R/3 main screen, choose *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Tables / Indexes*.

   Alternatively, enter transaction code `DB02`.

2. Choose *Checks*.

3. In the section *Cost based optimizer,* choose *Dates of table analysis*.

The initial screen then displays the following data:

1. The **init<SID>.ora** parameter **optimizer_mode**. Possible values are:

   SELECT: The cost-based optimizer is active.

   RULE: The rule-based optimizer is active.

2. Data from the last table analysis of program SAPDBA. You can use the function *SAPDBA logs* to display directly the corresponding SPDBA logs.

3. Statistics on how many tables were analyzed and at what time.

The function *All tables* displays detailed information about the last table analysis. For each database table the system displays the date of the last analysis **(dba_tab_coumns.last_analyzed)**, the number of lines in a table as determined by the last analysis **(dba_tables.num_rows)**, as well as the sample size used to determine the last statistics **(dba_tab_coumns.sample_size/dba_tables.num_rows).**

You can find further information on this in your Oracle documentation under:

- Cost based optimizer
- ANALYZE TABLE
- DBA_TABLES, DBA_TAB_COLUMNS

as well as in the R/3 Extended Help (SAP Oracle Database Administration) for the auxiliary program SAPDBA.

See also:

SAP Database Management: Oracle [Extern]

Monitoring the Shared Pool (Oracle) [Seite 388]

SAP/Oracle Database Monitor: Status of the Data

To display the status of the data:

From the R/3 main screen, choose *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Tables / Indexes*.

Alternatively, use transaction code `DB02`.

The information on the *Database Performance: Tables and Indexes* screen is subdivided into various sections. The most important display screens are explained below:

Database system

In this section, you will find general information on the database:

- the name of the database
- the time when the details of this screen were generated

See also: Data for the Screen: Database Performance: Tables and Indexes [Seite 422].

The table shows the analysis functions available.

Table: Database system analysis function

Pushbutton	Explanation
Refresh	Updates the statistics on the entire screen. Choose this option only if absolutely necessary, as it can take a long time to determine all the information depending on the size of the database.
Check	Consistency Checks [Seite 370], Extent Analysis (Oracle) [Seite 379]
Space statistics	Displays the database history.

Tablespaces

In this section, you will find overview information on all tablespaces of the SAP system: number, size, freespace, information on freespace problems.

The table shows the analysis functions available.

Table: Tablespaces analysis functions

Pushbutton	Explanation
Current sizes	Tablespace analysis
Space statistics	Displays tablespace history
Freespace statistics	Freespace analysis

See also:

Tablespace Analysis (Oracle) [Seite 380]

Tables and indexes

In this section, you will find overview information on all tables/indexes of the SAP system: number, size, number of objects with more than one extent, number of objects missing in the database, number of objects missing in the ABAP Dictionary, number of objects with freespace problems.

The table shows the analysis functions available.

Table: Tables and indexes analysis functions

Pushbutton	Explanation
Detailed analysis	Analysis of individual objects
Missing indexes	Missing Indexes [Seite 384]
Space critical objects	Displays storage-critical objects
Space statistics	Displays the history of tables/indexes

See also:

Tables/Index Analysis (Oracle) [Seite 382]

Consistency Checks

Three different functions are provided for checking consistency between the ABAP Dictionary and the database:

- **Missing indexes**

  Displays indexes that are not known in the database or ABAP Dictionary.

- **Database - ABAP Dictionary consistency**

  The existence of *all* database objects defined in the ABAP Dictionary is checked.

- **Database tables without a unique index**

  Displays database tables without a unique index.

To perform the consistency checks:

1. From the R/3 main screen, choose *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Tables / Indexes*.

   Alternatively, use transaction code `DB02`.

2. Display missing indexes with *Missing indexes*.
   Use the *Checks* function to perform the consistency checks described above.

The results are displayed in a hierarchy:

Table: Results in the hierarchy

hierarchy level	result
top	distinction between the objects defined in the ABAP Dictionary and the objects defined in the database
middle	subdivision by object type, for example by table, view or index, and the number of missing objects is displayed in each case
lowest	displaying the names of the individual objects

Missing Indexes [Seite 384]

Database - ABAP Dictionary Consistency [Seite 371]

Database Tables without a Unique Index [Seite 374]

Creating Objects in the Database [Seite 375]

Displaying Object Definitions [Seite 376]

Naming Conventions for Indexes [Seite 377]

Database - ABAP Dictionary Consistency

With this function, you can find all the database objects that are defined in the ABAP Dictionary but have not been created in the database (or were deleted). This function also displays objects that were created *directly* in the database and are therefore unknown in the ABAP Dictionary.

To perform the consistency check:

1. From the R/3 main screen, choose Tools → *Administration* → *Computing Center* → Management System → Control → Performance Menu → *Database* → *Tables / Indexes* → *Checks* → *Database <->ABAP Dictionary consistency.*

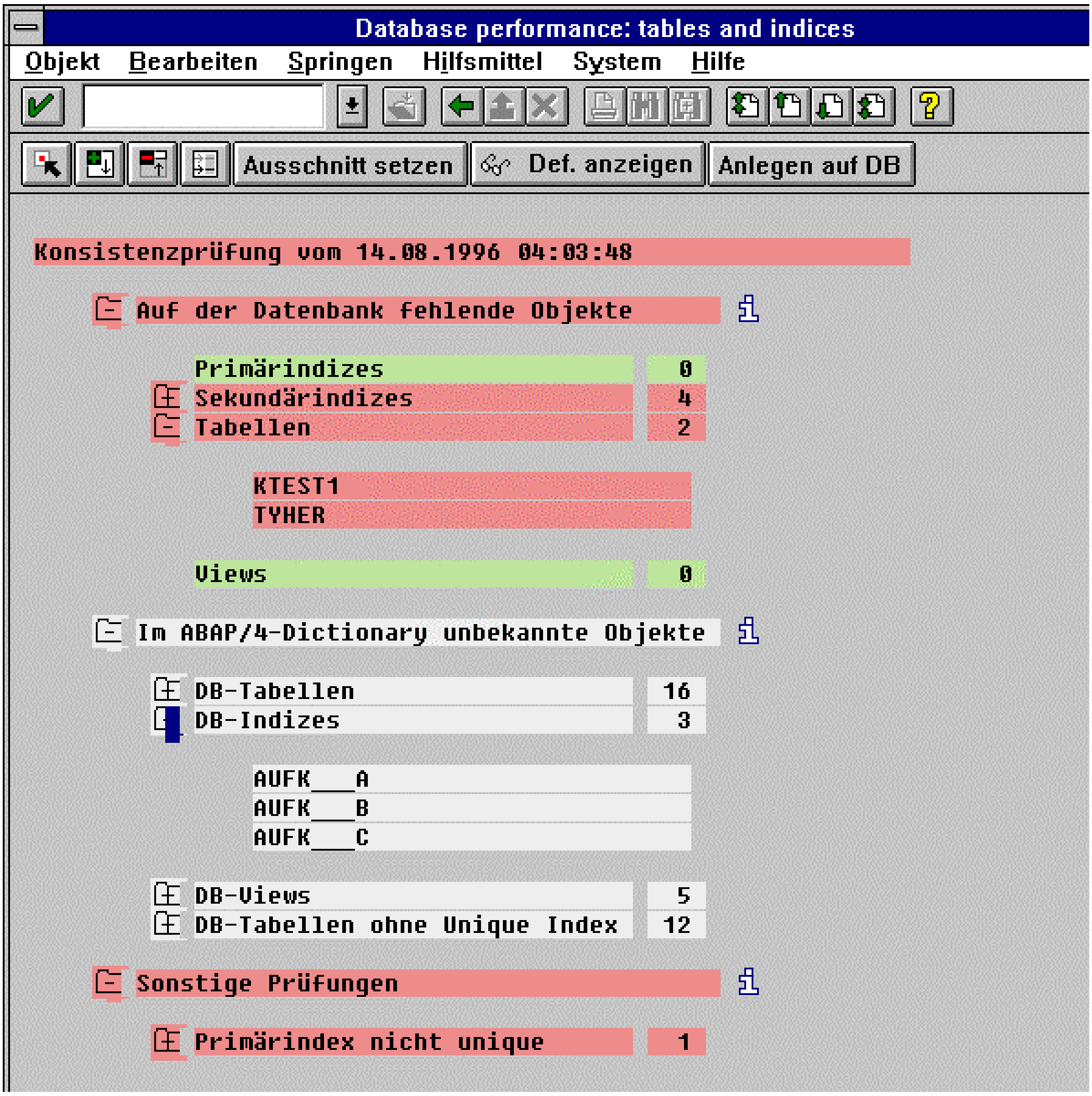
   Alternatively, use transaction code DB02.

2. Choose *Checks* → *Database <-> ABAP Dictionary consistency*.

When you choose this function, the date of the last check is displayed in a second window. You can now choose whether you want to see the result of the last check (from the performance database), or whether you want to start a new check online. In the latter case, you can expect a wait time of a few minutes (depending on system load). The online check updates the relevant data in the performance database.

The inconsistencies found are displayed in a hierarchy:

Database - ABAP Dictionary Consistency



Use *Display def.* to branch directly to the ABAP Dictionary where you can look at the object definitions. With *Create in DB*, you can create the missing objects directly in the database.

In most cases, you can perform a closer analysis of objects unknown to the ABAP Dictionary using *Display def.*, their database definitions are then displayed.

See also:

Creating Objects in the Database [Seite 375]

Displaying Object Definitions [Seite 376]

Database <-> ABAP Dictionary consistency only checks the existence of objects. A precise comparison of the objects would be too expensive.

If the checks show that objects are missing, you should first find out whether they are test objects (these should not be in the database). This is the most likely explanation for an inconsistency.

Database Tables without a Unique Index

This function checks whether the tables defined in the ABAP Dictionary have a primary index and whether it was created with the “unique” option. (The existence of a primary key constraint is sufficient for some databases. This is also taken into account here.)

To perform the consistency check:

1. From the R/3 main screen, choose *Tools* → *Administration* → *Computing Center* → *Management System* → *Control* → *Performance Menu* → *Database* → *Tables / Indexes* → *Checks* → `Database tables without unique indexes`.

   Alternatively, use transaction code `DB02`.

2. Choose *Checks* → `Database tables without unique indexes`.

This online check also displays tables that were created directly in the database and have no unique index. Whether a unique index is required for these tables depends on the purpose of their use.

An unique index ensures that no duplicates are entered in a table. If primary indexes are missing for ABAP Dictionary tables, it is essential that you create them with *Create in DB*. To correct primary indexes created without the unique option, you should go to the ABAP Dictionary via *Display def.* From here, you can branch to the *Database Utility* via *Utilities*.

If duplicate keys have already been inserted in the table, it is no longer possible to create the index. You must identify the incorrect keys and delete them. In difficult cases, contact SAP regarding the restoration of the index.

Normally, a table should only be defined via the ABAP Dictionary.

See also:

Naming Conventions for Indexes [Seite 377]

Creating Objects in the Database

Use

Objects that are missing in the database can be created directly from the display providing you have the necessary authorization (see BC Users and Authorizations [Extern]).

Procedure

Choose the object and then choose *Create in DB*. You can now choose whether you want to create the object directly (online) or in the background (as a batch job). (The latter option is only useful for indexes of large tables.)

You also have the option of creating objects using the database utility.

If an error occurs when you create the object, the system displays a log with information on the error.

The object is displayed until a new check is performed. The display only shows a “snapshot” at the time of the check.

Displaying Object Definitions

If you want to analyze an object more closely, choose the object and *Display def*. f

If the object is defined in the ABAP Dictionary, you will navigate directly to the ABAP Dictionary display for this object. All other navigation options are available from here. In particular, you can call the Database Utility in order to create the object in the database, or you can use the analysis options provided by the Database Utilities.

It is possible to display the database definition for objects that were created directly in the database. This can be useful, for example, to find out more on the purpose or author of an object.

For technical reasons, this function is currently only possible for objects with names of up to ten digits. For a closer examination of objects with longer names, you must use the database utilities.

Naming Conventions for Indexes

Indexes are identified in the ABAP Dictionary via the table name and an index ID of up to three digits. Together they make up a unique index name when you create the index in the database. The possibilities for index names are as follows:

- From Release 4.0, the index name is composed of the table name, a separator and an index ID. A tilde (~) is usually used as the separator. Some tables may require an alternative index name for the upgrade. In such cases, the separator ^‘ is used.

  Table name TAB456789 and the index ID "A2" make up the index name TAB456789~A2 in the database.

  Table names with a maximum of 18 digits are used. For tables with names 15 and 16 digits long, only the first two digits are relevant to the index ID, as only these are used for the index name in the database. If a name has 16 digits and a two-digit index ID, the separator is left out.

  Table name TAB4567890123456 and the index ID "A23“ therefore make up the index name TAB4567890123456A2 in the database.

- As it is not possible to rename indexes (in most database systems), the index names are retained from older releases, to avoid having to convert the indexes. Therefore you can still find the following naming convention in a system which originates from a release upgrade:

- A ten-digit table name followed by the one to three-digit index ID.

  Shorter table names are padded to ten digits with underscores (‘_’).

  From the table name TAB456789 and the index ID “A2“, you get the index name TAB456789_A2 in the database.

- A ten-digit table name followed by a three-digit index ID followed by the character “X”.

  Shorter table names are padded to ten digits with underscores. Likewise one-digit and two-digit index IDs are padded to three places with underscores. These alternative names are necessary for particular Basis tables because of the upgrade procedure technology. New indexes are then only created according to this naming convention if all other indexes of the table follow this convention.

  From the table name TAB456789 and the index ID “A2“; you get the index name TAB456789_A2_X in the database.

- A seven-digit table name followed by a one-digit index ID.

  Longer table names are truncated to seven digits; shorter table names are padded to seven digits with underscores. This naming convention is only found for indexes that were created in the database before Release 3.0. When you delete and recreate this kind of index, it immediately follows one of the first two naming conventions. (Before Release 3.0, the index ID consisted of one digit and table names often had seven digits.)

Extent Analysis (Oracle)

To analyze the extent structure:

1. From the R/3 main screen, choose *Tools → Administration → Computing Center → Management System → Control → Performance Menu → Database → Tables / Indexes* the Database system

   Alternatively, use transaction code `DB02`

2. Choose Checks in the database system section

   The *Check for reorganization* section provides the following options:

Extents of tables and indexes

– Displays all objects with more than 10 extents. Alongside the size of the objects (in Kilobytes and blocks), the number of used extents and the value defined for the object for MAXEXTENTS are displayed. The DBA should check these details on a regular basis to avoid possible Problems with Maximum Number of Extents (Oracle) [Seite 366]. The value for the NEXT storage parameter is also listed. You can, if you wish, start a detailed storage space analysis as a background job.

– Extents per tablespace

  You can analyze the extent structure for all tablespaces. In addition to this list output, a second overview displays all tables and indexes with more than 4 extents. Alongside the size of the objects (in Kilobytes and blocks), the number of used extents and the value defined for the object for MAXEXTENTS are also displayed. You can, if you wish, start a detailed storage space analysis as a background job.

– Check next extent size

  Displays all objects that have exhibited critical growth within the last four weeks. This allows you to trace the growth in number and size of extents. You can immediately see the size of the first extent and the NEXT value defined for the object. You will also find details on how close the number of extents has come to reaching the limit for the number of extents set in the MAXEXTENTS parameter.

You can display a detailed history of an object.

See also:

Tablespace Analysis (Oracle) [Seite 380]

Tables/Index Analysis (Oracle) [Seite 382]

Checking for Full Tablespaces (Oracle) [Seite 359]

Storage Management Errors (Oracle) [Seite 361]

Checking Storage Parameters (Oracle) [Seite 364]

Tablespace Analysis (Oracle)

To analyze the tablespaces:

From the R/3 main screen, choose *Tools → Administration → Computing Center → Management System → Control → Performance Menu → Database → Tables / Indexes.*

Alternatively, use transaction code `DB02`.

In the Tablespaces section there are extensive options available for analyzing tablespaces. Some of these options are described below:

- ***Current sizes***

  You get a complete list of tablespaces with details on:

  – size

  – freespace

  – used space

  – number of objects

  – number of extents

  You can *Sort* the tablespaces according to a particular feature. For example, you can quickly find out which tablespaces are almost full.

  See also: Checking for Full Tablespaces (Oracle) [Seite 359]

  Choose *Storage parameter* to display the storage parameters of a selected tablespace.

  Choose *Analysis* to examine the selected tablespace in more detail.

  – *Tables and indexes*: displays the objects in this tablespace. Alongside the size of the objects (in Kilobytes and blocks), the number of used extents and the value defined for the object for MAXEXTENTS are also displayed. The DBA should check these details on a regular basis to avoid possible Problems with Maximum Number of Extents (Oracle) [Seite 366]. You can, if you wish, start a detailed storage analysis as a background job..

  – *Files*: displays the files which make up the tablespace. The assignment of files to individual disks is relevant to the DBA, as they may influence performance.

  – *Detail Analysis*: you can, if you wish, start a detailed storage space analysis as a background job.

  – *Statistics*: displays the history of a tablespace. The DBA can trace the growth of a tablespace over a particular time period. Changes to the size, freespace or number of extents of a tablespace, for example, are recorded.

  – *Freespace analysis*: displays a freespace analysis organized by the files of a tablespace. Freespace (in Kilobytes) and number of free blocks are displayed.

  See also: Checking for Freespace Problems (Oracle) [Seite 363]

- ***Space statistics***

  Displays a history of all tablespaces.

  Choose *Choose* to display a detailed history of an individual tablespace.

- ***Freespace statistics***

  You get a breakdown of the freespace situation for all tablespaces. Alongside the total freespace available, the largest freespace area (*Freespace Maximum*) and the number of fragments are also displayed. This provides you with a good overview of the fragmentation of tablespaces. The size of the largest extent is also displayed. The DBA can then assess whether the next extent (if required) can be inserted in the freespace of the tablespace.

  Choose *Critical tables/indexes* to display the critical objects for the storage situation.

See also:

Storage Management Errors (Oracle) [Seite 361]

Checking Storage Parameters (Oracle) [Seite 364]

Monitoring Table and Index Fragmentation (Oracle) [Seite 400]

Tables/Index Analysis (Oracle) [Seite 382]

Extent Analysis (Oracle) [Seite 379]

Tables/Index Analysis (Oracle)

To analyze tables and indexes:

From the R/3 main menu, choose *Tools → Administration → Computing Center → Management System → Control → Performance Menu → Database → Tables / Indexes.*

Alternatively, use transaction code `DB02`.

In the *Tables and Indexes* section there are extensive options available for analyzing tables/indexes. Some of these options are described below:

- ***Detailed analysis***

  Limit the number of tables to be analyzed. If, for example, you only enter the name of the tablespace as the selection criterion, the analysis can take a very long time depending on the number of objects contained in the tablespace.

  Choose *Analysis* to examine one of the tables listed in more detail.

  – *Tables and its indexes*: displays the table and the indexes defined for the table. Alongside the size of the objects (in Kilobytes and blocks), the number of used extents and the value defined for the object for MAXEXTENTS are also displayed. The DBA should check these details on a regular basis to avoid any possible Problems with Maximum Number of Extents (Oracle) [Seite 366]. You can use the *Storage Parameter* pushbutton to display additional storage parameters for the individual objects.

  – *Extents*: here you will find the size (in Kilobytes and blocks) of the extents of the table, and their location (file ID, block number).

  – *Detail Analysis*:.

  – *History*: displays the history of a table. The DBA can trace the growth of a table over a particular time period. Changes to size and extent assignment, for example, are recorded. The value for the NEXT storage parameter is also listed. This helps the DBA to monitor the situation in storage-critical tablespaces, that is, to determine whether a next extent of this size will fit into the freespace of a tablespace.

  – *Table columns*: displays the structure of the table in the SAP ABAP Dictionary and in the database.

- ***Missing indexes***

  See Missing Indexes [Seite 384]

- ***Space critical objects***

  Displays critical objects for the storage space situation.

- ***Space statistics***

  Displays the history of tables/indexes. Limit the number of tables/indexes to be analyzed as best you can. If, for example, you only enter the name of the tablespace as the selection criterion, the analysis can take a very long time depending on the number of objects contained in the tablespace.

  The DBA can trace the growth of a table over a particular time period. Changes to size and extent assignment, for example, are recorded. The value for the NEXT storage parameter is also listed. This helps the DBA to monitor the situation in storage-critical tablespaces, that is, to determine whether a next extent of this size will fit into the freespace of a tablespace.

See also:

Checking for Freespace Problems (Oracle) [Seite 363]

Storage Management Errors (Oracle) [Seite 361]

Checking Storage Parameters (Oracle) [Seite 364]

Monitoring Table and Index Fragmentation (Oracle) [Seite 400]

Tablespace Analysis (Oracle) [Seite 380]

Extent Analysis (Oracle) [Seite 379]

Missing Indexes

Indexes which are defined in the ABAP Dictionary but are missing in the database or indexes which were created in the database but are unknown to the ABAP Dictionary are an especially important factor in performance problems. For this reason, there is a separate *Missing indexes* display, even though one is already included in the full check accessed via *Database* <-> *ABAP Dictionary consistency*.

Incorrectly defined and superfluous indexes may also impair database performance. They can cause the database optimizer to make an inefficient index selection. Moreover, whenever the database is updated, the superfluous indexes also have to be taken into account.

To check whether indexes are missing:

1. From the R/3 main screen, choose Tools → *Administration* → *Computing Center* → Management System → Control → Performance Menu → *Database* → *Tables / Indexes* → *Missing indexes*.

   Alternatively, use transaction code DB02.

2. Choose *Missing indexes*.

The tables in the ABAP Dictionary and the database tables are analyzed. The data displayed either comes from the regular batch runs for performance analysis, or it is created/updated with *Refresh*.

Missing indexes may occur if you ignore an error message when creating a table (table created, index not created) or if an index is deleted. The latter case may occur during an incorrect reorganization.

Indexes that are defined in the ABAP Dictionary but are missing in the database can be created in the database directly from the display (Creating Objects in the Database [Seite 375]). You can also display the respective definition in the ABAP Dictionary (Displaying Object Definitions [Seite 376]).

Primary indexes (ending with 0) ensure that the line keys (row keys) are unique. Missing primary indexes are therefore a critical problem.

Secondary indexes (ending with 0) are used for particular scans and are only important for performance.

See also:

Consistency Checks [Seite 370]

Database Tables without a Unique Index [Seite 374]

Naming Conventions for Indexes [Seite 377]

SAP/Oracle Performance Monitoring Strategies

Monitoring the Data Buffer (Oracle)

The database buffer cache (also known as the data buffer or Oracle data buffer) is the area of the System Global Area (SGA) used to hold copies of data blocks read from the disk. Oracle user processes cannot read data directly from data files, which is why all data must first be read into this buffer cache.

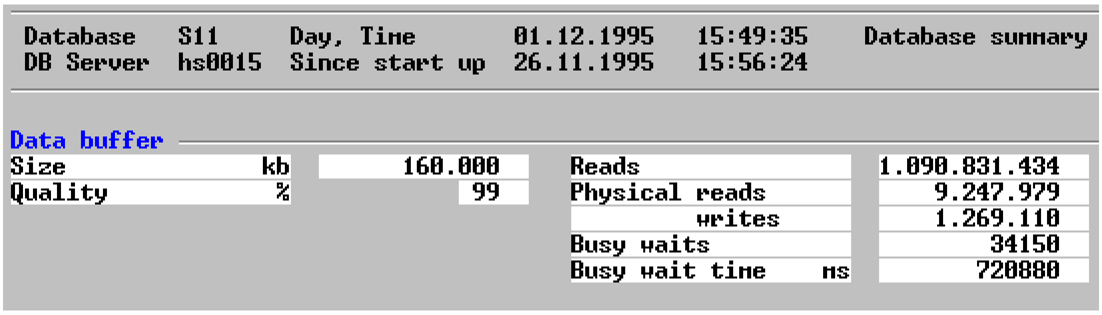
When a user process requests a data block which is already in the data buffer, it can be read without having to access the disk again (providing the block has not been changed since it was last read into the buffer). This saves considerable processing time. In this situation, the user process has made a “hit” on that data block. When a user process requests a data block which is not in the data buffer, this is called a “miss”. The relationship between hits and misses is known as the “hit ratio” Hit ratio can also be thought of as the “quality” of the database buffer cache.

To display information on the data buffer:

From the R/3 main screen, choose *Tools → Administration → Computing Center → Management System → Control → Performance Menu → Database → Activity.*

Alternatively, use transaction code **ST04.**

The following data is displayed:



Database	S11	Day, Time	01.12.1995	15:49:35	Database summary	
DB Server	hs0015	Since start up	26.11.1995	15:56:24		

Data buffer

Size	kb	160.000	Reads	1.090.831.434
Quality	%	99	Physical reads	9.247.979
			writes	1.269.110
			Busy waits	34150
			Busy wait time ms	720880

The following data buffer information is displayed in the section *Data buffer*:

- The size of this data buffer is 160 MB
- The overall quality of the data buffer is 99%
- There have been 1,090,831,434 total Oracle blocks read from the data buffer since database instance startup
- Of these total reads, 9,247,979 reads have resulted in blocks being physically read from disk
- 1,269,110 Oracle blocks have been written to disk by the Database Writer process
- There was a total of 34,150 waits when accessing blocks of various classes in the data buffer
- The total wait time was 720.880 milliseconds

Data Buffer Size

Data buffer size is determined by the product of the block size (DB_BLOCK_SIZE (Oracle) [Seite 408]) and the number of database block buffers specified in the **init<SID>.ora** parameter file (DB_BLOCK_BUFFERS (Oracle) [Seite 407]). SAP uses a default **db_block_size** of 8192

Bytes for most Oracle databases. Once the database is created, this value cannot be changed. The value for `db_block_buffers` can however be changed as required.

Data Buffer Quality

SAP recommends that you maintain a data buffer quality of at least 97% on a production R/3 System.

If the database instance has just been started, the hit ratios shown may be somewhat misleading. A database should be "warmed up" before you look at hit ratios.

Read and Write Operations (Reads, Physical reads/writes)

Statistics for read and write operations let you quickly determine the level of activity of a database since instance startup. If the number of physical writes is on the same scale as the number of physical reads, you should also monitor the activities of the database writer, the rollback activities and the redo log activity. This situation may occur particularly in online transaction environments when there are many updates of individual tables lines.

Wait Situations (Busy waits, Busy wait time)

A wait situation in a buffer occurs when an Oracle process attempts to access a block that is still in an inconsistent status. The number of wait situations displayed on the main screen is the average number for all Oracle block classes. There are a number of Oracle block classes that may play a part in the occurrence of wait situations, but only four are commonly found when monitoring the SAP system. These are: data block, segment header, undo header and undo block.

If the total number of wait situations exceeds 5% of the total number of reads, you should analyze the situation more closely. In the Detail Analysis Menu (Oracle) [Seite 341], choose the Buffer Busy Waits (Oracle) [Seite 343].pushbutton. This gives you a breakdown of wait situations.

Name of operation	Class	Range(ms)	Number	Time(ms)	Avg.(ms)
waiting for	data block	N/A	32.758	640.190	20
waiting for	segment header	N/A	228	12.030	53
waiting for	undo header	N/A	883	67.350	76
waiting for	undo block	N/A	290	1.350	5
Totals			34.159	720.920	21

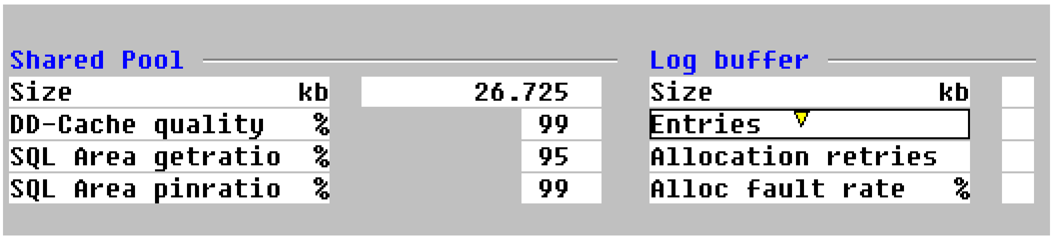
If the number of waits on any one of the block classes specified exceeds 1% of the reads, this might indicate excessive contention for this class.

Waits on the undo header and undo block classes can be reduced by adding more rollback segments to the database. Waits for data blocks may be due to the data buffer size not being large enough (check quality ratio above). Waits on segment headers often indicate contention for freelists. For more information, refer to the relevant Oracle documentation.

Monitoring the Shared Pool (Oracle)

The shared pool is the area of the System Global Area (SGA) that contains structures such as the data dictionary cache and the shared SQL area. This is one of the most important storage structures in an Oracle database system.

The Database Monitor displays the following information on the shared pool:



Size of the Shared Pool

The size of the shared pool is specified in Kilobytes. For a productive system, this value should not be less than 50 MB. Depending on the system workload, it may be necessary to increase this value (taking into account the total amount of storage space available). This size of the shared pool is controlled by the `init<SID>.ora` parameter `shared_pool_size` (SHARED_POOL_SIZE (Oracle) [Seite 414]). Note that you must restart the database instance for the change to this parameter to take effect.

Data Dictionary Cache Quality (DD Cache quality)

The data dictionary cache holds information needed by Oracle administrators, users and the Oracle database itself. Since the data dictionary is accessed often, it is best to retain as much of this information as possible in the SGA. The data dictionary cache quality statistics show the overall average hit ratio for the various Oracle dictionary caches. This value should ideally be above 90% for production systems.

The data dictionary cache will be empty when Oracle is started and will fill with use. For this reason, it is not practical to examine these statistics until the database has reached its normal operating activity.

Shared SQL Area (SQL Area getratio/pinratio)

A shared SQL area (or shared cursor cache) is an area in the shared pool which contains the parse tree and execution schedule for an individual SQL statement. Shared SQL areas are shared by identical SQL statements.

The values under *SQL Area get/pinratio* measure the success rate for accessing SQL statements in the Oracle shared SQL area. The ability to reuse identical SQL statements greatly reduces the work load associated with parsing and loading statements into working memory. Reusing identical SQL statements not only improves the transaction response time, but also allows for more efficient space management within the shared pool since fewer parsed statements are moved in and out of the shared SQL area.

Most important here is the pinratio, which should be close to 99%. SAP recommends Note that SQL statements must be parsed when executing transactions for the first time after database instance startup. This results in a low shared SQL area cache quality, which should however improve over time. If these figures remain low when normal activity level is reached, you should check the text of the SQL statements in the shared SQL area (SQL Request (Shared SQL Area) [Seite 349]). Determine whether some of the statements can be re-coded for common use. If this is not possible, increase the value of the `init<SID>.ora` parameter `shared_pool_size` (SHARED_POOL_SIZE (Oracle) [Seite 414]).

Monitoring the Redo Log Buffer (Oracle)

The redo log buffer is the part of the System Global Area (SGA) that holds information about changes made to the database. Each of these changes generates a ‘redo entry’. Redo entries are needed to reconstruct these changes during the recovery process.

The Database Monitor displays the following information on the redo log buffer:

Shared Pool			Log buffer		
Size	kb	26.725	Size	kb	320
DD-Cache quality	%	97	Entries		63.528
SQL Area getratio	%	95	Allocation retries		2
SQL Area pinratio	%	99	Alloc fault rate	%	0

Size of the Redo Log Buffer (Size)

The default size for the Oracle redo log buffer on SAP systems is 320 KB. It is set by the init.ora initialization parameter `log_buffer` (LOG_BUFFER (Oracle) [Seite 411]). This setting is normally adequate for most installations.

Allocation Retries (Allocation retries/Alloc fault rate)

Allocation retries shows the number of failed attempts to allocate space in the redo log buffer. A value greater than zero normally indicates that the Oracle log writer process (LGWR) could not write redo entries from the buffer to disk (in the online redo log files) immediately, but had to wait for a redo log file switch to perform this action.

If the number of allocation retries constantly increases during normal database operation, you may need to increase the redo log buffer.

Note the following: Larger online redo log files reduce the number of redo log file switches since more data is stored in each file. If you then perform high volume insert operations (common in table reorganization or data loading), large amounts of redo entries will be generated making allocation errors more likely.

Alloc fault rate shows the ratio of allocation retries to total redo entries in the redo log buffer since database instance startup. Error rates of more than 1% under normal operating conditions should be investigated.

Monitoring Calls (Oracle)

The total number of calls made to the Oracle kernel since database instance startup is recorded. In a busy production system, the value will be high. Any reduction in the number of calls sent to the kernel will ease the load put on the database system.

The Database Monitor displays the following information on *Calls*:

Calls			
User calls	257.599	Recursive calls	202.679
commits	10.355	Parses	9.134
rollbacks	1		

Commits

Commits is the total number of committed transactions since database instance startup.

Rollbacks

Rollbacks indicate the total number of transactions that were rolled back since the database system was started. These rollbacks could be caused by failing programs, application deadlocks or abnormal application termination. If a high number of rollbacks are reported, you should check the database ALERT file (Database Message Log (Oracle) [Seite 352]) and the trace files for possible problems.

- The Oracle database ALERT file and trace files for the Oracle background processes are usually found in: `/oracle/<SID>/saptrace/background`
- Trace files for Oracle user processes are found in: `/oracle/<SID>/saptrace/usertrace`

Recursive Calls

Recursive calls occur when Oracle itself must issue a SQL statement in addition to the SQL statement issued by a user process. The most common causes of recursive calls are:

- *Misses* in the data dictionary cache (Monitoring the Data Buffer (Oracle) [Seite 386])
- Dynamic storage extension
- Execution of DDL statements, the enforcement of referential integrity constraints, use of PL/SQL (refer to Oracle documentation for more information)

Recursive calls can impair the performance of the database system and should be minimized when possible.

The recursive call ratio is calculated as *Recursive calls/User calls*. If the number of *Recursive calls* is greater than the number of *User calls,* then you should start a detailed examination. Check the data dictionary cache hit ratio and average parse ratio. Increasing `shared_pool_size` (SHARED_POOL_SIZE (Oracle) [Seite 414]) should help. Ensure that the `init<SID>.ora` parameter `row_cache_cursors` (ROW_CACHE_CURSORS (Oracle) [Seite 413]) is set to at least the SAP recommended minimum of 100.

Dynamic storage extension occurs when a database object (a table or index) must extend beyond its allocated space (that is, a new extent is allocated). SAP recommends that you always create the original extent (INITIAL parameter) and the following extents (NEXT parameter) large enough to minimize dynamic extent assignment. It is only possible to change the INITIAL storage parameter for an object through a reorganization. You should adjust the NEXT parameter to SAP requirements on a regular basis using the SAPDBA option `next` (SAP Databse Management [Extern]).

A table reorganization is generally not necessary. However, an index reorganization can sometimes prove helpful.

As in the case of the cache hit ratios, the value for recursive calls will be high after database instance startup. Since the data dictionary cache is at first empty, all calls needed to load information into working memory will be recursive.

Parses

Parses shows the total number of times an SQL statement was parsed (for information on the term “parses”, refer to the Oracle documentation). To calculate the average parse ratio, you divide *parses* by *user calls*. If this ratio is above 25%, there may be a problem with retaining cursors in the shared cursor cache (SQL Request (Shared SQL Area) [Seite 349]). Check the hit rates discussed in the shared SQL area statistics (Monitoring the Shared Pool (Oracle) [Seite 388]). It may be necessary to increase `shared_pool_size` (SHARED_POOL_SIZE (Oracle) [Seite 414]).

Reads / User calls

The value Reads / User calls displays the number of Oracle blocks on average that were read from the data buffer to satisfy a request (call) sent to the database. If this value is greater than 30, this indicates expensive SQL statements. You should, therefore, begin to examine the shared SQL area.

See also:

Monitoring the Shared SQL Area (Oracle) [Seite 398]

SQL Request (Shared SQL Area) [Seite 349]

Monitoring Table Access Methods (Oracle)

A full table scan occurs when a user process queries data from the database table without the use of an index. The entire table must be read to retrieve the requested information. Sometimes this is desirable, for example if the table is only short. Often, it is more efficient to use an index.

The Database Monitor displays the following information on the table accesses:

Table scans		Table fetch	
Short tables	12.894	By rowid	1.587.204
Long tables	4.493	Continued row	10.164
Rows gotten	11.834.786		
Blocks gotten	1.583.445		

Short Tables

Short tables shows the total number of full table scans that were performed on short tables (tables having less than 5 Oracle data blocks). It is generally more efficient to perform full table scans on short tables rather than access the data using indexes.

Long Tables

Long tables shows the total number of full table scans done on long tables (tables containing 5 or more Oracle data blocks). It is usually advantageous to access long tables using indexes.

The sum of the values for *Short tables* and *Long tables* gives the total number of full table scans performed since database instance startup.

A high number of full table scans on long tables might be an indication that table indexes are missing or should be created. You can check whether indexes are missing using the functions of the *Database Performance: Tables and Indexes* screen (Missing Indexes [Seite 384]).

Use *Explain one SQL request* of the SQL trace to examine the optimizer access path of expensive statements (use the SQL Request (Shared SQL Area) [Seite 349]). Determine whether adding a new index or reordering an existing one may be beneficial (Table Scans: Problem Analysis (Oracle) [Seite 357]).

Table Fetch by ROWID (By rowid)

By rowid in the *Table Fetch* section shows the number of lines that were accessed either by index lookup or by specifying a distinct line ID (ROWID) in an SQL statement. High values for this entry indicate heavy use of indexes. This is generally a good sign, though you should examine whether non-selective indexes used in range scans substantially add to this number. Indexes should be made as selective as possible. The index fields should always be arranged thus that the most commonly accessed table fields come first. If more than 20% of the lines in a table are output for a selection, it is advisable to perform a full table scan (that is, do not create an index for this query).

Chained Data Records (Continued row)

Continued Row shows how often the database system has accessed chained data records. Chained data records are lines that are distributed across several data blocks.

Monitoring Table Access Methods (Oracle)

Chained data records lead to an increase in search time as the database system has to read several blocks to merge the data record. This means that additional I/O operations are required. For these reasons, data chaining should be avoided whenever possible.

When accessing tables with fields that use the “long“ Oracle data type, chaining is often unavoidable since the line may be too long to fit in one data block. If the ratio *Table fetch Continued row / Table fetch By rowid* is greater than 1:1000, you should perform a more detailed analysis. Identify the objects with chained data records using, for example, the SAPDBA option `analyze`. You might need to eliminate chaining through a reorganization. Use the SAP utility program SAPDBA (SAP Database Management: Oracle [Extern]).

Monitoring Sorting (Oracle)

Sorting is done in Oracle for SQL statements that use ORDER BY, GROUP BY and SORT MERGE JOIN operations, and for index creation.

Sorts	
Memory	422.991
Disk	858
Rows	18.723.241

Memory Sorts

Memory gives the number of sorts performed in memory. Sorts performed in memory are generally much faster than those done on disk. The init<SID>.ora parameter sort_area_size (SORT_AREA_SIZE (Oracle) [Seite 415]) determines the amount of process memory which can be used for sort operations. The memory space is allocated for each process, so you must ensure that there is sufficient operating system memory to accommodate an increase in this value. Otherwise, unnecessary paging or swapping may occur.

Disk Sorts

Disk gives the number of sorts that had to be written to temporary segments on disk in order to be sorted. SAP uses the tablespace PSAPTEMP to hold these segments.

The *Disk* to *Memory* ratio should be no more than 5%. If it is higher, investigate increasing the sort_area_size parameter.

Diagnosing SAP/Oracle Performance Problems

A Transaction is Running Very Slowly (Oracle)

As a database administrator, end users will often ask you to investigate why a particular transaction or set of transactions are running slow. There are many factors to take into account when tracking down these types of problems. As this search may take a considerable amount of your time, you should gather as much background information as possible from the responsible parties before you begin.

Questions you should ask the end users include:

- Has the transaction always been slow or did you only recently notice the slowdown?
- Is it a new program or transaction?
- Is the slowdown only during peak periods or is it fairly constant?
- Has the user workload changed recently?
- Does it appear that just this one transaction is slow or are other transactions/applications also now performing poorly?

Having gathered this information, the DBA can try to identify where the performance bottleneck resides. Keep in mind that performance tuning is an iterative process and you will probably have to involve the end users at some point to determine whether the tuning steps you have taken have alleviated their problems.

If you feel this issue may be isolated to one particular transaction, program or application, you may also need assistance from the application developers. They will better understand the process flow of the application and can help to change and test statements in the program. It is not unusual for a program to function correctly in a development environment and then to perform poorly in production. A production system frequently has more data and more users than a test system.

There are three major areas which you should check when the above symptoms are reported.

- Check for poorly programmed SQL statements which are utilizing a disproportionate amount of system resources, sometimes causing other transactions to slow down as well.

  Monitoring the shared SQL area (Oracle)Monitoring the Shared SQL Area (Oracle) [Seite 398]

- Find out whether exclusive lockwait situations have occurred, in which one or more processes are competing for the same object.

  Exclusive Lockwaits (Oracle) [Seite 351]

Monitoring the Shared SQL Area (Oracle)

Purpose

Poorly written SQL statements have perhaps the greatest impact on application performance. An SQL statement with which the Oracle database system reads and/or sorts thousands or even millions of rows of data can bring the database to a standstill. Proper use of indexes is vital to prevent such situations from occurring.

You should use the SQL trace to analyze any problems, provided that you know which transaction caused them. Alternatively, for Oracle databases you can analyze the shared SQL area. In order to identify resource-intensive operations, database administrators should be familiar with monitoring SQL statements in the shared SQL area.

To analyze the problem, first you should sort the shared SQL area according to the column *Disk reads* or *Buffer gets* and then analyze the SQL statements from top to bottom.

Process Flow

1. First check whether any indexes are missing [Seite 384] in the tables that are accessed in the statement.
2. Check that current statistics exist for the tables that are accessed in the statement. For more information, see Tables/Index Analysis (Oracle) [Seite 382]
3. Use SAPNet to search for notes using the keyword “Performance” and the table name of the resource-intensive SQL statement. SAP may recommend that you create or modify indexes or make program corrections.
4. You can use the explain plan (choose *Explain*) to decide whether creating or modifying an index would increase the performance of the SQL statement. Remember that a new index can also be counterproductive to system performance. Therefore you should only create new indexes after careful consideration.

To check the effect of creating an index on the performance of your system, SAP recommends the following procedure:

1. Before you create an index on the *Database performance: Shared SQL* screen, choose *Select Table* and enter the name of the table that should have the new index. The system displays all SQL statements in which the table is accessed. Save the selection results to a local file.
2. Create the index and repeat the procedure described above after the database has been productive for a short time. By comparing the *Reads/Execution* values before and after the indexes were created, you can now determine which statements were affected positively and which were affected negatively by the new index. (You should also examine the *Gets/Execution* column.)

    An object accessed by a program may not be in an optimal state as far as performance is concerned. For more information, see Monitoring Table and Index Fragmentation (Oracle) [Seite 400]

See also:

SQL Request (Shared SQL Area) [Seite 349]

Missing Indexes [Seite 384]

Checking the Optimizer Mode (Oracle) [Seite 403]

Tables/Index Analysis (Oracle) [Seite 382]

Monitoring Table and Index Fragmentation (Oracle) [Seite 400]

Table Scans: Problem Analysis (Oracle) [Seite 357]

Monitoring Table Access Methods (Oracle) [Seite 393]

Monitoring the Shared Pool (Oracle) [Seite 388]

Monitoring Table and Index Fragmentation (Oracle)

Fragmentation of tables and indexes may reduce performance, depending on the way data is accessed. Fragmentation also leads to greater overall storage space usage. This problem can be eliminated by reorganizing the particular object. However you should bear in mind that this process is very expensive and that the system is not available during a reorganization. It is often not advisable to immediately start a reorganization to eliminate fragmentation. Weigh up the pros and cons for each individual case (for example, see SAP Database Management: Oracle [Extern]).

Table Fragmentation

Table fragmentation will result in longer query times when a full table scan is performed. Since data is not as evenly packed in the data blocks, many blocks may have to be read during a scan to satisfy the query. These blocks may be distributed on various extents. In this case, Oracle must issue recursive calls to locate the address of the next extent in the table to scan.

Recent studies have shown that table fragmentation has hardly any effect on the performance of the database system. This is mainly because full table scans are somewhat rare in an SAP system since data is accessed using an index. Reorganizing table data is generally not as beneficial to performance as previously thought.

Index Fragmentation

Index fragmentation may bring a higher penalty to application performance. When accessing data through an index and an index range scan (common in R/3 Systems), Oracle must read each block in the specified range to retrieve the indexed values. If the index is highly fragmented, Oracle may have to search many more blocks, and possibly levels, to get this information. To eliminate index fragmentation, you must reorganize the index concerned. To do this, use the SAP utility program SAPDBA.

In both cases - table fragmentation and index fragmentation - you should always ensure that the soft and hard limits for the number of extents (MAXEXTENTS parameter) are not reached. If this happens, the database administrator must intervene.

See also:

Problems with Maximum Number of Extents (Oracle) [Seite 366]

Monitoring Calls (Oracle) [Seite 391]

Monitoring Table Access Methods (Oracle) [Seite 393]

Table Scans: Problem Analysis (Oracle) [Seite 357]

All Transactions are Running Slowly (Oracle)

You may encounter situations where all applications appear to be performing poorly.

The questions to ask should include:

- Has anything changed?
- When did this performance degradation start?
- Are all applications really running slowly or is the user telling you this out of frustration over their application performing poorly?
- Is this problem intermittent or constant?

If the answers to these questions lead you to believe there is a system-wide performance problem, the following topics may help you determine the cause.

Monitoring the Shared SQL Area (Oracle) [Seite 398]

Checkpoint Monitoring (Oracle) [Seite 402]

Checking the Optimizer Mode (Oracle) [Seite 403]

Monitoring Oracle Resources [Seite 404]

Missing Indexes [Seite 384]

Keep in mind that a **single** poorly-performing application may use considerable system resources causing other non-related applications to falter. For this reason, it is a good idea to also review the subjects discussed in other sections when tracing down a system-wide problem.

Checkpoint Monitoring (Oracle)

A checkpoint is an operation that Oracle performs to ensure data file consistency. When a checkpoint occurs, Oracle ensures all modified buffers are written from the data buffer to disk files. Frequent checkpoints decrease the time necessary for recovery should the database crash, but may decrease overall database performance.

Checkpoints also lead to the updating of data file headers. If the Oracle background process `CKPT` is not available for your system or is not started, the Oracle log writer (LGWR) will have to perform this task. The background process `CKPT` is active if the **`init<SID>.ora`** parameter **`log_checkpoint_process`** is set to **`true`**.

A checkpoint is automatically performed by Oracle each time an on-line redo log fills and the LGWR writes the redo entries from the redo log buffer in the next log group. The **`init<SID>.ora`** parameter **`log_checkpoint_interval`** (LOG_CHECKPOINT_INTERVAL (Oracle) [Seite 412]) determines how many checkpoints are performed between these redo log switches. If this parameter is set to a value larger than the size of the largest on-line redo log file, Oracle will not perform additional checkpoints in between the redo log file switches. The SAP default setting is usually sufficient for most productive systems.

When Oracle performs a switch of the on-line redo log files from one group to the next, archiving of the file (that has just been filled) in the archiving directory is started. This work is normally carried out by the background process ARCH. If the archiving process is not yet complete by the time the background process LGWR want to write this log again, Oracle will have to wait for the log to become available again. Only then can the LGWR process write the next redo entries from the buffer to the online redo log files. When this happens, all processes performing updates on the database may stop, since no changes can be recorded in the redo log buffer.

First check why the ARCH process cannot archive the online redo log files. Perhaps the archiving directory is full or may be the process is no longer active.

Then check whether you can avoid this problem in the future, for example, by increasing the size of the online redo log files. The LGWR will then be able to write considerably more redo entries from the buffer to the online redo log files and will not come into conflict with the ARCH process so quickly. The default size for SAP redo log files is 20 MB. You should only change this size if recommended by SAP or Oracle.

Checking the Optimizer Mode (Oracle)

Starting with version 7 of Oracle, two optimizer modes are available. The Optimizer session is established using the **init<SID>.ora**-parameter **optimizer_mode**.

- cost-based (**init<SID>.ora** parameter **optimizer_mode = choose)**• rule-based (**init<SID>.ora** parameter **optimizer_mode = rule**

The parameter **optimizer_mode** is set for R/3 Systems and should only be changed if recommended by SAP. You should also take into account the approporiate notes.

In contrast to the rule-based optimizer, you should create statistical tables for the cost-based optimizer. If you do not regularly create these table statistics, this may cause the cost-based optimizer to make wrong decisions and result in performance problems.

See also:

Displaying Oracle Table Statistics [Extern]

Monitoring Oracle Resources

Often you will need to monitor caches in the Oracle SGA to find the quality of the buffer areas. The following three areas in the SGA are the most critical. Oversizing these memory areas could result in poor system performance. More space allocated to the SGA means that less resources are available to other Oracle and non-Oracle processes.

See also:

Monitoring the Data Buffer (Oracle) [Seite 386]

Monitoring the Shared Pool (Oracle) [Seite 388]

Monitoring the Redo Log Buffer (Oracle) [Seite 390]

No Applications Can Run (‘Frozen’ System)

Database Message Log (Oracle) [Seite 352]

At times you may find that your system has completely frozen. No application can proceed, unlike a slow running system. If this happens, it will usually happen abruptly. It is often the result of a resource being completely depleted, and the Oracle database system can no longer continue.

The most common reasons for a frozen system are:

- The archiving of offline redo log files (that is, the copying of online redo log files to the archiving directory by the Oracle background process ARCH) has frozen. SAP databases in production run in ARCHIVELOG mode. This means that an online redo log file can only be overwritten with information from the redo log buffer by the log writer (LGWR) when its contents have been archived. If archiving is not possible, the database system cannot continue.

    Check whether the ARCH process is running (LOG_ARCHIVE_START (Oracle) [Seite 410]). Check whether the archiving directory is full (Archive Stuck). To do this, you can use the SAP utility program SAPDBA.

    You should monitor the archiving process on a regular basis to avoid the freezing of the system due to errors in this area. You can use the options provided by the CCMS (Displaying Redo Log Backups: Oracle [Seite 1128]).

    See also: SAP Database Management: Oracle [Extern] and the SAP notes database (for example, note number 391)

- All SAP processes are locked.

See also:

Database Message Log (Oracle)

Checkpoint Monitoring (Oracle) [Seite 402]

Exclusive Lockwaits (Oracle) [Seite 351]

Important init.ora Parameters (Oracle)

Here are some of the parameters of the Oracle profile `init<SID>.ora`. You should check the setting of these parameters if bottlenecks occur in your database system. Refer to the Oracle documentation for more information.

DB_BLOCK_BUFFERS (Oracle) [Seite 407]

DB_BLOCK_SIZE (Oracle) [Seite 408]

DB_WRITERS (Oracle) [Seite 409]

LOG_ARCHIVE_START (Oracle) [Seite 410]

LOG_BUFFER (Oracle) [Seite 411]

LOG_CHECKPOINT_INTERVAL (Oracle) [Seite 412]

ROW_CACHE_CURSORS (Oracle) [Seite 413]

SHARED_POOL_SIZE (Oracle) [Seite 414]

SORT_AREA_SIZE (Oracle) [Seite 415]

TIMED_STATISTICS (Oracle) [Seite 416]

See also:

Parameter Changes (Oracle) [Seite 356]

DB_BLOCK_BUFFERS (Oracle)

This parameter sets the number of Oracle database blocks stored in the database buffer cache of the System Global Area (SGA). Each database block buffer is equal to one Oracle database block. The data read from the data files is temporarily stored in the database buffer cache.

The product of **`db_block_buffers`** and **`db_block_size`** (DB_BLOCK_SIZE (Oracle) [Seite 408]) is the size of the Data Buffer (Oracle) [Seite 337].

The minimum value for this parameter is 4. The maximum value depends on the operating system. The SAP default is 3000. You should not normally reduce this value. Large systems typically run with much higher values.

See also:

Monitoring the Data Buffer (Oracle) [Seite 386]

DB_BLOCK_SIZE (Oracle)

This parameter sets the size in bytes of Oracle database blocks. Together with **`db_block_buffers`** (DB_BLOCK_BUFFERS (Oracle) [Seite 407]), it is used to determine the size of the Data Buffer (Oracle) [Seite 337] in the SGA. The parameter **`db_block_size`** can only be set at the time of creating the database and cannot be changed afterwards.

The parameter **`db_block_size`** is set to 8192 bytes for most R/3 installations. It should not be modified.

DB_WRITERS (Oracle)

This parameter determines the number of Oracle database writer processes that are started at database instance startup. Each database writer consumes one semaphore and an amount of process memory. Database writers are used by Oracle to write out modified blocks from the SGA to disk.

R/3 Systems generally use one database writer process. It is advisable to increase this amount if you will be performing comprehensive update or insert activities. For medium to large configurations, set this parameter so that there is one database writer process per Oracle data file disk. As there are several factors which determine the usefulness of changing this parameter, it is recommended to check with an Oracle or SAP specialist before modifying its value.

LOG_ARCHIVE_START (Oracle)

This parameter is used to activate (parameter value `true`) or deactivate (parameter value `false`) automatic archiving of the online redo log files in the archiving directory (`log_archive_dest`).

The database should always run in ARCHIVELOG mode for productive R/3 Systems. For a database system to be able to run in this mode, parameter `log_archive_start` must be set to `true`. In this case, the background process ARCH is automatically triggered for a redo log file switch in order to start archiving the online redo log files in the group that is just being used. These online redo log files cannot be written again until they have been archived successfully.

If ARCHIVELOG mode is activated and `log_archive_start` is set to `false`, the database freezes once all the online redo log files are filled. In this case, the background process ARCH is not triggered in order to write the online redo log files to the archiving directory. However, the online redo log files cannot be overwritten until they are archived.

LOG_BUFFER (Oracle)

This value specifies the size in bytes of the redo log buffer in the SGA. The redo log buffer is used by Oracle to store information about changes made to the database (redo entries). This information is needed for recovery purposes. The information in the redo log buffer is written to the online redo log files by the LGWR process.

The default setting for this parameter for R/3 Systems is 327680. This value is sufficient for most applications. If modifying this value, note that it must be set to a multiple of **`db_block_size`**.(DB_BLOCK_SIZE (Oracle) [Seite 408]).

See also:

Redo Log Buffer (Oracle) [Seite 335]

Monitoring the Redo Log Buffer (Oracle) [Seite 390]

LOG_CHECKPOINT_INTERVAL (Oracle)

This parameter specifies the number of filled on-line redo log blocks necessary to trigger a checkpoint. A checkpoint will always occur when there is an on-line redo log switch. It is usually not necessary to have Oracle perform checkpoints between log switches, though in some cases it may be desirable.

If the value specified for this parameter exceeds the overall size of the online redo log files, a checkpoint will only occur for a redo log file switch. For R/3 Systems, the value is set to 3000000000.

See also:

Checkpoint Monitoring (Oracle) [Seite 402]

ROW_CACHE_CURSORS (Oracle)

This parameter sets the number of cached recursive cursors used for selecting lines from the data dictionary. SAP recommends setting this parameter to a value of at least 100.

See also:

Monitoring Calls (Oracle) [Seite 391]

SHARED_POOL_SIZE (Oracle)

This parameter specifies the size of the shared pool in the SGA in bytes. The shared pool is made up of several memory structures. Most important among these are the data dictionary cache and the library cache (containing the shared SQL area).

Previous Oracle versions provided a more detailed approach to tuning these memory structures. With Oracle 7, these memory areas are dynamically managed by Oracle through the setting of this parameter.

See also:

Shared Pool (Oracle) [Seite 338]

Monitoring the Shared Pool (Oracle) [Seite 388]

Dictionary Buffer (Oracle) [Seite 346]

SQL Request (Shared SQL Area) [Seite 349]

SORT_AREA_SIZE (Oracle)

This parameter specifies the amount of memory that can be used for sort operations. Sorting is needed for SQL statements which use ORDER BY, GROUP BY and SORT MERGE JOIN operations. Sorting is also done during index creation.

Sort operations requiring more sort area space than specified by this parameter will use temporary segments on disk. It is important to note that the memory specified in **`sort_area_size`** is assigned for every Oracle process that performs a sort operation, so the total memory used can add up quickly on an active system.

The SAP default for **`sort_area_size`** is 2097152 bytes. This value is normally sufficient.

See also:

Sorts (Oracle) [Seite 332]

Monitoring Sorting (Oracle) [Seite 395]

TIMED_STATISTICS (Oracle)

This parameter determines whether database statistics for particular times will be logged by Oracle. This information may be useful to monitor system or application performance.

The value for this setting may be `true` or `false`. In R/3 Systems. To monitor the Oracle database and recognize problems in time, SAP recommends that you set `timed_statistics` to `true`. With a rise in statistical data, you accept a slight increase in database load.

Data for the Oracle Database Monitor

Data is provided in various ways for the individual screens of the Database Monitor. A more detailed description follows of how data is provided for the following Database Monitor screens:

- SAP/Oracle Database Monitor: Main Screen [Seite 331]

    In most cases, current data is read from the relevant Oracle tables.

    See: Data for the Main Screen of the Database Monitor [Seite 420]

- SAP/Oracle Database Monitor: Status of the Data [Seite 368]

    The data is provided by a series of ABAP programs. This collection of programs is called the “database collector” (or “DB collector”).

    – The DB collector is started regularly in the background.

    See: The Database Collector in Background Processing [Seite 418]

    – The DB collector can be started online.

    See: Data for the Screen: Database Performance: Tables and Indexes [Seite 422]

Database Collector in Background Processing

The data collector `RSCOLL00` is started every hour as a background job (name of the job: COLLECTOR_FOR_PERFORMANCEMONITOR or SAP_COLLECTOR_FOR_PERFMONITOR) (Configuring the Data Collector [Seite 991]). When this happens, the ABAP programs that belong to the database collector are also started.

Check that the configuration of the data collector allows the programs of the database collector to be started at the required times:

- Entries in table `TCOLL`:

  Each entry corresponds to a report executed within the execution of report `RSCOLL00`. The reports are executed on the days and at the times specified on the database instance of the R/3 System (entry system = C) if there is a dialog system available there. Otherwise the first available dialog system is automatically used (Scheduling Data Collector Reports [Seite 994]).

  Table: Entries in TCOLL that are relevant to the database collector

Report	Task of the report	Day	Time of day
`RSDBPREV`	Calls the database-specific report in order to receive current statistics from the database system (RSDBPREV [Seite 1001])	Daily	Every two hours
`RSORATDB`	Analyzes the tablespaces, tables and indexes and stores the results in table MONI (RSORATDB [Seite 1003])	Daily	Once a day
`RSORAPAR`	Reads the database parameters and stores them in table PAHI (RSORAPAR [Seite 1005])	Daily	Once a day
`RSORA811`	Deletes old BRBACKUP and BRARCHIVE logs (RSORA811 [Seite 1007])	Daily	Once a day

  To change the specifications in table `TCOLL`:

  Call Transaction `ST03` and choose *Environment* → *Data collector* → *Collector frequency*.

- Scheduling the data collector

  Check that the job COLLECTOR_FOR_PERFORMANCEMONITOR or SAP_COLLECTOR_FOR_PERFMONITOR is scheduled correctly.

  Using the Graphical Job Scheduling Monitor [Seite 135]

  Make sure that the user under whose ID the job is running has the necessary authorization (user `DDIC`)

  - Program: `RSCOLL00`
  - Variant: no
  - Repetition period: hourly
  - Client-dependent: no

  If you want to change these job scheduling specifications, choose Scheduling Background Jobs [Seite 106].

- You can display the logs of the data collector.

To display using the Workload Monitor:

1. Choose *Tools* → *CCMS*→ *Control/Monitoring* → *Performance Menu* → *Workload* → *Analysis*.

   Alternatively, use Transaction `ST03`.

2. Choose *Environment* → *Data collector* → *Display protocols*.

See also:

Workload Monitor [Seite 965]

Or start report `RSCOLL20`.

The logs are deleted automatically after a time period which you can set. You can set the retention time in the Workload Monitor under *Goto* → *Parameters* → *Performance database*, entry *Time comparison data - Days*.

See also: Configuring the Data Collector [Seite 991]

- To be able to access current Oracle statistics, the Oracle parameter TIMED_STATISTICS (Oracle) [Seite 416] must be set to `true`.

Data for the Main Screen of the Database Monitor

Main screen of the Database Monitor *Database Performance Analysis*: SAP/Oracle Database Monitor: Main Screen [Seite 331].

The statistics on the main screen are determined from the relevant Oracle V$ tables when you call the Monitor. An Oracle statistics "snapshot" recording current database activity is therefore provided. These details are constantly updated by the Oracle database system, beginning from the last system startup.

Choose *Refresh* to update the details (the relevant V$ tables will be read again).

Detail analysis menu

A series of more detailed analysis options is available via *Detail analysis menu* (Detail Analysis Menu (Oracle) [Seite 341]). The DB collector is started for some of these analyses.

Table: Analyses without the DB collector

Pushbutton	
Buffer busy waits, *Filesystem requests*, *Wait events*, *SAP client*, *Oracle session*, *SQL request*, *Exclusive lockwaits*, *Latch waits*	Statistical data determined directly from the dynamic performance tables (Oracle V$ tables).
Database message log	Displays the ALERT file (central Oracle log file).
Display V$ Tables	Displays the list of Oracle V$ tables.

Table: Analyses using the DB collector

Pushbutton	
Performance database	Displays the performance statistics of the database. Snapshot data is provided, which is collected regularly in table DBSNP using reportRSDBPREV [Seite 1001].
State on disk	Displays the screen *Database Performance: Tables and Indexes*. The details on this screen are basically provided by report RSORATDB [Seite 1003]. **See also:** Data for the Screen: Database Performance: Tables and Indexes [Seite 422].
Parameter changes	Displays the init.ora parameters and their changes. These details are provided by report RSORAPAR [Seite 1005], which stores the old and new parameters in table PAHI.
Summary report	Displays up-to-date overall statistics (settings, database parameters, data status). These statistics are provided by reports RSORATDB [Seite 1003], RSORAPAR [Seite 1005], and RSORA811 [Seite 1007] of the DB collector.

See also:

The Database Collector in Background Processing [Seite 418]

Data for the Screen: Database Performance: Tables and Indexes

Screen *Database Performance:Tables and Indexes*: SAP/Oracle Database Monitor: Status of the Data [Seite 368].

You can use the analysis options on this screen to get detailed information on the data of the database system. These details are updated regularly by report RSORATDB [Seite 1003] (The Database Collector in Background Processing [Seite 418]). The *Database system* section shows the time of the last analysis.

Choose *Refresh* if you require more up-to-date status data. RSORATDB is then started again. You should only use this option if absolutely necessary, as it can take a long time to determine all the information depending on the size of the database.

Database system	
Refresh	RSORATDB is started immediately to update all statistical data.
Checks	Up-to-date statistics are created, and some statistics already generated with RSORATDB are displayed. Details on extents and missing indexes are always up-to-date.
Space statistics	History of database size. These values are read from table MONI.
Tablespaces	
Current sizes	These details were determined during the last RSORATDB run.
Space statistics	History of tablespace size. These values are read from table MONI.
Freespace statistics	These details were determined during the last RSORATDB run.
Tables and indexes	
Detailed analysis	These details were determined during the last RSORATDB run.
Missing indexes	RSORATDB examines the database for missing indexes.
Space critical objects	These details were determined during the last RSORATDB run.
Space statistics	History of table/index size. These values are read from table MONI.

Data for the Database Alert Monitor

The data for the Database Alert Monitor is mainly read from dynamic performance tables (Oracle V$ tables). Some information is also read from table MONI. Table MONI is filled by collector runs which you schedule as required.

See also: The Database Collector in Background Processing [Seite 418]

Example: The DB collector determines whether there are freespace problems in the tablespaces or whether indexes are missing (report RSORATDB [Seite 1003]), and writes this information to table MONI. The status information written at the time of the collector run appears on the Alert Monitor.

SAP/Informix Database Monitor

Use

You can use the SAP/Informix database monitor [Seite 328] to check the status and performance of the database system. The monitor displays the most important indicators of Informix database performance, that is, shared memory, data buffers, cleaners, log buffers, database activity, checkpointing, and asynchronous IO. All data for the display comes from the database host, either through the SQL interface or the interface to the Informix tool `onstat`.

Integration

All the information provided by the SAP/Informix database monitor is based on the diagnostic information that the Informix database supplies to the SAP System. Since all data comes from the database host, you can look at the same data from every application server in your SAP System.

The definitive documentation for Informix performance tuning comes from the DBMS manufacturer, Informix. Use recommendations from the Informix documentation where they differ from any recommendations given here.

Features

Informix stores statistics on database state and activity either in real tables on disk or virtual tables in memory. These tables in memory are also called “pseudotables” and the interface to these tables is called the “sysmaster interface.” The SAP/Informix database monitor is implemented using queries to these tables. The tables can be queried either on the database server or from a separate computer.

The following features are available on many screens in the database monitor:

Choose	**To**
Refresh	Refresh the values shown on the screen
Reset	Reset values to 0
Since Reset	Analyze the time period since you last pressed *Reset*
Since DB start	Display the values that have accumulated since you started the database

Activities

- To see the database overview [Seite 425], choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity*.
- To see detailed information [Seite 433], choose *Detail Analysis Menu* from the database overview.
- You decide on a performance monitoring strategy [Seite 448] for your database.
- If necessary, you diagnose performance problems [Seite 462].
- If necessary, you tune specific parameters [Seite 470].

Looking at the Informix Database Overview

Use

The database overview in the SAP/Informix database monitor [Seite 424] is the starting point for you to monitor your database. You can call the database monitor from every application host of your SAP System.

For more information about the detailed database analysis, see Detailed Database Analysis for Informix [Seite 433].

Prerequisites

- For access to the SAP/Informix database monitor, the interface to the Informix tool `onstat` (running locally on the database host) must be working correctly. For more information, see Checking Onstat Commands [Seite 447].
- Read SAP Note 84060 for the latest information on the SAP/Informix database monitor.
- To use all features of the SAP/Informix Database Monitor, you must have Informix version 7.x.

Procedure

Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* to see the overview screen in the database monitor.

Use the database monitor to monitor your database regularly to ensure the best possible performance.

Result

The system displays the most important indicators of Informix database performance:

- Shared Memory (Informix) [Seite 426]
- Data Buffers (Informix) [Seite 427]
- Cleaners (Informix) [Seite 428]
- Log Buffers (Informix) [Seite 429]
- Database Activity (Informix) [Seite 430]
- Checkpointing (Informix) [Seite 431]
- Asynchronous I/O (Informix) [Seite 432]

Shared Memory (Informix)

Definition

Shared memory is memory that can be accessed by more than one process. Almost all memory used by an Informix database is shared.

Use

You can use the database overview [Seite 425] of the SAP/Informix database monitor to view shared memory.

If you are running Informix version 6, select *Shared Memory Incl.* from the main screen to display an overview of shared memory.

Structure

Informix divides its shared memory into the following classes:

Class	Description
Resident	The resident portion is fixed at startup and contains, among other objects, the data buffer and the log buffers.
Virtual	Important in the virtual portion are the session pools, that is, memory used for each session. The virtual portion of shared memory can also grow during normal business.
Message	The message portion is typically very small and not relevant for tuning.

Data Buffers (Informix)

Definition

Data buffers are Informix pages held in memory to reduce disk I/O. Pages are the smallest physical unit of storage, which is platform dependent and either 2 KB or 4 KB. In fact, Informix refers to each such page as a “buffer”, and the sum of all buffers as the “data buffer cache.”

The SAP/Informix database monitor shows current read and write quality in the buffers, that is, what percent of total reads and writes could be satisfied in the buffer without going to disk.

Use

You can use the database overview [Seite 425] of the SAP/Informix database monitor to view data buffers.

See also:

BUFFERS (Informix) [Seite 473]

Cleaners (Informix)

Definition

Cleaners in the Informix database are responsible for writing modified pages to disk. “Page cleaners” are the Informix threads that organize disk I/O. The number of cleaners is set using the CLEANERS parameter.

Data buffers are grouped together into “least recently used (LRU) queues” to minimize contention when processes look for free buffers. The number of LRU queues is set using the LRUS parameter. Refer to LRUS (Informix) [Seite 480].

Use

You can use the database overview [Seite 425] of the SAP/Informix database monitor to view cleaners and LRU queues.

When tuning cleaners, it is important to decide how many modified (that is, “dirty”) pages should be allowed to exist in the buffer. The values *LRU start cleaning* and *LRU end cleaning* tell the system when to start writing modified pages between checkpoints, and when to stop. The values are expressed as a percentage of total pages and are set using the LRU_MAX_DIRTY and LRU_MIN_DIRTY parameters respectively. Refer to LRU_MAX_DIRTY, LRU_MIN_DIRTY (Informix) [Seite 479].

Informix distinguishes between various classes of writes to disk, as shown in the table below. Statistics on these writes describe I/O activity in the system.

Types of Write To Disk for Informix

Type of Write	Explanation
Foreground writes	These occur synchronously when a process needs a buffer, does not find a clean one, and is forced to activate a cleaner and wait for the buffer to be written out. Foreground writes are very bad for performance but fortunately rare in Informix version 6 or later.
LRU writes	These are performed by the page cleaners between checkpoints. These writes are not as efficient as “chunk writes” (see below), but are important in reducing the total I/O to be performed at checkpoint.
Chunk writes	These are highly efficient writes that Informix uses to move large numbers of data pages at a checkpoint, for example, by sorting pages and sending them in large write operations to disk.

Log Buffers (Informix)

Definition

There are physical and logical-log buffers for the Informix database.

The log buffers store pages to be written to the physical and logical-log files. Refer to Checkpointing (Informix) [Seite 431]. The amount of data written per I/O is also displayed. This value is used in tuning the size of the buffers.

Use

You can use the database overview [Seite 425] of the SAP/Informix database monitor to view log buffers.

Database Activity (Informix)

Definition

The database activity display collects miscellaneous information on configuration and activity in the database. The number of CPU virtual processors (CPU VPs) is important because these are the processes that do CPU-intensive work in the Informix database.

Use

You can use the database overview [Seite 425] of the SAP/Informix database monitor to view database activity.

Checkpointing (Informix)

Definition

A checkpoint is the action or point of time when Informix has written all modified pages from the buffers to disk, so the database is physically consistent. The following logs are maintained by the Informix database and are relevant to checkpoints:

- Physical log

  This enables restore to a consistent database state as of the last checkpoint in the event of a crash or power failure.

- Logical log

  This is used for:

  – Transaction logging, to enable rollbacks of aborted transactions

  – Logging of all changes to the database, to enable a restore as of the last archive and up to the point of failure

Use

You can use the database overview [Seite 425] of the SAP/Informix database monitor to view checkpointing.

The size of the physical log is relevant for checkpointing because a checkpoint is forced when the physical log becomes 75% full. The figures *Chkpoint Correlation %* and *Physical Correlation %* are intended to give an approximate idea of how often checkpoints have occurred because the checkpoint interval has finished and how often because the physical log is 75% full. The checkpoint interval has a default value of 900 seconds in an SAP installation.

The figures are approximate and so do not always add up to exactly 100%.

Asynchronous I/O (Informix)

Definition

Asynchronous I/O (AIO) is performed either by special processes, called AIO virtual processors (AIO VPs), or, when kernel asynchronous I/O is supported by the operating system, by the CPU VPs using kernel asynchronous I/O (KAIO) threads. This part of the main screen tells how many of each of these entities are configured.

The *Files* column is simply a count of distinct paths used for the raw devices in the database (normally with the path `/informix/SID/sapdata/...`). Informix assumes that these distinct paths refer to separate physical disks, though with LVM and RAID5 this might not in fact be true.

Use

You can use the database overview [Seite 425] of the SAP/Informix database monitor to view AIO.

Detailed Database Analysis for Informix

Purpose

With the detailed database analysis in the SAP/Informix database monitor [Seite 424] you can look in depth at the Informix database for help with advanced monitoring of the database. For more information about the overview, see Looking at the Informix Database Overview [Seite 425].

Process Flow

You choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* to look at the detailed database analysis.

The most commonly used functions are the following:

- Checking Wait Situations (Informix) [Seite 434]
- Checking Chunk I/O Activity (Informix) [Seite 435]
- Checking Sessions (Informix) [Seite 437]
- Checking Table Activity (Informix) [Seite 438]
- Checking SQL Statements (Informix) [Seite 440]
- Checking Exclusive Lockwaits (Informix) [Seite 442]
- Checking the Database Message Log (Informix) [Seite 443]
- Checking State on Disk (Informix) [Seite 444]
- Checking Pseudotables (Informix) [Seite 445]
- Checking Parameter Changes (Informix) [Seite 446]
- Checking Onstat Commands (Informix) [Seite 447]

Checking Wait Situations (Informix)

Use

You can use the detailed database analysis [Seite 433] in the SAP/Informix database monitor to check statistics on the thread status. Although called wait statistics, these statistics cover all thread states, that is, they also include information on how long the threads are running, so providing a global overview of how time is spent in the database.

For the first part of the *Wait Situations* display, the possible thread states are summarized for all threads in the database. The state of threads that are simply idling and waiting for work does not appear in the display, so that you only see the important wait reasons and time spent running.

Prerequisites

The statistics on thread status are collected if the parameter `WSTATS` is set to 1 in the Informix `ONCONFIG` file

Procedure

Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Wait Situations*.

Result

The thread states are displayed:

Thread State	**Meaning**
Running	Being processed by its virtual processor (VP)
AIO	Waiting on I/O
Mt Ready	Ready and waiting for VP
Buffer	Waiting for a buffer
Lock	Waiting on a lock
Checkpoint	Waiting during checkpoint
Mt yield	Has voluntarily given up control in VP
Mutex	Waiting on a mutex

Checking Chunk I/O Activity (Informix)

Use

You can use the detailed database analysis [Seite 433] in the SAP/Informix database monitor to check chunk input/output (I/O) activity.

Prerequisites

You need version 7.x or later of the Informix database management system (DBMS) to use the full functionality described in this procedure.

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Chunk IO Activity* ("IO" refers to disk input and output).

   You can alter the kind of information you see in the display as follows:

   – By choosing different categories of information (see step 2). The system first displays *Disk I/O Summary Statistics per Path*.

   – By choosing a different starting point for the information (see step 3)

2. You can view different categories of information:

Choose	**To See**	**With the Information**
Chunks per path	*Chunk Activity Statistics per Path*	Disk activity broken down by chunk and disk device (that is, path)
Chunks per Dbspace	*Chunk Activity Statistics per Dbspace*	Disk activity broken down by chunk and dbspace
Total per path	*Disk I/O Summary Statistics per Path*	Disk activity broken down by path (that is, a summary of what you see when you choose *Chunks per path*)
Total per Dbspace	*Disk I/O Summary Statistics per Dbspace*	Disk activity broken down by dbspace (that is, a summary of what you see when you choose *Chunks per Dbspace*)

3. You can view different starting points for the information (the system displays *Informix Disk I/O Statistics Since DB Start* at first):

Choose	**To**
Reset	Set all figures to zero
Since Reset	Display disk activity since the last time *Reset* was chosen
Since DB Start	Display disk activity since your Informix DBMS was started

   When you choose *Reset* or *Since Reset*, the system alters the title of the display accordingly, for example:

   Informix Disk I/O Statistics Since 11:39:08

4. Choose *Sort* with the cursor on a column heading to sort the data by that column. This is useful, for example, to see which disks have the most I/O.

   The statistics used for these displays are stored in shared memory. These statistics can overrun, that is, there is no longer enough storage to hold all values. The result is negative figures – or, still later, misleading positive figures – appearing on the display. This might occur particularly if you are looking at figures since DB start. For more information about this difficulty, see SAP Note 40669.

Result

By monitoring disk I/O activity regularly, you can identify performance problems (for example, disk “hotspots”) early, so avoiding unplanned downtime. Therefore, you can make sure that the availability of your SAP System is maximized.

Checking Sessions (Informix)

Use

You can use the detailed database analysis [Seite 433] in the SAP/Informix database monitor to check sessions.

The session monitor provides activity information on each session in the database. In the SAP environment every work process has one session, so that this function represents the database view of work process activity. The host name and process ID can be used in conjunction with transaction `SM51` to determine which work process belongs to the session.

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Informix session*.

    The first screen of the session monitor shows, in addition to the *Host Name* and *Process ID*, the *Session ID*, the *Thread ID* of the `sqlexec` thread, total *RSAM calls*, *Running time*, *Wait time* and the current thread status.

2. To see the most recent SQL statement issued by the work process, select a session and choose *Current sql*.

Checking Table Activity (Informix)

Use

You can use the detailed database analysis [Seite 433] in the SAP/Informix database monitor to check table activity.

Prerequisites

You need Informix version 7.x to use the full functionality described in this procedure.

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Table Activity*.

   The system displays the numbers of reads and writes for each table in the database. For each table, look at the fields *Bufreads* and *Bufwrites* to see information on access through the buffer (that is, from copies of the data held in main storage). You can see information on disk accesses under the fields *Pagereads* and *Pagewrites*.

   You can change the information you see in the display as follows:

   – By choosing different fields (see steps 2 to 4)

   – By choosing to also display statistics information for each table (see steps 5 to 8), but note that this requires extra processing power, so it might take the system longer to display the information

   – By choosing a different starting point for the information (see step 9)

   – By sorting the information according to different fields (see step 10)

2. To view information about the number of times a table has been accessed by a sequential scan and the number of rows in each table, choose *Next info*.

   The system displays different fields, including *Seqscans* and *rows*.

3. To view information about ISAM reads and writes, choose *Next info* again.

   The system displays different fields, including *Isreads, Iswrites, Isrewrites* and *Isdeletes*.

4. To view information about locking, choose *Next info* again.

   The system displays different fields, including *Users*, *Lockreqs Deadlocks*, *Lktouts*, *Lockwts*.

   You can change the display to alternately show the fields described in steps 1 (what you see when you start), 2, 3, and 4 by choosing *Next info* and *Prev info*.

5. To view statistics information for a single table, select the table you want to examine and choose *Last Statistic*.

   The system displays the date the table's statistics were last updated and the level of the update.

6. To view statistics information for all tables, choose *Edit → Update Statistics → Display all tables*.

   For all tables, the system displays the date the table's statistics were last updated and the level of the update.

7. To always see statistics information for all tables, choose *Edit → Update Statistics → Display permanently → On*.

   For all tables, the system always displays the date the table's statistics were last updated and the level of the update.

8. To refresh the statistics information for all tables, choose *Edit → Update Statistics → Read/Display all new*.

   The system displays the updated statistics information for all tables.

9. You can view different starting points for the information (the system displays *Informix Table Activity Monitor since DB start* at first):

Choose	To
Reset	Set all figures to zero
Since Reset	Display table activity since the last time *Reset* was chosen
Since DB Start	Display table activity since your Informix database management system (DBMS) was started

   When you choose *Reset* or *Since Reset*, the system alters the title of the display accordingly, for example:

   Informix Table Activity Monitor since 11:36:18

10. To sort the data by a field, place the cursor on a data element in a column heading and choose *Sort*. This is useful, for example, to see which tables have the most activity.

   If you are viewing information on the number of sequential scans (see step 2), you can sort the information on this field, enabling you to see the tables with the most sequential scans. Look at the field *rows* to see whether the sequential scan makes sense (for tables with less than approximately 100 rows, it is normally better to do a sequential scan than to use an index).

Checking SQL Statements (Informix)

Use

You can use the detailed database analysis [Seite 433] in the SAP/Informix database monitor to check SQL statements. This allows you to monitor the statistics for SQL statements, so enabling identification of statements that are consuming an excessive amount of resources, for example, when the amount of data to be processed is very large.

Prerequisites

Check the settings of the following parameters from the Informix `ONCONFIG` file before starting this procedure:

- `SQLSTATS`

    You must set this variable in one of the following ways if you want to see information on SQL statements:

    - `SQLSTATS = 1`

        No time measurement is performed.

    - `SQLSTATS = 2`

        Time measurement is performed.

- `USEOSTIME`

    If you are using time measurement (see `SQLSTATS = 2`), then make sure you set the variable as follows:

    `USEOSTIME = 1`

    If you set the variable to 0, the time measurement is inexact. Setting it to 1 makes sure you get more accurate measurements, but uses more processing resources. Therefore, we recommend you set it to 0 when you have finished looking at SQL statements.

To view the statistics for SQL statements from all applications, you must set the environment variable as the user who starts the database (that is, `<sid>adm`). Otherwise, you can only view statistics for the applications you start. If you only set `SQLSTATS` for one application host, the values displayed in the procedure below only refer to SQL statements sent from the single application host.

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *SQL Statement*.

    The system prompts you for *Selection Criteria* (*Session ID*, *Buffer Reads*, and so on).

2. Enter selection criteria if you want to restrict the display.
3. Choose *Continue*.

    The system displays details of SQL statements. For more information, see Monitoring SQL Statement (Informix) [Seite 459].

See also:

USEOSTIME (Informix) [Seite 495]

Checking Exclusive Lockwaits (Informix)

Use

You can use the detailed database analysis [Seite 433] in the SAP/Informix database monitor to check the following information on lockwaits:

Lock Holder	Lock Waiter
Database user name	Database user name
Host name	Host name
Process ID (PID)	Process ID (PID)
Session ID (SID)	Session ID (SID)
ABAP program name	Name of the locked table
Work process time	ABAP program name
Status information	Work process time
SAP user name	Wait period
	How long the lock has already been held
	SAP user name

The host name and PID enable you to identify holder and waiter SAP work processes.

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Exclusive Lockwaits*.
2. Choose *SQL Statement* to display the most recent SQL statement from holder or waiter (not necessarily the one causing the lock for the holder, since the holder might have gone on to other statements within one logical unit of work).

   If there is no work process left for the lock monitor itself, use `onstat` instead to determine which process is holding the lock. Refer to Analyzing Lockwaits with onstat (Informix) [Seite 468].

Checking the Database Message Log (Informix)

Use

You can use the detailed database analysis [Seite 433] in the SAP/Informix database monitor to check the database message log. This log is used by Informix to record major events such as system startup and stop, archives, backups, checkpoints, log file full situations, and major errors (for example, hardware errors causing a chunk to go offline).

The interface to the database message log, which is written to a local file system on the database server, is implemented through `infstrfc` (only when there is no application server running on the database server).

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *State on disk* → *Database Message Log*.

   The system prompts you to choose between *Only Alerts* and *All Messages*.

2. Choose *All Messages* normally. However, if you want to filter out the message *Checkpoint complete*, choose *Only Alerts*. This message is useful for checkpoint tuning [Seite 452].

Checking State on Disk (Informix)

Use

You can use the detailed database analysis [Seite 433] in the SAP/Informix database monitor to check the state of tables, indexes, and dbspaces on disk. For example, you can find out how large your database is, how large certain tables are, and which objects have an excessive number of extents. Checks for full dbspaces and missing indexes are particularly important.

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *State on disk*.

   The screen *Database Performance: Tables* is displayed.

2. To check whether any database indexes are missing, select *Missing Indexes*.

   A hierarchical list is displayed.

3. To find history information about tables, choose *Goto* → *Space Management* → *Performance Database* → *Of Tables*.

   The system only displays tables for which the storage parameters have changed.

4. To display all tables, choose *History* → *All objects on/off*.

Checking Pseudotables (Informix)

Use

You can use the detailed database analysis [Seite 433] in the SAP/Informix database monitor to check pseudotables, which are the Informix tables providing information for the monitor. This is provided for troubleshooting, in case there are errors or omissions in the formatted displays that make up the rest of the monitor.

Procedure

Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Display Pseudotables*.

Checking Parameter Changes (Informix)

Use

You can use the detailed database analysis [Seite 433] in the SAP/Informix database monitor to check parameter changes. You can use this procedure to display current and historical values of the `ONCONFIG` parameters. The information displayed in this procedure is read from the `SYSMASTER` table `SYSCFGTAB`. For more information about `SYSMASTER` tables, see your Informix documentation.

Prerequisites

You need Informix version 7.x or later to use the full functionality described in this procedure. With version 6.x, certain important parameters (for example, `NETTYPE`) are not included in the display described in this procedure.

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Parameter Changes*.

   The system displays the date of the most recent change to a database parameter.

2. To view the changes within a specific time period, choose *Select Period*, entering a valid date in the format DD.MM.YYYY (for example, 24.11.1997) when the system prompts you.

   The system displays a list of changes with the most recent change at the top of the list. For each change the system displays the *Mod.Date* (that is, the date the parameter was changed), the *Parameter* name, and the parameter's *New Value*.

3. To view the current value of parameters, choose *Active Parameters*.

   The system displays the current parameter values.

4. To view the full history of parameter changes, choose *History of file*.

   The system displays the *Parameter* names in alphabetical order. For each parameter, the system displays a list of changes, showing the *Mod.Date* (that is, the date the parameter was changed) and the parameter's *New Value* after the change.

   You can also display `ONCONFIG` parameters using `onstat -c`. However, this reads directly from the current `ONCONFIG` file. This file might contain new parameter changes, which take effect only after the next database restart.

Checking Onstat Commands

Use

You can use the detailed database analysis [Seite 433] in the SAP/Informix database monitor to check `onstat` commands.

Prerequisites

Make sure that the interface to the Informix tool `ontstat` is working correctly. For more information if you have problems calling `onstat` (or the interface program `infstrfc`), see SAP Note 16454. This kind of problem is indicated if you see an error message like the one shown in the procedure below.

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Onstat commands*.

   The system displays a list of `onstat` commands.

2. Select the required command and choose *Choose*.

   You might see an error like the following:

   RFC error /No such file or directory /usr/sap/ARG/SYS/exe/run/infstrfc

   Please check TCP/IP destination DBSERVER with transaction SM59

   Choose *Recreate DBSERVER* to try to fix the problem. If this fails, see SAP Note 16454.

See also:

Informix documentation

SAP/Informix Performance Monitoring

Purpose

It is important to regularly monitor the performance of the Informix database.

Prerequisites

The information given in this section contains suggestions based on SAP's experience with Informix customers. The definitive documentation for Informix performance tuning comes from the database manufacturer, Informix. Use recommendations from the Informix documentation where they differ from any recommendations given here.

Process Flow

You monitor the database regularly as follows:

- Monitoring Data Buffers (Informix) [Seite 449]
- Monitoring the I/O Subsystem (Informix) [Seite 450]
- Monitoring Checkpoints (Informix) [Seite 452]
- Monitoring LRUS (Informix) [Seite 454]
- Monitoring CPU VPs (Informix) [Seite 455]
- Monitoring the Informix Network Interface [Seite 456]
- Monitoring Virtual Memory Usage (Informix) [Seite 457]
- Monitoring Optimizer Statistics (Informix) [Seite 458]
- Monitoring SQL Statement (Informix) [Seite 459]

Monitoring Data Buffers (Informix)

Use

You can use the SAP/Informix database monitor to monitor Informix data buffers.

Tuning buffers makes accesses to memory much faster than those to disk. The basic rule when tuning buffers is to keep increasing the buffers until new increases provide no improvement in performance. However, avoid significantly increasing the operating system paging.

For up-to-date recommendations on data buffers, see SAP Note 38307.

Procedure

Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* to see the display for Data Buffers (Informix) [Seite 427] on the main screen of the database monitor.

As a general rule for buffer quality, the read quality should be greater than 95% and write quality greater than 82%.

Monitoring the I/O Subsystem (Informix)

Use

You can use the SAP/Informix database monitor to monitor the Informix I/O subsystem.

In addition to the important question of when pages are written to disk, you can determine how many processes (that is, virtual processors or VPs) actually do the writing and how many threads (that is, page cleaner threads) should manage the I/O. Refer to Monitoring Checkpoints (Informix) [Seite 452].

Procedure

1. Check for disk hotspots:
   - Choose *Tools → CCMS → Control/Monitoring → Performance Menu → Database → Activity → Detail Analysis Menu → Chunk I/O Activity*.

     For more information, see Checking Chunk I/O Activity (Informix) [Seite 435].
   - Try to distribute I/O load over all disks that are available to the database.
2. Check the cleaners.

   With the `CLEANERS` parameter in the Informix `ONCONFIG` file you specify how many threads do the work of organizing I/O. Here the general rule is to configure one cleaner thread per physical disk, with a rough maximum of three per controller. The maximum is 32 cleaner threads in total.
3. Check the asynchronous I/O virtual processors (AIOVPs).

   For most current SAP hardware platforms, I/O is performed using separate AIO VPs. You configure the number of AIO VPs using the `NUMAIOVPS` parameter in the Informix `ONCONFIG` file.
4. Check the AIO.

   Certain operating systems (for example, Sun Solaris) support kernel AIO (KAIO). When the operating system allows it, this support is built into the Informix system (that is, you do not need to configure it). When present, the CPU VPs use KAIO threads to tell the operating system to perform the I/O asynchronously. There is no context switch to an AIO VP, and the CPU VP can continue processing without waiting for the I/O.

   We suggest the following:
   - If kernel asynchronous I/O is implemented, then make sure that `NUMAIOVPS` is 2.
   - If you are **not** using kernel asynchronous I/O, you can find out whether you need to add AIO VPs as follows:
     - Use the global view of your system status

       Choose *Tools → CCMS → Control/Monitoring → Performance Menu → Database → Activity → Detail Analysis Menu → Wait Situations*.

       If threads spend more time running than waiting on I/O, you probably do not have an I/O problem.
     - Use `onstat -g iov`

This command shows you all I/O VPs. Look at those in the class AIO. Informix takes the first free AIO VP in order when I/O is performed. So if you see that the first AIO VPs in the list have performed most of the operations and the last AIO VPs relatively few operations, then you probably have enough AIO VPs.

The general rule is to set `NUMAIOVPS = <number of disks>`.

Monitoring Checkpoints (Informix)

Use

You can use the SAP/Informix database monitor to monitor Informix checkpoints.

Prerequisites

When tuning checkpoints, remember that:

- The most efficient writes, called “chunk writes,” occur at checkpoint.
- Other threads effectively stop working during a checkpoint.

You can use the following approaches:

Approach	Advantage	Disadvantage
Save modified pages to be written at checkpoint	Chunk writes are the most efficient	The checkpoint might last several minutes if you have a large volume of modified data. User response times might significantly deteriorate during the checkpoint.
Write modified pages steadily between checkpoints	Steady performance for the user with no deterioration at checkpoint	The writes between checkpoints are less efficient

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Wait Situations*.

   You can now see how threads in your system are spending their time. If you see waits on checkpoints, you might need to tune the length of checkpoints in your system.

2. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *State on disk* → *Database Message Log*.

   You can now see the length of the checkpoints. If there are no very long waits on checkpoints and the checkpoint duration is below 20 to 30 seconds, then you probably do not need to tune checkpoints.

3. If you need to tune checkpoints, consider the following:

   - Long checkpoints could be caused by an I/O bottleneck. If this is the case, you can tune your asynchronous I/O virtual processors (AIO VPs) or better distribute data and log dbspaces on disk. Refer to Monitoring the I/O Subsystem (Informix) [Seite 450].

   - Long checkpoints can be caused by there being too much data to write at checkpoint time. In this case, reduce the number of buffers that need cleaning, by reducing the value of the parameters `LRU_MAX_DIRTY` and `LRU_MIN_DIRTY` in the Informix `ONCONFIG` file.

     The second measure outlined above – that is, reducing `LRU_MAX_DIRTY` and `LRU_MIN_DIRTY` – is the standard measure for reducing checkpoint duration.

     The SAP defaults for these parameters are 20 and 10 respectively, and it is not uncommon for large systems to have values as low as 5 and 2. Therefore, in a standard

SAP System, pages start being written to disk when 20% are dirty and stop being written when only 10% are dirty. When `LRU_MAX_DIRTY` is set to 5% on a large highly active system, modified pages are written more or less continuously by the AIO VPs.

Monitoring LRUS (Informix)

Use

You can use the SAP/Informix database monitor to monitor the least recently used queues (LRUS) in the Informix database.

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* to see the display for LRUS (Informix) [Seite 480] under *Cleaners* on the main screen of the database monitor.
2. Set the `LRUS` parameter in the Informix `ONCONFIG` file according to SAP Note 41360.

Monitoring CPU VPs (Informix)

Use

You can use the SAP/Informix database monitor to monitor Informix CPU virtual processors (VPs). For up-to-date recommendations, see SAP Note 41360.

Prerequisites

You specify the number of CPU VPs with the `NUMCPUVPS` and `SINGLE_CPU_VP` parameters of the Informix `ONCONFIG` file.

Procedure

- If you have a **single processor** machine

  Do not configure more than one CPU VP. The settings for one CPU VP are `NUMCPUVPS` = 1 and `SINGLE_CPU_VP` = 1.

- If you have a **two processor** machine
  - Normally use only one CPU VP, particularly if you are running an SAP instance on the database server. The correct settings are `NUMCPUVPS` = 1 and `SINGLE_CPU_VP` = 1.
  - If the machine is a standalone database system – that is, dedicated to the database – you might want to experiment by setting `NUMCPUVPS` to 2.

- If you have a **multiprocessor** machine

  There is no simple rule for the number of CPU VPs for multiprocessor machines. You should normally leave some CPU for the operating system and other tasks.

  A good rule is to set `NUMCPUVPS` to the number of physical CPUs minus 1.

  To estimate your CPU load, so that you can see if you need to add a CPU VP:

  - Run the Informix command `onstat -g glo`, wait one minute, then run it again. Subtract the total CPU columns for your CPU VPs to see how much CPU they used in the period.
  - Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Virtual Processor.*

  If the CPUs are all nearly 100% busy, you might benefit by adding a further CPU VP, but do **not** exceed the number of physical CPUs.

For more information, see MULTIPROCESSOR, NUMCPUVPS & SINGLE_CPU_VP (Informix) [Seite 483].

Monitoring the Informix Network Interface

Use

You can use the SAP/Informix database monitor to monitor the Informix network interface. For up-to-date recommendations, see SAP Note 41360.

Prerequisites

The default configuration of the Informix network interface is adequate for small and medium-sized installations. However, if you have more than about 50 SAP work processes, consider tuning the Informix network interface.

Procedure

Tune the Informix network interface using the `NETTYPE` parameter, which has the form (“VP” means “virtual processor”):

```
NETTYPE  <Protocol>, <number of poll threads>, <number of users>,
<VP class>
```

Sub-parameters for the NETTYPE Parameter

Sub-parameter	Meaning
`Protocol`	You can use either `ipcshm` (shared memory for SAP Systems on the database server) or `soctcp` (tcp connection for remote application servers).
`Number of poll threads`	The general rule is one poll thread for 30 to 50 work processes. Each poll thread needs its own VP (so if you only have one CPU virtual processor, you cannot put three poll threads in class CPU).
`Number of users`	Specify the total number of expected work processes (not higher than the `USERTHREADS` parameter in the Informix `ONCONFIG` file).
`VP class`	You can specify either `CPU` or `NET`. `CPU` results in the best performance. If, however, you only have one CPU VP, and you want both shared memory and network connections, you need to use a NET VP for one of the threads (or use TCP for the central system).

Monitoring Virtual Memory Usage (Informix)

Use

You can use the SAP/Informix database monitor to monitor Informix virtual memory usage. Informix uses almost no `malloc` memory. Almost all memory used is shared memory. This means that Informix can sometimes run into operating system limits.

For example, on HP-UX 9.0 the maximum amount of shared memory per process is 750 MB. The data buffers, dictionary cache, administrative memory, and especially all session-specific memory must fit within this limit. Customers with large installations might have problems remaining within the limit.

Reducing shared memory reduces performance and can even increase the possibility of deadlocks. Reduce it only if you are close to the operating system limit for the maximum amount of shared memory.

For up-to-date recommendations, see SAP Note 41360.

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Shared Memory*.

   This brings you to the memory monitor where you can view and, if necessary, optimize session pools.

2. Choose *Pools* or use the Informix command, `onstat -g mem`.

   Here you can see a list of all pools with their used, free and total memory requirements (that is, *used amount, free amount, total amount*). You can recognize the session pools by their session id, given in the *pool name* column.

3. Look at the column *total amount* for the size, given in bytes. Session pools are normally smaller than 5 MB.

4. Intervene in the case of larger session pools. See SAP Note 38307 for a detailed description of what to do.

Monitoring Optimizer Statistics (Informix)

Use

You can use the SAP/Informix database monitor to monitor Informix optimizer statistics. The Informix cost-based optimizer needs information on the data in a table to work out how best to process a query, as shown in the following example:

> Consider a table `EMPLOYEE` which includes the columns `NAME` and `STATE`. Assume there is one index on each of these columns. If a query comes with a `WHERE` clause "`WHERE NAME = 'JONES' and STATE = 'NY'`", how does the database know which index to use? What would happen if all employees lived in New York? If so, it would not make sense to use the `STATE` index.
>
> The update statistics command tells the Informix optimizer about the distribution and selectivity of data in table columns. In this example, the optimizer could then recognize that it is much more efficient to retrieve data with the `NAME` index than the `STATE` index.

Prerequisites

SAP and Informix have developed tools to run update statistics efficiently. These tools are preferable to running update statistics with the Informix tool `dbaccess`. For more information on using these tools, see SAP Note 12184.

Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Table Activity* to display the table activity monitor.

   This provides information on the most recent update statistics run for the tables displayed. If you see a table with a high level of activity and an old date for the last statistics update, this might indicate that the optimizer is choosing the wrong path for querying this table.

2. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *State on Disk* → *Checks* → *Update Statistics* to display information about the statistics update.

   You can display information for a specified table or for all tables. Note that this operation consumes a lot of processing power if you run it for the entire database.

Monitoring SQL Statement (Informix)

Use

You can use the SAP/Informix database monitor to monitor Informix SQL statements. This lets you monitor the statistics for SQL statements, so enabling identification of statements which are “expensive,” that is, consuming an excessive amount of resources.

Running a single SQL statement can sometimes adversely affect performance for all database users. This can happen when the amount of data being searched is very large or when a great deal of processing, such as sorting, needs to be done to the returned set of data. The result is statements which are inefficient in their use of CPU time, database buffers and disk I/Os, leading to degraded performance for all other users.

It is important to monitor SQL statistics to identify these expensive statements and to determine whether action can be taken to improve their performance.

Prerequisites

For more information about what to do **before** you perform this procedure, see Checking SQL Statements (Informix) [Seite 440].

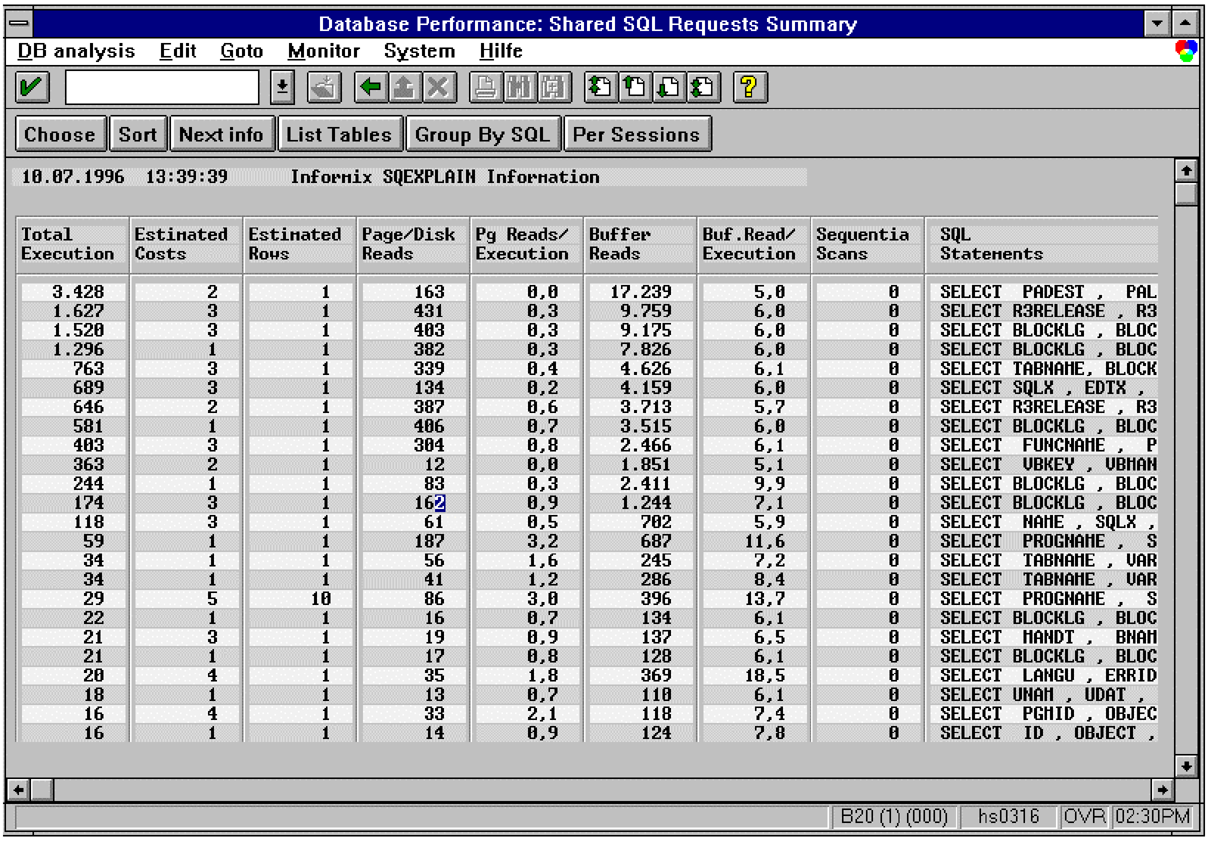
Procedure

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *SQL Statement*.

   The system prompts you for *Selection Criteria* (*Session ID*, *Buffer Reads*, and so on).
2. Enter selection criteria if you want to restrict the display.
3. Choose *Continue*.

   The system displays details of SQL statements, as shown below:

Monitoring SQL Statement (Informix)



Total Execution	Estimated Costs	Estimated Rows	Page/Disk Reads	Pg Reads/ Execution	Buffer Reads	Buf.Read/ Execution	Sequentia Scans	SQL Statements
3.428	2	1	163	0,0	17.239	5,0	0	SELECT PADEST , PAL
1.627	3	1	431	0,3	9.759	6,0	0	SELECT R3RELEASE , R3
1.520	3	1	403	0,3	9.175	6,0	0	SELECT BLOCKLG , BLOC
1.296	1	1	382	0,3	7.826	6,0	0	SELECT BLOCKLG , BLOC
763	3	1	339	0,4	4.626	6,1	0	SELECT TABNAME, BLOCK
689	3	1	134	0,2	4.159	6,0	0	SELECT SQLX , EDTX ,
646	2	1	387	0,6	3.713	5,7	0	SELECT R3RELEASE , R3
581	1	1	406	0,7	3.515	6,0	0	SELECT BLOCKLG , BLOC
403	3	1	304	0,8	2.466	6,1	0	SELECT FUNCNAME , P
363	2	1	12	0,0	1.851	5,1	0	SELECT VBKEY , VBMAN
244	1	1	83	0,3	2.411	9,9	0	SELECT BLOCKLG , BLOC
174	3	1	162	0,9	1.244	7,1	0	SELECT BLOCKLG , BLOC
118	3	1	61	0,5	702	5,9	0	SELECT NAME , SQLX ,
59	1	1	187	3,2	687	11,6	0	SELECT PROGNAME , S
34	1	1	56	1,6	245	7,2	0	SELECT TABNAME , VAR
34	1	1	41	1,2	286	8,4	0	SELECT TABNAME , VAR
29	5	10	86	3,0	396	13,7	0	SELECT PROGNAME , S
22	1	1	16	0,7	134	6,1	0	SELECT BLOCKLG , BLOC
21	3	1	19	0,9	137	6,5	0	SELECT MANDT , BNAM
21	1	1	17	0,8	128	6,1	0	SELECT BLOCKLG , BLOC
20	4	1	35	1,8	369	18,5	0	SELECT LANGU , ERRID
18	1	1	13	0,7	110	6,1	0	SELECT UNAM , UDAT ,
16	4	1	33	2,1	118	7,4	0	SELECT PGMID , OBJEC
16	1	1	14	0,9	124	7,8	0	SELECT ID , OBJECT ,

4. Pay special attention to the following:

– *Total Execution* – total number of times a statement has been executed

– *Page/Disk Reads* – total number of pages read from disk for this statement

– *Pg Reads/Execution* – average number of pages read from disk for this statement per execution

– *Buffer Reads* – total number of pages read for this statement

– *Buf.Read/Execution* – average number of pages read for this statement per execution

5. Select *Sort* to order the display by each of the above areas. This helps to get an overall picture of the types of statements that are often executed.

   A high number of total executions is not necessarily a bad indication since some statements need to be frequently executed

6. Investigate cases where a repeatedly run SQL statement shows a high number of “reads” or “gets” each time it is executed. Check such statements more closely to determine whether indexes are missing or fragmented.

   Perhaps indexes exist on the table, but the SQL statement is written in such a way that it cannot make good use of them. Such expensive SQL statements often reference tables which might benefit from a new secondary index.

The information displayed cannot tell you which user or ABAP program is responsible for an expensive statement. It is sometimes difficult to find which program contains the “select” responsible for the expensive SQL statement.

7. Use the SAP Repository Information System to see a description of the given table and its “where used” list. This normally helps to narrow down your search.

   Since Informix does not use a shared cursor cache, statements can be open redundantly in many sessions. The default view offered by the SQL statement monitor shows information about identical statements summarized in one line for each equivalent statement.

8. Choose *Per Sessions* to expand the view to show all statements individually, even if they contain the same text. *Group By SQL* returns you to the consolidated view.

9. If a statement appears to be performing poorly choose *DB Analysis* → *Explain*.

   This tells you what access method the optimizer chooses to gather the result set for the SQL statement. In the Informix implementation, the path chosen by the optimizer depends on the values of the variables used in the “where” clause (therefore, the optimizer is the “cost-based” type). Unfortunately, the Informix system database from which the information about the SQL statements is obtained does not store these values.

   To enable you to enter reasonable values in the “where” clause of the affected statement, the “Explain” function of the SQL Statement Monitor first presents you the statement in an editor, where you can change the statement as required. When you exit the editor the explain function is performed.

   The explain function is only supported on “select” statements.

Diagnosis of SAP/Informix Performance Problems

Purpose

This section tells you about the diagnosis of performance problems with the Informix database.

Prerequisites

The information given in this section contains suggestions based on SAP's experience with Informix customers. The definitive documentation for Informix performance tuning comes from the database manufacturer, Informix. Use recommendations from the Informix documentation where they differ from any recommendations given here.

Process Flow

1. You identify the problem:
   - Identifying Problems with One Transaction Running Slow (Informix) [Seite 463]
   - Identifying Problems with All Transactions Running Slow (Informix) [Seite 464]
   - Identifying Problems with a Frozen System (Informix) [Seite 465]
2. You find the cause of the problem:
   - Analyzing Sessions and Tables (Informix) [Seite 466]
   - Analyzing Lockwaits (Informix) [Seite 467]
   - Analyzing Lockwaits with onstat (Informix) [Seite 468]

Identifying Problems with One Transaction Running Slow (Informix)

Use

This procedure tells you how to identify problems that **only affects certain users**. If the problem affects all users, see Identifying Problems with All Transactions Running Slow (Informix) [Seite 464].

The normal method is to use the SQL and ABAP trace facilities to trace the transaction that is causing difficulties. Most transaction-specific performance problems are due to inefficient SQL statements or ABAP coding. The trace facilities can almost always help you identify these problems.

This procedure explains how to identify performance problems due to:

- Long running SQL statements
- Lock wait situations
- Missing secondary indexes

Procedure

1. To capture a statement with the SQL trace facility, turn the trace on before a statement goes to the database and do not turn off the trace until the statement has returned.

   Sometimes it is difficult to find long running SQL statements due to the following:

   – Poor SQL statements can take several minutes to return.

   – The SQL trace file wraps around when it is full.

   For more information, see Monitoring Sessions and Tables (Informix) [Seite 466].

2. Use the database monitor to identify performance problems that affect certain users when the cause of the problem is a lock wait situation.

   Sometimes a transaction doing collective processing or a mass data input transaction sends a large number of posting requests on related objects, causing each to wait on locks held by the others.

   For more information, see Monitoring Lockwaits (Informix) [Seite 467].

3. Use the database monitor to compare the definition of secondary indexes requested by SAP developers with the actual indexes on the database.

   Sometimes indexes can get lost as a result of handling errors in copy or reorganization operations. This can cause a severe performance degradation for statements designed to use the index. To determine if any indexes are missing, check the state on disk [Seite 444].

Identifying Problems with All Transactions Running Slow (Informix)

Use

This procedure tells you how to identify problems that **affect all users**. If the problem only affects certain users, see Identifying Problems with One Transaction Running Slow (Informix) [Seite 463].

Procedure

1. Run a basic health check on your database, as described in SAP/Informix Performance Monitoring [Seite 448].
2. Check the wait situations [Seite 434].

   This shows you the state of the database threads. Ideally, the threads should spend more time “running“ than in other states. If *mt yield n* or *ready* is at the top of the list, you might need to add CPU virtual processors (VPs) to the system. If you see *AIO* at the top of the list, see if you can speed up or reduce the quantity of your I/O. You might need to tune your application to improve the situation.

   For more information, see:

   – Monitoring Data Buffers (Informix) [Seite 449]

   – Monitoring the I/O Subsystem (Informix) [Seite 450]
3. If your system normally runs satisfactorily but appears to be slowing down due to a high processing load, check which session – and, therefore, which user and program – is responsible for the most load. Sometimes a single poor program can negatively impact performance for all users.

   For more information, see:

   – Analyzing Sessions and Tables (Informix) [Seite 466]

   – Detailed Database Analysis for Informix [Seite 433]

Identifying Problems with a Frozen System (Informix)

Use

This procedure tells you how to identify the problem when your system has frozen, that is, completely stopped processing. These are the main reasons for a frozen system:

- The logical-log files are full and an emergency backup is needed
- Lockwait situations are occupying all work processes

Procedure

1. Check to see if the logical-log files are full and perform an emergency backup if so:
   a. Check the message log in one of the following ways:
      - Checking the Database Message Log (Informix) [Seite 443] in the database monitor
      - Listing System Information with SAPDBA [Extern]
      - Using `onstat -m` as user `informix`

        If the following message appears, you need to perform an emergency backup:

        Logical Log Files are Full -- Backup is Needed

   b. If necessary, use your normal data recovery tool – `ON-Bar`, `ON-Archive`, or `ontape` – to perform an emergency backup [Extern]. This frees the logical logs, allowing normal system processing to continue.

      To avoid this situation in future, see Preventing Emergency Logical-Log Backup [Extern].
2. Check to see if there is a lockwait situation [Seite 467].

   If lock holders and waiters occupy all work processes, then even users with no interest in the locked records are prevented from working.

Analyzing Sessions and Tables (Informix)

Use

To find the cause of acute Informix database performance problems, you can analyze the database sessions and tables. This is particularly helpful if the problems are caused by errors or bad coding in the application.

Prerequisites

The basic approach is to:

1. Check the activity for the last time period, and sort by relevant statistic values, for example, buffer writes, or (for sessions only) runtime.
2. See which table has the most activity, or which session is causing the highest load on the database. With this information, you can usually determine the user, program, and SQL statement that are responsible for the high resource consumption.

Procedure

To analyze sessions and tables:

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance Menu* → *Database* → *Activity* → *Detail Analysis Menu* → *Informix Session* **or** *Table Activity*.
2. Choose *Reset* to reset the database statistics to zero.
3. Choose *Since Reset* to get values for the period since you chose *Reset* in the previous step.
4. Sort on a performance-relevant value. *Run Time* is a good value for sessions. For tables, interesting values to sort on are *Pagereads* (used by the system when the list is first displayed), *Bufreads*, *Bufwrites* and *Seqscans*.
5. Choose *Current SQL* to find out what a session that is using an excessive amount of resources is doing. Use transaction `SM51` to determine which user and program are currently in the work process and are so responsible for the database activity of the session.
6. If you find a table that is being accessed particularly frequently, check the Informix session display for the most active sessions looking for SQL statements on the table. Then, use transaction `SM51` to find the user and program.

Analyzing Lockwaits (Informix)

Use

This procedure tells you how to find the cause of Informix database performance problems when sessions are waiting on locks held by other sessions, which is a common problem. As a solution, you might need to consider buffering number ranges.

If a program has a problem while holding a lock on a record that is to be updated by many users – for example, a record with an important material number – then all work processes might fill with users who are waiting for that record. This affects even those users who do not need the record, because there are no free work processes.

Procedure

1. Check for exclusive lockwaits [Seite 442].
2. If there is no work process left for the lock monitor itself, use `onstat` instead to determine which process is holding the lock. Refer to Analyzing Lockwaits with onstat (Informix) [Seite 468].

Analyzing Lockwaits with onstat (Informix)

Use

This procedure tells you how to find the cause of Informix database performance problems when sessions are waiting on locks held by other sessions, which is a common problem.

You can normally use the database monitor to identify the cause of this problem, as described in Analyzing Lockwaits (Informix) [Seite 467]. Use the procedure described here to analyze lockwaits with `onstat` if there is no work process left for the monitor itself.

Procedure

1. Enter the command `onstat -u` to display user activity.

   The following example shows the output from `onstat –u` with one user holding a lock and two users waiting on that lock:

address	flags	sessid	user	wait	tout	locks
80705758	---P--D	0	informix	0	0	0
807178e0	Y-BP---	86	sapr3	8414a800	0	3
80717c48	L--PR--	89	sapr3	80886930	-1	2
80718318	L--PR--	88	sapr3	80886930	-1	2
80718680	Y--P---	17	sapr3	842e1b40	0	1

2. Enter the command `onstat -k` to display active locks.

   The following example shows the output from `onstat –k`:

address	wtlist	owner	lklist	type	tblsnum
80885558	0	80718d50	80885418	S	100002
808862f0	0	80717c48	80887470	IX	a001b3
80886908	0	80718318	0	S	100002
80886930	80886930	807178e0	808878a8	HDR+X	a001b3
808869f8	0	80718318	80886908	IX	a001b3
			.		
			.		

3. Link the information in the above displays as follows:

   a. Look for sessions that are waiting on a lock as follows:

   `onstat –u | grep L`

   b. Use the value displayed in the column `Wait` as follows:

   `onstat -k | grep <Wait>`

   c. Use the value displayed in the `Owner` column as follows:

   `onstat -u | grep <Owner>`

This tells you which session is holding the lock.

Checking Key Tuning Parameters (Informix)

Use

This procedure tells you about checking key tuning parameters from the Informix `ONCONFIG` file that influence the performance of the Informix database.

For up-to-date recommendations on Informix parameters, see SAP Note 41360.

Prerequisites

The information given in this section contains suggestions based on SAP's experience with Informix customers. The definitive documentation for Informix performance tuning comes from the database manufacturer, Informix. Use recommendations from the Informix documentation where they differ from any recommendations given here.

Procedure

You check the key tuning parameters in your Informix database:

- ALARMPROGRAM (Informix) [Seite 472]
- BUFFERS (Informix) [Seite 473]
- `CHUNKS`: This parameter is no longer used from Informix version 7.10.UD1.
- CKPTINTVL (Informix) [Seite 474]
- CLEANERS (Informix) [Seite 475]
- DD_HASHSIZE and DD_HASHMAX (Informix) [Seite 476]
- LBU_PRESERVE (Informix) [Seite 477]
- LOGBUFF (Informix) [Seite 478]
- LRU_MAX_DIRTY, LRU_MIN_DIRTY (Informix) [Seite 479]
- LRUS (Informix) [Seite 480]
- LTAPEDEV (Informix) [Seite 481]
- MULTIPROCESSOR (Informix) [Seite 482]
- NETTYPE (Informix) [Seite 484]
- NOAGE (Informix) [Seite 485]
- NUMAIOVPS (Informix) [Seite 486]
- NUMCPUVPS (Informix) [Seite 487]
- PHYSBUFF (Informix) [Seite 488]
- PHYSDBS (Informix) [Seite 489]
- PHYSFILE (Informix) [Seite 490]
- SHMADD (Informix) [Seite 491]
- SHMTOTAL (Informix) [Seite 492]

- SHMVIRTSIZE (Informix) [Seite 493]
- SINGLE_CPU_VP (Informix) [Seite 494]
- USEOSTIME (Informix) [Seite 495]

In MULTIPROCESSOR, NUMCPUVPS & SINGLE_CPU_VP (Informix) [Seite 483] you can find a discussion of the interaction of several parameters.

ALARMPROGRAM (Informix)

The ALARMPROGRAM parameter in the Informix ONCONFIG file defines a user-written program that is executed when an alarm is triggered on the Informix database. The program runs in the background and performs a logical-log backup when the fill-level is reached. This avoids the database stopping when the logs fill up, which might require a difficult and time-consuming emergency backup.

SAP and Informix provide you with alarm programs that you can use if your data recovery tool is ON-Bar or ON-Archive.

If your data recovery tool is ontape you can **not** use these alarm programs.

Use

The alarm program for logical-log backup depends on your data recovery tool:

Data recovery tool	Alarm program	Entry in ONCONFIG file
ON-Bar	log_full.sh	alarmprogram /informix/<SID>/etc/log_full.sh
ON-Archive	logevent.sh	alarmprogram /informix/<SID>/etc/logevent.sh

See also:

Configuring ON-Bar [Extern]

Configuration of ON-Archive [Extern]

Logical-Log File Backup [Extern]

LBU_PRESERVE [Seite 477]

BUFFERS (Informix)

Definition

The `BUFFERS` parameter in the Informix `ONCONFIG` file specifies the maximum number of shared-memory buffers that the database server user threads have available for disk I/O on behalf of client applications. Therefore, the number of buffers that the database server requires depends on the applications. For example, if the database server accesses 15 percent of the application data 90 percent of the time, you need to allocate enough buffers to hold that 15 percent.

The buffers reside in the resident portion of shared memory and are used to cache database data pages in memory. The more buffers are available the more likely it is that a needed page might already reside in memory as the result of previous request.

Use

For up-to-date recommendations, see SAP Notes 41360 and 38307.

Generally, you do not need to change this parameter. Look on the main screen of the database monitor [Seite 425] to check the following:

- *Data Buffers Read Quality* should be greater than > 95%
- *Write Quality* should be greater than 82%

A good minimum buffer size is around 40MB.

It is difficult to offer a single rule here if the quality is not good. The most frequent cause of poor buffer quality is problems in the application. The normal rule is to set `BUFFERS` so that their total size is 20% to 25% of the total physical memory. However, if you are working with a system with strict memory limits, remember that SAP buffers are generally more important than database buffers.

If total physical memory = 100 MB and page size = 2 KB then `BUFFERS` = 10,000.

CKPTINTVL (Informix)

Definition

The `CKPTINTVL` parameter in the Informix `ONCONFIG` file specifies the frequency, expressed in seconds, at which the database server checks to determine whether a checkpoint is needed. When a checkpoint occurs, pages in the shared-memory buffer pool are written to disk.

Use

For up-to-date recommendations, see SAP Note 41360.

Do not set this value too high because this causes very long fast recoveries after power loss or other disruptions. The default SAP value is 900 seconds.

CLEANERS (Informix)

Definition

The `CLEANERS` parameter in the Informix `ONCONFIG` file determines how many threads are started to manage I/O operations. For example, cleaner threads verify that a page has been written to the physical log before it is modified. The cleaner threads run on CPU VPs.

Use

For up-to-date recommendations, see SAP Note 41360.

Refer to Monitoring the I/O Subsystem (Informix) [Seite 450].

DD_HASHSIZE and DD_HASHMAX (Informix)

Definition

The `DD_HASHSIZE and DD_HASHMAX` parameters in the Informix `ONCONFIG` file specify how the hash table of the database server is configured.

Use

For up-to-date recommendations, see SAP Note 38307.

LBU_PRESERVE (Informix)

Definition

The LBU_PRESERVE parameter in the Informix ONCONFIG file reserves the last logical log for administrative tasks by setting the "logs-full high-water mark." When LBU_PRESERVE is enabled, the database server blocks further processing when the next-to-last log fills, rather than the last log. Setting LBU_PRESERVE prevents backups from failing due to lack of logical-log space.

This parameter can help you avoid having to perform an emergency logical-log backup [Extern], which is necessary when the logical-log files are full, and is both difficult and time-consuming.

Use

We recommend you to set this parameter as follows:

```
LBU_PRESERVE        1
```

This means that – when **one** logical-log file is left unfilled – the database stops processing to give you a chance to do a **normal** logical-log file backup. This is easier than doing an emergency backup when all logical-log files are full and the database has stopped processing.

Do **not** rely only on this parameter alone for controlling logical-log backups. You also need to develop an effective approach to logical-log backup. Refer to Preventing Emergency Logical-Log Backup [Extern].

See also:

Editing the ONCONFIG File for ON-Archive [Extern]

ALARMPROGRAM [Seite 472]

LOGBUFF (Informix)

Definition

The LOGBUFF parameter of the Informix ONCONFIG file specifies the size in kilobytes (KB) of each of the three logical-log buffers in shared memory. Triple buffering permits user threads to write to the active buffer while one of the other buffers is being flushed to disk. If flushing is not complete by the time the active buffer fills, the user thread begins writing to the third buffer.

Use

For up-to-date recommendations, see SAP Note 41360.

On the overview screen of the database monitor, check log buffers [Seite 429]:

- If *Log Written/IO* is 75% or less than *Logical log buf* do **not** change LOGBUFF.
- Otherwise, increase LOGBUFF so that *Log Written/IO* is approximately 75%, making sure that the new value is a multiple of page size in KB.

LRU_MAX_DIRTY, LRU_MIN_DIRTY (Informix)

Definition

These parameters from the Informix `ONCONFIG` file have the following meanings

- `LRU_MAX_DIRTY` specifies the percentage of modified pages in the least recently used (LRU) queues that, when reached, flags the queue to be cleaned.
- `LRU_MIN_DIRTY` specifies the percentage of modified pages in the LRU queues that, when reached, flags the page cleaners that cleaning is no longer mandatory. Page cleaners might continue cleaning beyond this point under some circumstances.

Use

For up-to-date recommendations, see SAP Note 41360.

Watch out for the dependence of these parameters on the parameter BUFFERS (Informix) [Seite 473] when you are tuning. The following example shows the effect of increasing `BUFFERS`:

On a system with `LRU_MIN_DIRTY` set to 20 and `BUFFERS` set to 20,000, there could be 4,000 modified pages at checkpoint time.

If you increase `BUFFERS` to 40,000 without changing `LRU_MIN_DIRTY`, there could be 8,000 modified pages at checkpoint time.

See also:

Monitoring Checkpoints (Informix) [Seite 452]

LRUS (Informix) [Seite 480]

LRUS (Informix)

Definition

The LRUS parameter from the Informix ONCONFIG file specifies the number of least recently used (LRU) queues in the shared memory buffer pool. Informix divides its buffers into LRU queues to minimize contention when processes look for free buffers. The number of these queues is set using this parameter.

Use

For up-to-date recommendations, see SAP Note 41360.

You can tune the value of LRUS, in combination with the LRU_MIN_DIRTY and LRU_MAX_DIRTY [Seite 479] parameters, to control how frequently the shared memory buffers are flushed to disk.

Refer to Monitoring LRUS (Informix) [Seite 454].

LTAPEDEV (Informix)

Definition

The LTAPEDEV parameter in the Informix ONCONFIG file is used for logical-log backup but its meaning varies according to whether you use ON-Bar, ON-Archive or ontape for data recovery:

- ON-Bar and ON-Archive

  LTAPEDEV is not actually used by these tools as a path name to a tape device. Instead, it is simply a toggle, that is, any value other than /dev/null (UNIX) or nul (Windows NT) means that the logs can be backed up.

- ontape

  LTAPEDEV specifies the path name to a device used by ontape, onunload, or onload. In the case of ontape, the defined device performs logical-log backup.

Use

Be sure to set LTAPEDEV correctly.

Do **not** set LTAPEDEV to /dev/null on UNIX or nul on Windows NT.

These settings on a productive system are not recommended, because the logical log cannot be backed up. This applies whether you are using ON-Bar, ON-Archive, or ontape.

See also:

LBU_PRESERVE (Informix) [Seite 477]

Configuring ON-Bar [Extern]

Editing the ONCONFIG File for ON-Archive [Extern]

Configuring ontape [Extern]

MULTIPROCESSOR (Informix)

Definition

The `MULTIPROCESSOR` parameter in the Informix `ONCONFIG` file specifies whether the database server performs locking in a manner that is suitable for a single-processor computer or a multiprocessor computer.

With this parameter you can adapt the locking mechanism of the Informix database to your computer's system architecture. For multiprocessor systems, make sure the parameter is not set to zero, since this causes the database to act as if there was only a single CPU.

Use

To determine how to set this parameter, you need to take account of a range of factors. For more information, see:

- SAP Note 41360 for up-to-date recommendations
- MULTIPROCESSOR, NUMCPUVPS & SINGLE_CPU_VP (Informix) [Seite 483].

In general, set this parameter as follows:

- For multiprocessor systems:

  `MULTIPROCESSOR = 1`

- For single-processor systems:

  `MULTIPROCESSOR = 0`

MULTIPROCESSOR, NUMCPUVPS, and SINGLE_CPU_VP (Informix)

Definition

These parameters in the Informix `ONCONFIG` file influence one another and should be considered as a group.

Use

For up-to-date recommendations on these parameters, see SAP Note 41360.

For multi-processor machines, consider whether:

- Kernel-asynchronous I/O (KAIO) is on or off

  Use the database monitor overview [Seite 425] to check whether KAIO is on or off. If KAIO is on, the *Kaio threads* field has a value greater than zero.

- The host machine is running as a:
  - Standalone database, that is, only as a database service
  - Central system, that is, with SAP application services on the same host

See also:

MULTIPROCESSOR (Informix) [Seite 482]

NUMCPUVPS (Informix) [Seite 487]

SINGLE_CPU_VP (Informix) [Seite 494]

NETTYPE (Informix)

Definition

The NETTYPE parameter in the Informix ONCONFIG file provides tuning options for the protocols defined by dbservername entries in the sqlhosts file or the registry, except for the NETTYPE parameter that lets the database server implement multiplexed connections on UNIX.

The parameter NETTYPE in the Informix ONCONFIG file has the form:

```
NETTYPE  Protocol, number of poll threads, number of users, VP
class
```

Use

For up-to-date recommendations, see SAP Note 41360.

The NETTYPE parameter is **not** displayed on the database monitor parameter history list. Use onstat -c to see how it is set.

For more information, see Monitoring the Informix Network Interface [Seite 456].

NOAGE (Informix)

Definition

The `NOAGE` parameter in the Informix `ONCONFIG` file defines how much processing resources are allocated to long-running database processes. However, this parameter is not supported by all operating systems and you can achieve the same effect by tuning the operating system itself.

Use

For up-to-date recommendations, see SAP Note 41360.

From our experience at SAP, we recommend you **not** to use this feature.

- To **not** use this feature (recommended), set the parameter as follows:

```
NOAGE          0
```

- To use this feature (not recommended), set the parameter as follows:

```
NOAGE          1
```

See also:

The Informix release notes in directory `$INFORMIXDIR/release`.

NUMAIOVPS (Informix)

Definition

The `NUMAIOVPS` parameter in the Informix `ONCONFIG` file specifies the number of AIO virtual processors (AIO VPs) to be started when the database is started. To set this parameter correctly, you need to consider whether kernel-asynchronous I/O (KAIO) is implemented on your system.

Use

For up-to-date recommendations, see SAP Note 41360.

1. Use the database monitor overview [Seite 425] to check whether KAIO is on or off. If KAIO is on, the *Kaio threads* field has a value greater than zero.
2. Set the `NUMAIOVPS` parameter as follows:
   – If KAIO is on, set this parameter to 2.
   – If KAIO is off, set this parameter to a suitable value. When set correctly, the last AIO VP should show a considerably smaller workload than the first one. You can check the workload for the AIO VPs by looking at the length of the queues shown by the Informix utility `onstat -g iov`.

See also:

Monitoring the I/O Subsystem (Informix) [Seite 450]

NUMCPUVPS (Informix)

Definition

The NUMCPUVPS parameter in the Informix ONCONFIG file determines how many CPU virtual processors are started when the database is started.

Use

To determine how to set this parameter, you need to take account of a range of factors. For more information, see:

- SAP Note 41360 for up-to-date recommendations
- MULTIPROCESSOR, NUMCPUVPS & SINGLE_CPU_VP (Informix) [Seite 483].
- Monitoring CPU VPs (Informix) [Seite 455]

In general, set this parameter as follows:

- On a single processor system:

```
NUMCPUVPS   1
```

- On a multi-processor system:

  Set NUMCPUVPS to a value less than or equal to the number of physical CPUs on your system.

PHYSBUFF (Informix)

Definition

The PHYSBUFF parameter in the Informix ONCONFIG file specifies the size of the physical log buffers in KB. The default for new R/3 installations is 1024.

Use

For up-to-date recommendations, see SAP Note 41360.

On the main screen of the database monitor [Seite 425], look in the section *under Log Buffers* and check the following ratio:

PhysLog Written/IO divided by *Phys Log Buf* as a percentage

- If this is approximately 75%, do not change PHYSBUFF.
- Otherwise, change PHYSBUFF so that the ratio is approximately 75%, making sure that the new value of PHYSBUFF is a multiple of page size in KB.

PHYSDBS (Informix)

Definition

The `PHYSDBS` parameter from the Informix `ONCONFIG` file specifies the name of the dbspace where the physical log file resides. For standard R/3 installations, `PHYSDBS` is set to **physdbs**.

PHYSFILE (Informix)

Definition

The PHYSFILE parameter in the Informix ONCONFIG file specifies the size of the physical log file in KB.

Use

For up-to-date recommendations, see SAP Note 41360.

If checkpoints are occurring more frequently than the checkpoint interval, you might want to increase the size of the physical log file. You can do this using the Informix ONMONITOR tool.

Do **not** change PHYSFILE with the "vi" editor in the ONCONFIG file.

There must be enough contiguous space in the physdbs dbspace to accommodate the new PHYSFILE size.

SHMADD (Informix)

Definition

The SHMADD parameter in the Informix ONCONFIG file specifies how much shared memory is used to increase the size of the virtual area.

Use

For up-to-date recommendations, see SAP Note 38307.

You should normally set SHMADD to a value high enough to prevent Informix having to add shared memory at all during normal system operation.

The following table shows a general rule for setting this parameter:

Physical Memory Size in MB	SHMADD **Value in KB**
< = 512	16384
> 512	32768

On AIX systems, remember the absolute limit of 10 shared memory segments per process. Therefore, do not set SHMADD too small.

SHMTOTAL (Informix)

Definition

The `SHMTOTAL` parameter in the Informix `ONCONFIG` file specifies the total amount of shared memory. A value of 0 means that there is no limit on memory.

Use

For up-to-date recommendations, see SAP Note 38307.

We recommend that you set `SHMTOTAL` to 0 because otherwise application errors or even a database crash might occur if Informix reaches the memory limit.

SHMVIRTSIZE (Informix)

Definition

The SHMVIRTSIZE parameter in the Informix ONCONFIG file specifies how much shared memory in KB to use for the “virtual portion” at system start. Informix stores various objects here, including the dictionary cache.

The most important factor affecting the size of virtual portion is the session pools. If a large number of users are very active, Informix adds memory to this area. This dynamic allocation of shared memory can affect performance considerably, so you should allocate a large initial segment to avoid dynamic increases.

Use

For up-to-date recommendations, see SAP Note 38307.

You can use the main screen of the database monitor [Seite 425] to check for *Virtual Portion* in the *Shared Memory* display, which shows the value of SHMVIRTSIZE.

SINGLE_CPU_VP (Informix)

Definition

The `SINGLE_CPU_VP` parameter in the Informix `ONCONFIG` file lets you limit the number of CPU virtual processors (CPU VPs) to one.

Use

To determine how to set this parameter, you need to take account of a range of factors. For more information, see:

- SAP Note 41360 for up-to-date recommendations
- MULTIPROCESSOR, NUMCPUVPS & SINGLE_CPU_VP (Informix) [Seite 483].

In general, you set it according to the value of the parameter `NUMCPUVPS`, as follows:

- if `NUMCPUVPS` has the value 1:

  ```
  SINGLE_CPU_VP   1
  ```

- Otherwise:

  ```
  SINGLE_CPU_VP   0
  ```

USEOSTIME (Informix)

Definition

The `USEOSTIME` parameter in the Informix `ONCONFIG` file lets you specify that the database server is to use subsecond granularity when it obtains the current time from the operating system for SQL statements. If you set `USEOSTIME` to nonzero, the performance of your system worsens.

Use

For up-to-date recommendations, see SAP Note 41360.

Normally, set this parameter to 0. However, if you want to monitor SQL statements, set it to 1.

For more information, see Checking SQL Statements (Informix) [Seite 440].

SAP/DB2/400 Database Monitor

Use

You can use the DB2/400 database monitor to perform these tasks:

- Daily monitoring of the database for both test and production systems
- Analysis of certain types of problems
- Database tuning

Integration

The R/3 database monitor for DB2/400 is part of the Computing Center Management System (CCMS).

Prerequisites

The parameter `as4/dbmon/enable` in the instance profile is set to the value 1. The DB2/400 database monitor is started automatically for an R/3 instance provided the parameter has this value. If you change this value to 0, the database monitor is switched off.

Under normal circumstances, you should not change this value. The only situation when we would recommend you to switch off the database monitor is if you have serious performance problems that have already been analyzed and you are waiting for additional hardware to be delivered.

See also:

Data for the Database Monitor (DB2/400) [Seite 497]

SAP/DB2/400 Database Monitor: Detail Analysis Menu [Seite 499]

SAP/DB2/400 Database Monitor: Debugging and Tuning Tips [Seite 517]

Data for the Database Monitor (DB2/400)

The R/3 database monitor for DB2/400 requires data from the memory-based SQL database monitor. This data is only supplied if the database monitor is switched on. Other sources of data are the system catalogs and data stored in OS/400 file objects.

Since all data comes from the database server, you can retrieve the same data from every application server of your R/3 System.

See also:

Checking the Status of the OXDAEDRSQL Job (DB2/400) [Seite 526]

Database Monitor: Entry Screen (DB2/400)

The entry screen of the R/3 database monitor for DB2/400 shows the most important indicators of DB2/400 database performance.

See also:

Database Monitor [Seite 328]

The information is divided into the following categories:

- *Calls*
- *Wait Statistics*
- *Table Scans*
- *Sorts*
- *Index Creates*
- *Temporary Files*
- *Indexes Advised*

To access the entry screen of the R/3 database monitor for DB2/400, choose:

Tools → *Administration* → *Monitoring* → *Performance* → *Database* → *Activity*.

Alternatively, call Transaction ST04.

If you require more detailed information, call the Detail analysis menu [Seite 499].

Detail Analysis Menu (DB2/400)

Features

This menu is divided into three main sections:

Functions for Analyzing Database Activity

Database lock monitor [Seite 500]

Wait situations [Seite 501]

File activity [Seite 502]

DB monitor history [Seite 503]

SQL statements [Seite 504]

50 slowest statements [Seite 504]

Functions for Analyzing Query Activity

Table scans [Seite 505]

Sorts [Seite 506]

Temporary files [Seite 507]

Index advised [Seite 508]

Index usage [Seite 509]

Index creates [Seite 510]

Other Functions

State on disk [Seite 511]

System catalog tables [Seite 514]

Performance tables [Seite 515]

SQL packages [Seite 516]

Activities

To display the *Detail analysis menu*, choose *Tools* → *Administration* → *Monitoring* → *Performance* → *Database* → *Activity* and choose *Detail analysis menu*.

Alternatively, call the transaction ST04 and choose *Detail analysis menu.*

Database Lock Monitor (DB2/400)

Use

The database lock monitor is an online monitoring tool for detecting and analyzing database lock situations. A database lock situation is when several jobs compete for a lock on one (or more) rows of a physical file member with update or read stability intent. One job obtains the lock, the remaining jobs have to wait until this lock is released. In the context of a 3-tier R/3 System, this means that the jobs competing for a lock are the database shadow jobs of the application servers (each work process with a database connection on an application server has one corresponding AS/400 job on the database server: the shadow job). The rows of a physical file member are the records of an R/3 database table (an SQL table is a physical file with one member).

Activities

- To call the DB2/400 database lock monitor, choose *Database lock monitor* in the *Detail analysis menu*.

  Alternatively, choose *Goto* → *Activity* → *Current statistics* → *Database lock monitor* on the database overview screen.

- On the first screen a list of all R/3 application servers currently active is displayed. To display the database shadow jobs for one or more application servers, select the application servers you are interested in. The database shadow jobs are displayed on the next screen. All jobs with the status LCKW are highlighted.

  To display the record logs for a particular job, double-click the job you are interested in. The lock status indicates whether the job is waiting for the record lock to be released (lock status WAIT) or whether it is holding the record lock (lock status HELD).

  To display all jobs that are waiting for a particular record lock to be released (lock status WAIT) and the job actually holding the lock (lock status HELD), select the record lock in question.

  To toggle between the list of record locks for one job and the list of all jobs waiting for or holding a record lock, select a line of the display. Provided the table is known to the ABAP Dictionary, you can display the content of the record currently under lock competition from both lists.

  You can display the AS/400 job log for a selected job from both lists. If a job is an R/3 work process of a shadow job of an R/3 work process, there is a message at the beginning of the job log with message ID CPF9898 detailing which R/3 work process (that is, SAP-internal work process number as listed by Transaction SM50) of which application server is connected to the database.

Wait Situations (DB2/400)

Use

To allow you to locate servers with unusually high wait activity, this function provides an analysis of wait situations for each R/3 instance.

Activities

- To display the wait statistics, choose *Wait situations* in the *Detail analysis menu*.

  Alternatively, choose *Goto* → *Activity* → *Current statistics* → *Wait situations* on the database overview screen.

- On the first screen, the database and non-database wait times are displayed by instance for all the instances currently running. To display the wait times per job for a specific instance, double-click on the instance you are interested in.

File Activity (DB2/400)

Use

This function provides an analysis of activity for all physical files in the R/3 database. This includes statistics on number of OPENs, CLOSEs, INSERTs, UPDATEs, DELETEs, and logical and physical reads performed on the files.

Activities

To see the file activity statistics, choose *File activity* in the *Detail analysis menu*.

Alternatively, choose *Goto* → *Activity* → *Current statistics* → *File system* on the database overview screen.

Database Monitor History (DB2/400)

Use

This function provides an overview of the database workload statistics recorded by the memory-based SQL database monitor.

The database workload statistics consist of the following:

- The number of user calls. That is, the number of SQL statements executed.
- The number of different types of SQL statements executed, for example SELECTs and COMMITs.
- Statistics about how the database executed the statements, for example by means of executed table scans, index creates, sorts, and so on.

Activities

- To display an overview of the database workload statistics recorded by the memory-based SQL database monitor, choose *DB monitor history* in the *Detail analysis menu*.

  Alternatively, choose *Goto* → *Activity* → *Current statistics* → *Database monitor history* on the database overview screen.

- The time periods for which statistics exist are listed on the first screen. To display detailed statistics for a particular time period, double-click on the time period you are interested in. The numbers shown on this screen detail the workload that was recorded between two timestamps.

Timestamp	User calls
08.12.45	351
09.17.32	369
10.17.33	106

369 statements were executed between 08.12.45 and 09.17.32. In the following hour, only 106 statements were executed.

- The database workload statistics program is scheduled automatically to compute new workload statistics. The scheduling report is RSDB4DMP, which is executed by program SAPMSSY6 every NNN seconds where NNN is the value of the profile parameter `rdisp/autoabaptime`.

- You can set the interval between two history dumps using Transaction ST04. See Configuring the Database Monitor (DB2/400) [Seite 521].

- The statistics are not deleted automatically. However, you can delete weekly data from the display (second screen). To schedule automatic deletion of the statistics, configure table SAPWLREORG, setting *ID* to D4 for DB2/400.

SQL Statements (DB2/400)

Features

- *SQL statements*

  This function displays all statements recorded by the memory-based SQL database monitor that correspond to the selection set specified by the user program. It includes statistics on execution time, statement type, sorts, table scans, temporary files, indexes, and dynamic access plan rebuilds per SQL statement.

- *50 slowest statements*

  This function displays the 50 slowest statements recorded by the memory-based SQL database monitor. It includes statistics on execution time, statement type, sorts, table scans, temporary files, indexes, and dynamic access plan rebuilds per SQL statement.

Activities

- To see SQL statement performance statistics, choose *SQL statements* on the *Detail analysis menu*.
- To see the 50 slowest statements, choose *50 slowest statements* on the *Detail analysis menu*.

Table Scans (DB2/400)

Use

This function provides an overview of table scan activity. It shows the tables against which table scans have been performed and specifies the SQL statement that caused the table scan. The OS/400 query optimizer indicates why it chose to perform a table scan and may advise you to create an index to improve performance.

Activities

To display table scan statistics, choose *Table scans* in the *Detail analysis menu*.

Alternatively, choose *Goto* → *Activity* → *Query statistics* → *Table scans* on the database overview screen.

See also:

When Should You Opt for Table Scan Access? (DB2/400) [Seite 518]

Sorts (DB2/400)

Use

This function provides an overview of the sort activity on database files. It shows the time and space required for the sort and lists the SQL statement that caused the sort. The DB2/400 query optimizer explains why it chose to perform a sort and may advise you to create an index to improve performance.

Activities

To display the sort statistics, choose *Sorts* in the *Detail analysis menu*.

Alternatively, choose *Goto* → *Activity* → *Query statistics* → *Sorts* on the database overview screen.

See also:

When Should You Opt for Sort Access? (DB2/400) [Seite 519]

Temporary Files (DB2/400)

Use

This function provides an overview of temporary file creation activity on the database. OS/400 may need to create temporary files for some complex queries. This screen shows the time required to build a temporary file, and the SQL statement that caused the temporary file to be built. The OS/400 query optimizer indicates why it needed to create the temporary file. This information will indicate how the query can be rewritten to avoid a temporary file being created in future.

Activities

To see an overview of temporary file creation activity, choose *Temporary files* in the *Detail analysis menu*.

Alternatively, choose *Goto* → *Activity* → *Query statistics* → *Temporary files* on the database overview screen.

Index Advised (DB2/400)

Use

The OS/400 query optimizer will recommend that you create an index if it considers that this would improve the performance of the query. It provides the names of the fields to be used as key fields when the index is created.

Activities

To display a list of queries for which the DB2/400 query optimizer recommends creation of an index, choose *Index advised* in the *Detail analysis menu*.

Alternatively, choose *Goto* → *Activity* → *Query statistics* → *Index advised* on the database overview screen.

Index Usage (DB2/400)

Use

Indexes are used to implement queries that contain record selection, joins, ordering or grouping.

This function provides an overview of indexes used to implement queries on the database. The OS/400 query optimizer indicates which indexes were used to implement queries and why the indexes were used.

Activities

To find out which indexes have been used to implement queries on the database, choose *Index usage* in the *Detail analysis menu*.

Alternatively, choose *Goto* → *Activity* → *Query statistics* → *Index usage* on the database overview screen.

Index Creates (DB2/400)

Use

This function provides an overview of the temporary index create activity on database files. OS/400 may create temporary indexes while performing queries if no permanent indexes exist to fulfill the requirements. This function shows the tables the indexes were created over and the SQL statement that caused the index to be created. The OS/400 query optimizer indicates why it needed to create the index, gives the names of the fields used to create the index, and may recommend that you build a permanent index to improve performance.

Activities

To display an overview of temporary index create activity, choose *Index creates* in the *Detail analysis menu*.

Alternatively, choose *Goto* → *Activity* → *Query statistics* → *Index creates* on the database overview screen.

See also:

Temporary Indexes versus Permanent Indexes (DB2/400) [Seite 520]

State on Disk (DB2/400)

Use

This function provides the following information:

- File statistics for tables and indexes in the R/3 database library.
- R/3 tables, indexes, and views missing from the R/3 database library or the ABAP Dictionary.
- Current disk status (available and used disk space)

Features

- *Refresh file statistics*: Performs a new file statistics analysis either in the background or online.
- Consistency checks [Seite 512]:
  - Displays R/3 tables with missing primary keys
  - Displays all objects missing from the R/3 database library or ABAP Dictionary
  - Displays the result of checking the R/3 kernel with the ABAP Dictionary (transaction SICK)
- *Detailed object analysis*: More detailed information on file statistics
- *Deleted row analysis* (for all tables): Displays how much space is reserved by deleted rows. You can use this information to decide which tables to reorganize to free disk space.
- *List damaged files*: Checks the R/3 database library for damaged files and display any that are found.
- *Missing indexes*: Displays all indexes (including primary keys) that are missing from the R/3 database library or ABAP Dictionary.
- *Space statistics*: Displays size and record count history for tables by day, week or month.

Activities

To see the state on disk information, choose *State on disk* in the *Detail analysis menu*.

Alternatively, choose *Goto* → *State on disk* on the database overview screen.

See also:

Finding Database Storage Problems (DB2/400) [Seite 522]

Performing Deleted Records Analysis (DB2/400) [Seite 524]

Finding Database Damage (DB2/400) [Seite 523]

Consistency Checks (DB2/400)

Use

Use this function to find out whether tables or indexes are missing from the database or from the ABAP Dictionary. You can also use these checks to determine whether missing indexes are primary or secondary indexes.

Activities

To find out how many tables and indexes are missing from the database and from the ABAP Dictionary, go to the *State on Disk* screen. To call this screen, choose *Goto* → *State on disk* in the database overview screen or choose *State on disk* on the *Detail analysis menu*.

To see whether indexes are missing from the ABAP Dictionary or from the database (and if so, which indexes are involved), choose *Missing indexes*.

To find out whether the indexes involved are primary or secondary indexes, choose *Consistency checks* on the *State on Disk* screen. A dialog box providing the following options is displayed:

- *Find database tables without primary keys*

  This option checks for database tables without indexes. If you discover that indexes are missing, use the consistency check function to find out whether they are primary or secondary indexes.

- *Database<->ABAP Dictionary*

  This option checks for inconsistencies between the database objects and the ABAP Dictionary. If you select this option, a second dialog box appears. It displays the types of objects that could be missing and the number missing for each type of object.

  Select the object type you are interested in and click *OK*. A report listing the objects of this type that are missing is displayed.

  If a primary index is missing from the database, contact SAP for help with restoring the index. Primary indexes ensure that row keys are unique. There is no simple procedure for restoring a primary index if duplicate keys have been created in the table concerned.

  If a secondary index is missing from the database, you can restore it using ABAP Dictionary functions (transaction SE11). These indexes are used for performance purposes only, which means that missing indexes do not represent a risk.

  If a view is missing from the database, you can restore it using ABAP Dictionary functions (transaction SE11). Missing views can cause some SAP applications to fail.

  If a table is missing from the database, check whether the objects involved are test dictionary objects that were possibly never meant to be in the database.

  If objects are missing from the ABAP Dictionary, they may be system objects that do not require an entry in the Dictionary. If you find that a table or index from an SAP application is missing from the ABAP Dictionary, please contact SAP for assistance.

- *SAP kernel<->ABAP Dictionary*

  The *SAP kernel<->ABAP Dictionary* option checks the consistency of the kernel with the ABAP Dictionary. If errors are found, contact SAP for assistance.

System Catalog Tables (DB2/400)

Use

This function displays information about the R/3 tables, indexes, and views in the DB2/400 database. For detailed descriptions of the system catalog views and their fields, see the IBM documentation *DB2 for OS/400 SQL Programming* (document number: SC41-4611).

Activities

To display the AS/400 system catalog tables, choose *System catalog tables* in the *Detail analysis menu*.

Alternatively, choose *Goto* → *Activity* → *Current statistics* → *System catalog tables* on the database overview screen.

Performance Tables (DB2/400)

Use

This function displays the R/3 performance statistics tables from which the database monitor gets its data.

This is an expert view of raw data from the internal tables used by transaction ST04.

Activities

To display database performance tables, choose *Performance tables* in the *Detail analysis menu*.

Alternatively, choose *Goto* → *Activity* → *Current statistics* → *Performance tables* on the database overview screen.

SQL Packages (DB2/400)

Use

This function displays a list of all the libraries containing SQL packages.

Activities

- To display SQL package information, choose *SQL packages* in the *Detail analysis menu*.

  Alternatively, choose *Goto* → *Activity* → *Current statistics* → *SQL packages* on the database overview screen.

- Select a library to display all the SQL packages in the library on a second screen.
- To display detailed information about a particular package, select the package you are interested in.

DB2/400 Database Monitor: Debugging and Tuning Tips

When Should You Opt for Table Scan Access? (DB2/400)

Table scan access implies that all records in a file are read. The records are processed in the order in which they reside in the file; hence, no ordering is guaranteed unless ordering is specified in the query. As each record is read, selection criteria (if any) are applied by the operating system, and only the records that match the criteria are returned to the application.

Table scan access provides good performance when approximately 20% or more of the records in the file match the selection criteria.

If the file contains a large percentage of deleted records, consider reorganizing the file to remove the deleted records to further improve performance.

Table scan access may not be the most efficient access method if only a small percentage (< 20%) of the records of a relatively large file (>2,000 records) is selected. In this case, you should consider creating an index with key fields that match the record selection fields.

When Should You Opt for Sort Access? (DB2/400)

Sort access is used to order records. Queries that may require ordering include those that specify ORDER BY, UNION or DISTINCT. When data is ordered, another access method is used to read and select the data. The result data set is then ordered by OS/400.

If a sort is performed because no index exists that can be used to satisfy the ordering, the size of the result set determines whether sort is the best ordering solution. If more than 20% of the records are selected and then sorted, sort provides good performance. If less than 20% of the records are selected and then sorted, consider creating an index that can be used to select and sort the records.

Temporary Indexes versus Permanent Indexes (DB2/400)

The DB2/400 optimizer can dynamically create temporary indexes to implement queries. Indexes can be created to implement queries that specify joins, ordering or grouping.

There are two types of index creates:

- An index can be created over a file.
- An existing index can be used to speed up selection during the creation of the new index. This type of index create is referred to as an “index-from-index” create.

When the DB2/400 optimizer creates a temporary index, this index can only be used for the query and work process (job) that “created” it. When the associated cursor is fully closed, the system deletes the index. Indexes created by the optimizer are “sparse” indexes. This means that all selection that can be applied at the time the index is created will be built into the index. These indexes, once created, usually perform very well. However, there is some up-front cost involved in creating the index.

The query optimizer will advise which key fields were used to create the temporary index. A permanent index could be created to avoid temporary index creates the next time the query is run. Consider the cost of creating and maintaining a permanent index as compared to the savings made by running the query with the permanent index before creating any additional indexes.

Configuring the Database Monitor (DB2/400)

Use

You can change the name of the server on which the database workload statistics are to run or change the time interval between regular or history statistics dumps.

Procedure

1. From the main screen of the database monitor (transaction ST04), choose *Edit → DB monitor configuration → Change*.
2. In the *Server name* field, enter the system name of the server on which the database workload statistics are to run.
3. In the group box *Regular database monitor dump*, in the *Time interval* field, enter the time interval you require between regular dumps.
4. In the *Time unit* field, enter `H` for hour(s) or `M` for minute(s) as required.
5. In the group box *History database monitor dump*, in the *Time interval* field, enter the time interval you require between regular dumps.
6. In the *Time unit* field, enter `H` for hour(s) or `M` for minute(s) as required.

Result

Your settings are saved. To return to the original settings, choose *Return to default setting*.

Finding Database Storage Problems (DB2/400)

1. Go to the *State on Disk* screen. To call this screen, choose *Goto* → *State on disk* on the database overview screen or choose *State on disk* in the *Detail analysis menu*.

   The total number of tables, constraint indexes, and other indexes is displayed. The total size of each type is also displayed. When combined, these sizes give you the total amount of space being used by the R/3 database.

   At the bottom of the screen, the total ASP (auxiliary storage pool) size and the percentage used is displayed. If the percentage used is nearing 95%, you must perform system cleanup to free up space or add more storage to the ASP.

2. To view size history information for tables, choose *Space statistics*. You can choose between a daily, weekly or monthly view on the data.
3. To view current size information for tables, choose *Detailed object analysis*. This report allows you to choose a table and view the indexes over the table and their sizes. You can also view the size history of the table or the table structure.

Finding Database Damage (DB2/400)

Procedure

1. Go to the *State on Disk* screen. To call this screen, choose *Goto* → *State on disk* in the database overview screen or choose *State on disk* in the *Detail analysis menu*.
2. Choose *List damaged files*. This function checks each object in the database for possible damage.

Result

Any damaged objects encountered will be listed in a report, along with the type of damage involved. If a file is damaged, consult SAP to find out how to proceed.

Performing Deleted Records Analysis (DB2/400)

Procedure

1. Go to the *State on Disk* screen. To call this screen, choose *Goto → State on disk* in the database overview screen or choose *State on disk* in the *Detail analysis menu*.
2. Choose *Deleted row analysis*.

Result

Any tables where more than 20% of the rows have been deleted are listed in a report, with the number of rows and the percentage deleted.

Consider reorganizing these tables. Reorganizing the tables removes the deleted rows and saves disk space. If a table is being used by R/3, you might not be able to reorganize it while the R/3 System is running.

Analyzing Wait Situations (DB2/400)

You can analyze Wait Situations [Seite 501] per application server. This lets you determine which application servers are being made to wait the longest for resources such as database locks.

Checking the Status of the QXDAEDRSQL Job (DB2/400)

Use

If some data is not being supplied to the database monitor, check the status of the job `QXDAEDRSQL`.

Procedure

1. Check that the job `QXDAEDRSQL` is active in the `QSYSWRK` subsystem. The job will probably have the status **`SELW`**.
2. If you do not find the job, enter the command **`STARTSAP *DB`** at OS/400 level.

SAP/DB2 UDB Database Monitor

Use

You can use the SAP/DB2 UDB database monitor to check the status and performance of the database system.

Integration

All the information provided by the SAP/DB2 UDB database monitor is based on the diagnostic information that the DB2 Universal Database system supplies via the Database System Monitor. Therefore, you can use the DB2 UDB database monitor to obtain critical information that you could also call directly via the DB2 Universal Database command line processor.

Since all data comes from the database server, you can retrieve the same data from every application server in your R/3 System.

Features

For detailed explanations of all parameters, see the *DB2 UDB System Monitor Guide and Reference V6* (SC09-2849-00) and the *DB2 UDB Administration Guide V6* (SC09-2840-00)

See Also

Using the DB2 UDB Database Monitor [Seite 528]

SAP/DB2 Universal Database Performance Monitoring Strategies [Seite 564]

Using the DB2 UDB Database Monitor

To start the SAP/DB2 UDB database monitor, call transaction ST04.

The SAP/DB2 UDB for UNIX & Windows: Database Snapshot [Seite 529] screen appears.

- To close and open the navigation frame, choose *Toggle*.
- To refresh the values shown on the screen, choose *Refresh*.
- To reset the values to 0, choose *Reset*.
- To analyze the time period since you last pressed *Reset* at a later time, choose *Since Reset*.
- To display the values that have accumulated since you started the database, choose *Since DBM start*.

  This logic has been implemented in all display transactions offering the functions *Reset*, *Since Reset* and *Since DBM start*.

To monitor the physical status of the database, use the tables and indexes functions. To access these functions, choose *Space* → *Tables and Indexes* or call transaction DB02.

DB2 UDB for UNIX & Windows: Database Snapshot Screen

The *Database Snapshot* screen provides an overview of critical database performance indicators, such as buffer pool activity, I/O, extended storage, logging, lock situations, calls (user activity), sorts, and caching data.

The information on the screen is arranged in the following order:

- Buffer pool
- Cache
- Asynchronous I/O
- Direct I/O
- Extended storage
- Locks and deadlocks
- Logging
- Calls
- Sorts

The system displays values collected since the database was started. If the database is shut down, the values are deleted.

The displayed values are not really meaningful until the database has been running for several days. The longer the database has been running, the more useful the values.

To access more detailed information, choose any of the following entries under *Performance* in the navigation frame:

- Database
- Buffer Pools
- Tablespaces
- Tables
- Applications
- SQL Cache
- Lockwaits
- Deadlocks

Buffer Pool – Overview (DB2 UDB)

For an overview of buffer pool information, call transaction ST04.

1. Choose *Performance* → *Database* in the navigation frame.

   The *Database Snapshot* screen appears.

2. Choose the *Buffer Pool* tab.

Field	Description
Total buffer pool size	Overall size of all buffer pools in KB. This is one of the most important values for performance optimization.
Overall buffer quality	Indicates the frequency with which the data or index data is read from the buffer pool, rather than directly from the hard disk. This is calculated using the formula: (1-physical reads / logical reads) * 100
Avg. phys. read time	Average time required to read data from disk into the buffer pool.
Avg. phys. write time	Average time required to write data from the buffer pool to disk.
Data – Logical reads	Read accesses to data in the buffer pool.
Data – Physical reads	Read accesses to data on disk (I/O).
Data – Physical writes	Write accesses to data on disk (I/O).
Index – Logical reads	Read accesses to index data in the buffer pool.
Index – Physical reads	Read accesses to index data on disk (I/O).
Index – Physical writes	Write accesses to index data on disk (I/O).
Data – Synchronous writes	Write accesses to data not in the buffer pool.
Data – Synchronous reads	Read accesses to data not in the buffer pool.

Data and index data is read/written in pages. A page can be 4k, 8k, 16k, or 32k in size.

Unless otherwise specified, no distinction is made between synchronous and asynchronous accesses.

Buffer Pool – Information (DB2 UDB)

To display information on buffer pools, call transaction ST04.

1. Choose *Performance* → *Buffer Pools* in the navigation frame.

   The *Buffer Pool* screen appears.

2. Choose the *Buffer Pool* and *Asynchronous I/O* tabs.

3. For detailed information on buffer pools, choose *Detail*.

Field	Description
Buffer Pool	
Buffer pool name	Name of the buffer pool
Data - Logical reads	Number of read accesses to data in the buffer pool
Data - Physical reads	Number of read accesses to data that was not in the buffer pool and that first had to be read from disk.
Data – Physical writes	Number of write operations from data pages to the disk.
Index – Logical reads	Number of read accesses to index data in the buffer pool.
Index – Physical reads	Number of read accesses to index data that is not in the buffer pool and that first had to be read from disk.
Index – Physical writes	Number of write operations from index data pages to disk.
Avg. phys. read time	Average time required for writing to the buffer pool (reading from disk)
Avg. phys. write time	Average time required for writing from the buffer pool onto disk
Asynchronous I/O	
Data – Async. physical reads	Number of data pages that were read asynchronously from disk and written to the buffer pool (I/OServers)
Data – Async. physical writes	Number of asynchronous write accesses of data pages from buffer pool to disk (I/OCleaners)
Index – Async. physical reads	Number of index pages that were read asynchronously from disk and written to the buffer pool (I/OServers)
Index – Async. physical writes	Number of asynchronous write accesses of index pages from buffer pool to disk (I/OCleaners)
Avg. async. physical read time	Average time required for asynchronous reading from disk into the buffer pool.

Buffer Pool – Direct Access (DB2 UDB)

To display information on buffer pools – direct access, call transaction ST04.

1. Choose *Performance* → *Buffer Pool* in the navigation frame.

   The *Buffer Pool* screen appears.

2. Choose the *Direct I/O* and *Extended Storage* tabs.

Field	Description
Direct I/O	
Direct reads	Read accesses to disk that do not use the buffer pool (LONG VARCHAR fields, backup)
Direct writes	Write accesses to disk that do not use the buffer pool (LONG VARCHAR, RESTORE, LOAD)
Avg. direct reads	Average number of requests to read directly from disk
Avg. direct writes	Average number of requests to write directly to disk
Avg. direct read time	Average time required to read directly
Avg. direct write time	Average time required to write directly
Extended Storage	
Pool data – To extended storage	Number of data pages in the buffer pool rolled out to extended storage
Pool index – To extended storage	Number of index pages in the buffer pool rolled out to extended storage
Pool data – From extended storage	Number of data pages returned from extended storage to the buffer pool
Pool index – From extended storage	Number of index pages returned from extended storage to the buffer pool

By default extended storage is not activated. The number of storage segments is defined via the database parameter `num_estore_segs`. The size of a segment is defined via the database parameter `estore_seg_sz`. Before a buffer pool can use extended storage, it must have been linked to extended storage via the `ALTER BUFFERPOOL <name> EXTENDED STORAGE` command. It normally only makes sense to use extended storage if sufficient main storage is available.

I/O Servers and I/O Cleaners (DB2 UDB)

To display information on I/O servers and I/O cleaners, call transaction ST04.

1. Choose *Performance* → *Database* in the navigation frame.

   The *Database Snapshot* screen appears.

2. Choose the *Asynchronous I/O* tab.

Field	Description
Number of I/O servers	I/O servers read data asynchronously from the hard disk into the buffer pool.
Avg. async. physical read time	Average time required by the I/O servers to read a page from disk and write it into the buffer pool.
Number of I/O cleaners	I/O cleaners write data asynchronously from the buffer pool to the hard disk.
Avg. async. physical write time	Average time required by the I/O cleaners to read a page from the buffer pool and write it to the hard disk.
Data – Async. physical reads	Number of data pages read asynchronously (prefetch).
Index – Async. physical reads	Number of index pages read asynchronously (prefetch).
Data – Async. physical writes	Number of data pages written asynchronously.
Index – Async. physical writes	Number of index pages written asynchronously.

Data is read/written in pages. A page can be 4k, 8k, 16k, or 32k in size.

Locks and Deadlocks (DB2 UDB)

To display information on locks and deadlocks, call transaction ST04.

1. Choose *Performance* → *Database* in the navigation frame.

   The *Database Snapshot* screen appears.

2. Choose the *Locks & Deadlocks* tab.

Field	Description
Lock list size	Database locks are managed in a list. This parameter determines the maximum length of the list (database configuration parameter `LOCKLIST`). The lock list is allocated dynamically.
Lock list in use	The current size of the lock list.
Lock escalations	If the maximum allowed length of the lock list is reached, row locks are converted to table locks to create space in the lock list. This process is called "lock escalation".
Excl. lock escalations	Like lock escalation with the distinction that exclusive row locks are converted to an exclusive table lock. An exclusive table lock means that no other application is able to access the table.
Total lock waits	An application that has to wait because of a lock is in "lock wait" state.
Time waited on locks	The total time applications waited until a lock was released.
Locks currently held	The total number of locks currently held on the database.
Deadlocks detected	The number of deadlocks that have been resolved. Deadlock situations are recognized and resolved automatically by the database. The database configuration parameter `DLCHKTIME` determines when a lock wait situation is resolved.
Lock timeouts	The number of lock wait situations resolved by the database. The database configuration parameter `LOCKTIMEOUT` determines when a lock wait situation is resolved.

Calls (DB2 UDB)

To display information on calls, call transaction ST04.

1. Choose *Performance* → *Database* in the navigation frame.

   The *Database Snapshot* screen appears.

2. Choose the *Calls* tab.

Field	Description
Commits	Number of SQL COMMIT statements executed
Rollbacks	Number of SQL ROLLBACK statements executed by applications. Automatic rollbacks caused by error situations or deadlocks are not included.
Select SQL	Number of SELECT commands executed
Update/Insert/Delete	Number of commands executed modifying data
Rows deleted	Number of data records deleted
Rows inserted	Number of data records inserted
Rows selected	Number of data records selected
Rows updated	Number of data records changed

Sorts (DB2 UDB)

To display information on sorts, call transaction ST04.

1. Choose *Performance* → *Database* in the navigation frame.

   The *Database Snapshot* screen appears.

2. Choose the *Sorts* tab.

Field	Description
Total sort heap size	Maximum allowed size of the storage area used for sorting. This is determined by the database configuration parameter SORTHEAP.
Sort heap allocated	Current size of the storage area
Total sorts	Number of sort processes
Sort overflows	If the storage area allocated for sorting is not large enough, a "sort overflow" occurs. The hard disk is then used temporarily.
Active sorts	Currently active sort processes using the sort heap
Total sort time	Total time required for all sort processes

Cache (DB2 UDB)

To display information on the cache, call transaction ST04.

1. Choose *Performance* → *Database* in the navigation frame.

   The *Database Snapshot* screen appears.

2. Choose the *Cache* tab.

Field	Description
Catalog cache size	Maximum allowed size for the catalog cache. The catalog cache is accessed each time a transaction accesses a table, view or alias. The cache is allocated dynamically from the heap. The maximum allowed size is determined by database configuration parameter `CATALOGCACHE_SZ`.
Catalog cache quality	The ratio of lookups to inserts.
Package cache size	The maximum allowed size for the package cache. The package cache contains access plans. The maximum allowed size is determined by database configuration parameter `PCKCACHESZ`.
Package cache quality	The ratio of lookups to inserts.
Catalog cache lookups	The number of attempts to read from the catalog cache
Catalog cache inserts	The number of attempts to write to the catalog cache
Catalog cache overflows	The number of failed write attempts (due to the cache being full)
Catalog cache heap full	The number of failed write attempts (due to the heap being full)
Package cache lookups	The number of attempts made by an application to read from the package cache
Package cache inserts	If the cache does not contain an access plan, it is loaded into the cache.

Tables (DB2 UDB)

To display information on tables, call transaction ST04.

1. Choose *Performance* → *Tables* in the navigation frame.

   The *Table Snapshot* screen appears.

This overview displays for all tables the number of rows read, the number of rows written, the number of accesses to datasets that have been moved out of the page due to overflow (*Overflow Access*) and page reorganizations (*Page reorgs*).

Pay particular attention to the column *Overflow Access*. If the value in this column is very high, you should consider reorganizing the table.

Tablespaces (DB2 UDB)

To display information on tablespaces, call transaction ST04.

1. Choose *Performance* → *Tablespaces* in the navigation frame.

   The *Tablespace Snapshot* screen appears.

This screen displays buffer pool activity and direct access information for each tablespace defined for the R/3 database.

Tablespaces – Buffer Pool [Seite 540]

For information on buffer pool access, choose *Detail* and choose the *Buffer Pool* and *Asynchronous I/O* tabs.

Tablespaces – Direct Access [Seite 542]

The *Direct I/O* and *Extended Storage* tabs give information on direct accesses, in other words, I/O activity that does not use the buffer pool (for example, access to long varchar columns, backup).

Tablespaces – Buffer Pool (DB2 UDB)

To access the *Tablespace Snapshot – Buffer Pool* screen, call transaction ST04.

1. Choose *Performance* → *Tablespaces* in the navigation frame.

   The *Tablespace Snapshot* screen appears.

2. Choose the *Buffer Pool* and the *Asynchronous I/O* tab.

Field	**Description**
Buffer Pool	
Name	Name of the tablespace
Data – Logical reads	The number of logical read requests for data pages that have gone through the buffer pool.
Data – Physical reads	The read requests that required I/O to get data pages into the buffer pool. The value includes the number of physical reads that were performed synchronously (by the database manager agents).
Data – Physical writes	The number of times a buffer pool data page was physically written to disk. It includes the number of physical writes that were performed synchronously (by the database manager agents).
Index – Logical reads	The number of logical read requests for index pages that have gone through the buffer pool.
Index – Physical reads	The number of physical read requests to get index pages into the buffer pool. This value includes the number of synchronously read index pages.
Index – Physical writes	The number of times a buffer pool index page was physically written to disk. It includes the number of physical writes performed synchronously (by the database manager agents).
Avg. phys. read time	The average amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool (includes asynchronous reads).
Avg. phys. write time	Provides the average amount of time spent physically writing data or index pages from the buffer pool to disk (includes asynchronous writes).
Asynchronous I/O	
Data – Async. physical writes	The number of times a buffer pool data page was physically written to disk. It specifies the number of physical writes that were performed asynchronously (by the I/OCleaners).
Data – Async. physical reads	Specifies how often a data page in the buffer pool is read physically from the hard disk. The value refers to the physical read operations that were performed asynchronously (by the I/OServers).

Index – Async. physical writes	The number of times a buffer pool index page was physically written to disk. It includes the number of physical writes performed asynchronously (by the I/OCleaners).
Avg. async. physical read time	The average elapsed time spent reading by database manager prefetchers.
Avg. async. physical write time	The average elapsed time spent writing data or index pages from the buffer pool to disk by database manager page cleaners.
Async. physical reads	The number of asynchronous read requests.

Tablespaces – Direct Access (DB2 UDB)

To display information on tablespaces – direct access, call transaction ST04.

1. Choose *Performance* → *Tablespaces* in the navigation frame.

   The *Tablespace Snapshot* screen appears.

2. Choose the *Direct I/O* and the *Extended Storage* tabs.

Field	Description
Direct I/O	
Direct reads	The number of read operations that do not use the buffer pool
Direct writes	The number of write operations that do not use the buffer pool.
Avg. direct read time	The elapsed time (in milliseconds) required to perform the direct reads.
Avg. direct write time	The elapsed time (in milliseconds) required to perform the direct writes.
Extended Storage	
Pool data – To extended storage	Indicates the number of data pages copied from the buffer pool to extended storage to create space in the buffer pool.
Pool index – To extended storage	Indicates the number of index pages copied from the buffer pool to extended storage to create space in the buffer pool.
Pool data – From extended storage	Indicates the number of data pages copied back from extended storage to the buffer pool because they are needed again.
Pool index – From extended storage	Indicates the number of index pages copied from extended storage to the buffer pool because they are needed again.

Applications (DB2 UDB)

To display the application snapshot, call transaction ST04.

1. Choose *Performance* → *Applications* in the navigation frame.

   The *Applications Snapshot* screen appears.

The table displays performance data for every DB2 application, that is, for every SAP work process. The information displayed helps you to determine which work processes are placing the highest load on the database.

For detailed information, choose *Detail*.

In addition to general information, information is displayed on:

- Buffer pool activity
- Current lock situations
- SQL statements
- Cache activity
- Direct I/O
- Extended Storage
- Calls
- Sorts
- Times
- Agents

Applications Snapshot – Applications [Seite 544]

Applications - Buffer Pool [Seite 546]

Applications – Locks & Deadlocks [Seite 547]

Applications - SQL Statements [Seite 548]

Applications - Cache [Seite 549]

Applications Snapshot – Applications (DB2 UDB)

To display the *Applications Snapshot – Applications* screen, call transaction ST04.

1. Choose *Performance* → *Applications* in the navigation frame.

   The *Application Snapshot* screen appears.

2. Choose the *Application* and the *Times* tab.

Field	Description
Application	
Handle	Several agent processes (DB2 agent) can be assigned to an "agenthdl".
Application PID	Process ID of the database process belonging to an SAP work process.
Client PID	Process ID of an SAP work process.
Host	Host name of the application server that the work process is running on.
Status	Current status of the application. Values for this field are: - Database Connect Pending - Database Connect Complete - Unit of Work Executing - Unit of Work Waiting - Lock Wait - Commit Active - Rollback Active - Recompiling - Compiling - Request Interrupted - Database Disconnect Pending - Transaction Prepared - Transaction Heuristically Rolled Back - Transaction Ended - Creating Database
Times	
User CPU (s), System CPU (s)	The CPU values provide the time (in seconds) used by the database manager agent process. They can help you understand the level of activity within the application, and may help you identify applications that could benefit from additional tuning. The values include CPU time for both SQL and non-SQL statements executed by the application.

Idle time	Time in seconds since the last action of an application

Applications - Buffer Pool (DB2 UDB)

To display the applications snapshot – buffer pool, call transaction ST04.

1. Choose *Performance* → *Applications* in the navigation frame.

   The *Applications Snapshot* screen appears.

2. Choose the *Buffer Pool* and the *Extended Storage* tab.

Field	Description
Buffer Pool	
Data – Logical reads	The number of logical read requests for data pages that have gone through the buffer pool.
Data – Physical reads	The read requests that required I/O to get data pages into the buffer pool. The value includes the number of physical reads that were performed synchronously (by the database manager agents) and asynchronously (by the I/OServers).
Data – Physical writes	The number of times a buffer pool data page was physically written to disk. It includes the number of physical writes that were performed synchronously (by the database manager agents) and asyncronously (by the I/OCleaners).
Index – Logical reads	The number of logical read requests for index pages that have gone through the buffer pool.
Index – Physical reads	The number of physical read requests to get index pages into the buffer pool. Since index data is read only by the database manager agents, this value contains the number of syncronously read index pages.
Index – Physical writes	The number of times a buffer pool index page was physically written to disk. It includes the number of physical writes that were performed synchronously (by the database manager agents) and asynchronously (by the I/OCleaners).
Extended Storage	
Pool data – To extended storage	The number of data pages copied from the buffer pool into extended storage to create space in the buffer pool.
Pool index – To extended storage	The number of index pages copied from the buffer pool into extended storage to create space in the buffer pool.
Pool data – From extended storage	The number of data pages copied back from extended storage to the buffer pool because they are needed.
Pool index – From extended storage	The number of index pages copied back from extended storage to the buffer pool because they are needed.

Applications – Locks & Deadlocks (DB2 UDB)

To display the *Applications Snapshot – Locks & Deadlocks* screen, call transaction ST04.

1. Choose *Performance* → *Applications* in the navigation frame.

   The *Applications Snapshot* screen appears.

2. Choose the *Locks & Deadlocks* tab.

Field	Description
Locks currently held	The total number of locks currently held by the application.
Total lock waits	The total number of times that this application requested a lock but had to wait because another application was already holding a lock on the data.
Time waited on locks	The total amount of elapsed time that this application has waited for a lock to be granted.
Lock escalations	The number of times that locks have been escalated from several row locks to a table lock.
Excl. lock escalations	The number of times that locks have been escalated from several row locks to one exclusive table lock or an exclusive lock on a row caused the table lock to become an exclusive lock. Exclusive locks are important to track since they can impact the concurrency of your data because other applications cannot access data held by an exclusive lock.
Deadlocks detected	The total number of deadlocks that has occurred.

Applications – Statements (DB2 UDB)

To display the *Applications Snapshot – Statements* screen, call transaction ST04.

1. Choose *Performance* → *Applications* in the navigation frame.

   The *Applications Snapshot* screen appears.

2. Choose the *Calls* and the *Statement* tab.

Field	Description
Calls	
Commits	The total number of SQL COMMIT statements that has been attempted.
Rollbacks	The total number of SQL ROLLBACK statements that has been attempted.
Dynamic SQL	The number of dynamic SQL statements attempted.
Static SQL	The number of static SQL statements attempted.
Failed SQL	The number of SQL statements that was attempted, but failed.
Select SQL	The number of SQL SELECT statements that was executed.
DDL	The number of SQL Data Definition Language (DDL) statements that was executed (a few examples of DDL statements are CREATE TABLE, CREATE VIEW, ALTER TABLE, and DROP INDEX).
Update/Insert/Delete	The number of SQL UPDATE, INSERT, and DELETE statements that was executed.
Statement	
Operation	Operation currently being processed or most recently processed. Possible operations are: - PREPARE - EXECUTE - EXECUTE IMMEDIATE - OPEN - FETCH - CLOSE - static COMMIT - static ROLLBACK
SQL statement	Text of the dynamic SQL statement that was being processed when the snapshot was taken. It can also be the text of the statement that was most recently processed, if no statement was being processed at the time when the snapshot was taken.

Applications – Cache (DB2 UDB)

To display the *Applications Snapshot – Cache* screen, call transaction ST04.

1. Choose *Performance* → *Applications* in the navigation frame.

   The *Applications Snapshot* screen appears.

2. Choose the *Cache* tab.

Field	**Description**
Package Cache – Lookups	The number of times an application looked for a section in the package cache.
Package Cache – Inserts	The total number of times that a request section was not available for use and had to be loaded into the package cache.
Catalog Cache – Lookups	The number of times that the catalog cache was referenced to obtain table descriptor information.
Catalog Cache – Inserts	The number of times that the system tried to insert table descriptor information into the catalog cache.
Catalog Cache – Overflows	The number of times that an insert into the catalog cache failed due to the catalog cache being full.
Catalog Cache – Heap Full	The number of times that an insert into the catalog cache failed due to a heap-full condition in the database heap.

Applications – Agents (DB2 UDB)

To display the *Applications Snapshot – Agents* screen, call transaction ST04.

1. Choose *Performance* → *Applications* in the navigation frame.

   The *Applications Snapshot* screen appears.

2. Choose the *Agents* tab.

Field	Description
Coordinator Agent PID	Coordinator agent process ID.
Associated with this appl.	Number of agents participating in this application.
Stolen from application	The number of agents removed from this application and subsequently used by another application. (This only happens if the agent was not busy.)

SQL Cache Snapshot (DB2 UDB)

Use

The information contained in the SQL cache snapshot gives you an overview of SQL statements that are executed very often from the SQL cache in your system. You can identify those statements that consume an excessive amount of resources and can determine whether to fine tune statements to improve performance.

Features

To display the SQL snapshot, call transaction ST04 and choose *Performance* → *SQL Cache* in the navigation frame.

The SQL Cache Snapshot screen appears.

At the top of this screen the following information is displayed under *SQL Cache Snapshot*:

- *System*

  SAP System ID
- *Partition*

  Number of the partition
- *DBM Start*

  Last start of the database manager
- *Last Snapshot*

  Date and time of the last snapshot (if one was taken)

The following functions are available:

- *Toggle*

  Closes and opens the navigation frame on the left.
- *Refresh*

  Updates the snapshot for the partition you have selected. When this is executed successfully, *Last Snapshot* displays the current system date and time.
- *Display…*

  If a snapshot already exists, that is, if you have already chosen *Refresh* at least once, you can display the last snapshot taken.

  Depending on your system, the snapshot can give you a wide range of extensive information, which might lead to a very large result set.When you choose Display, the *Selection Criteria* dialog box appears. You can limit the result set displayed in the table according to the following selection criteria:

  – *Executions*

    Number of times a statement has been executed

- *Total Execution Time (ms)*

  Total execution time for a statement in milliseconds

- *Rows Read*

  Number of rows read for a statement

- *Rows Written*

  Number of rows written for a statement

- *SQL Text (Case Sensitive)*

  Search using either the wild card * or using a text string, for example, INSERT, to limit the number of statements displayed

When you have made your selections and chosen *OK*, the result set is displayed in the table.

If no hits are found, the result set is empty and nothing is displayed.

The screen now also displays information under *Total Cache Sum* on the total execution time (*Exec Time ms*), total number of rows read (*Rows Read*) and total number of rows written (*Rows Written*) in the entire SQL cache regardless of the selection criteria.

The following functions are also activated: *EXPLAIN*, *Sort* and *Export*…

- *EXPLAIN*

  You select a line in the table and select *EXPLAIN* for more detailed information on the selected SQL statement.

- *Sort*

  You select a column in the table and select *Sort Up* or *Sort Down* to order the display in the table.

- *Export*

  You can export the snapshot, that is, you can save it locally in a file, by selecting *Export*. You can then process the data using a table calculation program, for example, and can compare different snapshots from the SQL cache.

The table displayed gives you detailed information on the following:

- *SQL Text*

  Text of a dynamic SQL statement that was in the SQL cache at the time of the snapshot

- *Executions*

  Number of times a statement has been executed. This value helps you to identify which SQL statements are executed very often. A high number of executions does not necessarily mean that a statement is using an excessive amount of resources. You should also check the number of rows read and rows written. If you find relatively high values here, you should use *EXPLAIN* to check whether indexes are not being efficiently used or whether indexes are missing.

- *Total Execution Time (ms)*

Total execution time for an SQL statement in milliseconds. You can use this value together with *Executions* to identify the statements that would benefit from further analysis.

- *% Total Execution Time*

  Total Execution Time (ms) divided by *Total Cache Sum Exec Time (ms)*

- *Total User CPU Time (ms)*

  Total user CPU time for an SQL statement in milliseconds. This value together with the total execution time gives you information on the longest running statements.

- *Total System CPU Time (ms)*

  Total system CPU time for an SQL statement in milliseconds. This value together with total execution time and total user CPU time helps you to identify statements that use an excessive amount of resources.

- *Rows Read*

  Number of rows read. You can use this value to identify statements that would benefit from additional indexes. Use *EXPLAIN* to analyze the statement.

  The given value does not necessarily correspond to the number of rows of the result set of the SQL statement. The Rows Read value is the number of rows that needs to be read in order to obtain the result set.

- *% Rows Read*

  Rows Read divided by *Total Cache Sum Rows Read*

- *Rows Written*

  Number of rows that were changed (inserted, deleted or modified) in a table. High values might indicate that you should update statistics using `RUNSTATS`.

- *% Rows Written*

  Rows Written divided by *Total Cache Sum Rows Written*

- *SQL Sort*

  Number of sorts that were necessary to execute the statement. You can use this value to determine whether new indexes are needed. Use *EXPLAIN* to check whether and which indexes were used when the statement was executed.

Lockwaits (DB2 UDB)

To display information on lockwaits, call transaction ST04.

1. Choose *Performance* → *Lockwaits* in the navigation frame.

   The *Lockwaits* screen appears.

This function searches for lockwaits indicating that at least one process is locked by the lock on another process. A request waits for a resource (a database table or a row of a table) that is locked exclusively by another user. The process causing the lock and the waiting process (there might only be one) are displayed.

Field	Description
Appl. handle holding	The agenthdl of the application holding the lock
Appl. handle waiting	The agenthdl of the application waiting for the locks to be released
Agent PID holding	Process ID of the coordinator agent of the application holding the lock
Agent PID waiting	Process ID of the coordinator agent of the application waiting for the lock to be released
Appl. name holding	Name of the application holding the lock
Appl. name waiting	Name of the application waiting for the lock to be released
Lock object type	The type of lock held
Lock holding	The form of lock held
Lock waiting	The form of lock that the waiting application would like to set. The following forms are possible: IS: intention share lock IX: intention exclusive lock S: share lock SIX: share with intention exclusive lock X: exclusive lock IN: intent none Z: super exclusive lock U: update lock NS: next key share lock NX: next key exclusive lock W: weak exclusive lock NW: next key weak exclusive lock
Table schema	Part of the table name
Table name	Table on which/on whose record the lock is held
Tablespace	The tablespace containing the table concerned
Lockwait start time	Time at which the exclusive lock was set
Lockwait start date	Date when the exclusive lock was set
Partition	Partition ID where the lock is being held

The fields *Agent PID holding*, *Agent PID waiting*, *Partition*, *Lockwait start date*, *Lockwait start time*, *Table schema* and *Tablespace* are only displayed when you select *Detail*.

Lockwait situations are recognized by DB2 UDB. The database parameter LOCKTIMEOUT specifies how many seconds the system is to wait before automatically resolving a lockwait situation.

If LOCKTIMEOUT is set to -1, lock wait situations are not resolved.

DB2 UDB Diag Log (DB2 UDB)

To access the *DB2 UDB Diag Log* screen, call transaction ST04.

1. Choose *Diagnostics* → *DB2 UDB Diag Log* in the navigation frame.

   The *DB2 UDB Diag Log* screen appears.

DB2 Universal Database uses the DB2 UDB system log (SYSLOG) to log errors and warning conditions. Entries are added to the SYSLOG based on priority and based on the facility that caused the error or warning condition. For example, security, the kernel and system daemons can cause entries to be logged in the SYSLOG. DB2 is represented by the facility called 'user'. The priority refers to the urgency of the message and can be one of the following (listed from highest to lowest priority as used by DB2 Universal Database):

- Alert (not available for Windows NT)
- Error (not available for Windows NT)
- Warning
- Information

You use the *DB2 UDB Diag Log* function to display the most recent entries of the log file `db2diag.log`. This file is located in the directory specified by the diagpath and contains errors logged by DB2 and some diagnostic information. In detail, the `db2diag.log` contains the following information:

- The location of the error being reported
- A diagnostic message explaining the reason for the error
- Any available supporting data, such as SQLCA data structures and pointers to the location of any extra dump information.

Depending on the value set for database parameter `DIAGLEVEL` (default value 3), the size of the file `db2diag.log` may rapidly increase to several MBs. This can result in slow response times when you attempt to display the message log. Therefore, when the file reaches 5 MB in size, you should save it in a different directory and then delete the original file.

Other Logs

You can display the alert messages written by DB2 Universal Database.

Choose *Diagnostics* → *DB2 UDB Alert Log* in the navigation frame.

You can also display the *Dump directory*, which displays the contents of the directory specified by the Diagnostic Data Directory Path (diagpath). This path is configured within the Database Manager Configuration.

The directory will contain dump files and an alert log file, the Diag log and may contain error logs.

Choose *Diagnostics* → *Dump Directory* in the navigation frame.

Deadlocks (DB2 UDB)

To display information on deadlocks, call transaction ST04.

1. Choose *Performance* → *Deadlocks* in the navigation frame.

   The *Deadlocks* screen appears.

The DB2 UDB database event monitor records the occurrence of deadlocks. These are situations where two or more applications lock each other, each waiting for the other(s) to release the lock. Deadlock situations are detected automatically by DB2 UDB and resolved after a specified time period. This time period is determined by database configuration parameter `DLCHKTIME`. Information on the SAP work processes involved, on the database manager agent processed, and detailed lock information on the objects involved in the deadlock is provided.

For technical reasons, two event monitors are used. These are then switched on and off alternately, so that one monitor is always active. The only reliable information is that recorded by the monitor that has just been switched off. This information is displayed when the function is called.

You can switch between the event monitors using the *Change Event Monitor* function. When you switch between event monitors, displayed information is lost. For this reason, it is essential to save information on deadlocks. To do this, choose *Save to a local file* to download the displayed information to your computer.

CLP Structures (DB2 UDB)

To access the *CLP Structures* screen, call transaction ST04.

1. Choose *Performance* → *CLP Structures* **or** *Configuration* → *CLP Structures* in the navigation frame.

   The *CLP Structures* screen appears.

From this screen you can select the output from the DB2 database system monitor from a list box.

The initial screen for the first access option (*Performance* → *CLP Structures*) is a database snapshot, and for the second access option (*Configuration* → *CLP Structures*) is the database configuration. Apart from this feature, both functions are the same

The following snapshots are available:

- Database Manager Configuration
- Database Configuration
- Database Manager Snapshot
- Database Snapshot
- Application Snapshot
- Lock Snapshot
- Table Snapshot
- Tablespace Snapshot
- Buffer Pool Snapshot

These functions display the snapshot data in the same way as the corresponding DB2 UDB command line processor calls.

Parameter Changes (DB2 UDB)

To display information on parameter changes, call transaction ST04.

1. Choose *Configuration* → *Parameter Changes* in the navigation frame.

   The *Parameter Changes* screen appears.

This screen displays current and previous settings of the DB2 UDB database manager configuration parameters and the DB2 UDB database configuration parameters, together with the respective date of change.

By comparing the database history with the parameter changes, you can see the effects of parameter changes.

- To display the current values of the parameters, choose *Active Parameters*.
- To see how the parameters were set in the past, choose *History of file*.
- To restrict the time period to be displayed, choose *Select period*.

The entry `db` in the second column indicates that a parameter is one of the database configuration parameters.

The entry `dbmgr` in the second column indicates that a parameter is one of the database manager configuration parameters.

Performance History (DB2 UDB)

To display information on performance history, call transaction ST04.

1. Choose *Performance* → *Performance History* in the navigation frame.

   The *Performance History* screen appears.

This screen provides day-by-day trend analysis of database activity. You can emphasize peak periods (select *Peaks*), display delta values (select *Interval*) after clicking on the required day and display these graphically (select *Graph by column*).

If you select a particular day, snapshots of the database activity are displayed in a two-hour cycle. In the header of this overview screen, two other dates appear in addition to the selected day, so that you can immediately select one of these days for further analysis, if required.

In addition to information on call statistics [Seite 561], information about lock statistics [Seite 562] is displayed.

Performance History: Call Statistics (DB2 UDB)

To display information on performance history (call statistics), call transaction ST04.

1. Choose *Performance* → *Performance History* in the navigation frame.

   The *Performance History* screen appears, containing information on call statistics.

Field	Description
day	Day
date	Date
endtime	Latest time at which the data collector collected performance data.
log reads	Read accesses to data in the buffer pool.
phy reads	Read accesses to data on the hard disk (I/Os).
writes	Write accesses to data on the hard disk (I/Os).
ix log reads	Read accesses to index data in the buffer pool.
ix phy reads	Read accesses to index data on the hard disk (I/Os).
ix writes	Write accesses to index data on the hard disk (I/Os)
commits	Number of SQL COMMIT statements executed.
rollbacks	Number of rollbacks executed by applications. This does not include automatic rollbacks caused by error situations or deadlocks.

Performance History: Lock Statistics (DB2 UDB)

To display information on performance history (lock statistics), call transaction ST04.

1. Choose *Performance* → *Performance History* in the navigation frame.

   The *Performance History* screen appears.

2. Choose *More* to display the lock statistics.

Field	Description
day	Day
date	Date
endtime	Latest time at which the data collector collected performance data.
lock waits	An application waiting due to a lock has the status *Lock wait*.
wait time	The total time the applications waited before a lock was released.
deadlocks	The number of deadlocks that were resolved. Deadlock situations are recognized automatically and resolved by the database.
lockesc	When the maximum size of the lock list has been reached, row locks are converted to table locks. This is referred to as lock escalation.
xlockesc	Corresponds to the lock escalation (*lockesc*) described above. However, in this case exclusive row locks are converted into an exclusive table lock. An exclusive table lock means that no other application can access the table.

Logging Parameters (DB2 UDB)

To display information on logging parameters, call transaction ST04.

1. Choose *Backup and Recovery* → *Logging Parameters* in the navigation frame.

   The *Logging Parameters* screen appears.

This screen provides information about the logging parameters configured, such as size of log files or Log Retain Status. Furthermore, you can check the space available of the file systems where your system's log archive and the `log_dir` directory are stored.

In a production system, Log Retain must be switched on. If it is not switched on, you risk losing data if serious database problems occur.

If Log Retain is not switched on, there is no Log Archive Path on your system and no information can be displayed for this. The range from the first to the next active log file specifies the number of log files containing active (that is, incomplete) transactions.

Full log files are copied from `log_dir` to `log_archive`. They are not deleted from `log_dir` until they cease to be active. This means that there are two copies of a full active log file, one in `log_dir` and one in `log_archive`.

SAP/DB2 Universal Database Performance Monitoring Strategies

For more information on performance monitoring strategies, see the following:

Checking the Size of the Database Buffers – DB2 UDB [Seite 565]

State on Disk – DB2 UDB [Seite 573]

Checking Consistency – DB2 UDB [Seite 572]

Checking the Size of the Database Buffers (DB2 UDB)

Database buffers are critical for the operation of the database. These are:

- Buffer pool
- Lock list
- Sort heap
- Catalog cache
- Package cache

You must make sure that these buffers are dimensioned adequately. You can display statistics for the buffers on the *Database Snapshot* screen.

To access this screen, call transaction ST04 and choose *Performance* → *Database* from the navigation frame.

Checking Buffer Performance

There are a number of rules you can use to find out whether the buffers are large enough and function correctly. If your buffers do not conform to these rules, you should increase their size:

Checking the Buffer Pool Hit Ratio – DB2 UDB [Seite 566]

Monitoring Lock List Utilization – DB2 UDB [Seite 567]

Monitoring the Sort Heap – DB2 UDB [Seite 568]

Monitoring Cache Quality – DB2 UDB [Seite 569]

Checking the Buffer Pool Hit Ratio (DB2 UDB)

The buffer pool hit ratio should lie above 95 % in a newly started up system.

If this is not the case, you should consider increasing the size of the buffer pool provided you have sufficient main storage available. (See the *Overall Buffer Quality %* value on the DB2 UDB *Database Snapshot* screen on the *Buffer Pool* tab.)

Too many prefetchers can generate unnecessary asynchronous I/Os, so it is important to watch hit ratios, as well as buffer pool size, number of I/O servers and number of I/O cleaners. (See the I/O servers and I/O cleaners on the *Asynchronous I/O* tab.

Changing the Size of the Buffer Pool

To change the buffer pool size, adjust database configuration parameter `buffpage`.

As of DB2 Version 5, you can create one or more buffer pools in the database. In this case, check each buffer pool. To do this, choose *Performance* → *Buffer Pools* in the navigation frame.

Changing the Number of I/O Servers and I/O Cleaners

To change the number of I/O servers and I/O cleaners, adjust database configuration parameters: `num_ioservers`, `num_iocleaners`.

Monitoring Lock List Utilization (DB2 UDB)

On the DB2 UDB *Database Snapshot* screen, the size of the lock list and the proportion of the lock list in use are shown.

You may consider increasing the size of the `locklist` database configuration parameter if the lock list utilization is high or if lock escalations occur. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application. "Escalation" means that locks held on individual data records are replaced by a lock on the whole table, creating space in the lock list. A table lock prevents other applications from accessing the table. This can lead to "lock wait" situations.

If lock escalations occur, check which applications have set the locks. It is possible that the applications may not release the locks (for example, in the case of batch imports or if commits occur too rarely).

The amount of lock list space available is determined by the `locklist` and `maxlocks` database configuration parameters. When an application reaches the maximum number of locks allowed and when there are no more locks to escalate, the application will use space in the lock list allocated to other applications. When the entire lock list is full, an error occurs.

Monitoring the Sort Heap (DB2 UDB)

The value of *Sort Overflows* on the *Database Snapshot* screen indicates the total number of sorts that ran out of sort heap [Seite 571] and required disk space for temporary storage. A sort occurs, for example, when a select accesses several tables.

When a sort overflows, data is written to disk to create space in the sort heap. This disk I/O results in reduced performance for the sort. Use this value in conjunction with the *Total Sorts* value to calculate the percentage of sorts that had to overflow to disk. If this percentage is high, you may want to increase the value of the `sortheap` database configuration parameter. The value of *Sort Heap allocated* shows the total number of allocated pages of sort heap space for all sorts at the time the database snapshot was taken. The amount of memory allocated for each sort may be some or all of the available sort heap size.

Monitoring Cache Quality (DB2 UDB)

Catalog Cache Size	Indicates the maximum amount of database heap space that the catalog cache can use. The catalog cache is referenced whenever a table, view or alias name is processed during the compilation of an SQL statement. It is dynamically allocated from `dbheap`, as required, until the *Catalog Cache Size* is reached.
Package Cache Size	Amount of application heap memory to be used for caching a package's static and dynamic SQL statements. As of DB2 Version 5 each database agent accesses a global cache.
Quality	Tells you whether or not the package or catalog cache or is being used effectively. If the hit ratio of the package or catalog cache is greater than 95%, the cache is performing well.

A lower value indicates that the cache size of the package cache or the catalog cache should be increased (use the database configuration parameters `pckcachesz` and `catalogcache_sz`).

(The package cache has a higher impact on the performance of the system than the catalog cache.)

Buffer Pool (DB2 UDB)

To access the *Buffer Pool Snapshot* screen, call transaction ST04.

1. Choose *Performance* → *Buffer Pools* in the navigation frame.

   The *Buffer Pool Snapshot* screen appears.

The buffer pool is the area of storage into which database pages are read and in which they are changed.

Buffer pool pages are written to disk either by the database manager agents (synchronous writes) or by I/O cleaners (asynchronous writes).

Buffer pool pages are read from disk synchronously (by database manager agents) or are prefetched asynchronously by I/O servers. Activities of the I/O cleaners and I/O servers are monitored and correlate with the buffer pool data.

The table displays selected information for each buffer pool in your R/3 database.

To display all available information on a buffer pool, choose *Detail*.

If you are running your system with several buffer pools, you should use this display to monitor performance, rather than using the *DB2 UDB Database Snapshot* screen, where the values displayed are the aggregates over all buffer pools.

Sort Heap (DB2 UDB)

The amount of memory available for each sort, as defined in the database configuration parameter `sortheap`.

Checking Consistency (DB2 UDB)

To find out whether tables or indexes are missing from either the database or the ABAP Dictionary:

1. Call transaction ST04 and choose *Space* → *Tables and Indexes* in the navigation frame.

   The *Database performance: Tables and Indexes* screen is displayed.

2. Choose *Missing indexes* in the *Tables and Indexes* section of this screen to display the names of the indexes missing from the database or ABAP Dictionary.

3. You can check both the internal consistency of the database system and the consistency of the database system with the ABAP Dictionary.

   The following check functions are available:

– **Database consistency**

  Tables without indexes

  Indexes without tables

– **Database and R/3 System**

  Missing database tables

  Missing database objects

  Objects missing from the ABAP Dictionary

4. To carry out consistency checks, choose *Checks*.

5. The *Database checks* dialog box is displayed. Some important checks provided are:

   Missing unique indexes

   R/3 Kernel

   Select *Tools* → *Administration* → *Administration* → *Installation check* or call transaction SICK.

   Database <-> SAP Dictionary consistency

State on Disk (DB2 UDB)

To monitor the physical state of the database, use the *State on Disk* functions. To access these functions, select *Tools → Administration → Monitoring → Performance → Database → Tables/Indexes* or call Transaction DB02.

Field	Description
Date/time of tablespace analysis	Date and time of the last analysis run
Date/time of table/index analysis	Date of oldest table/index statistics for all tables
Total number (Tablespaces)	Number of tablespaces in the database
Total size/KB (tablespaces)	Total size of all tablespaces
Total free/KB (tablespaces)	Free space within the allocated area (absolute/relative in %)
Minimum free/KB (tablespaces)	Smallest amount of space available in one of the tablespaces (absolute/relative in %)
Total number (tables and indexes)	Total number of tables/indexes
Total size/KB (tables and indexes)	Total size of tables/indexes (this value comprises only the net data)
Missing on database	Tables/indexes missing from the database (but existing in DDIC)
Missing in R/3-ddic	Tables/indexes missing from DDIC (but existing in the database)

- To update the information on this screen, choose *Refresh*. When you do this, a tablespace analysis and consistency check are carried out. If you choose *Include current statistics analysis* on the next screen, all tables/index values are recalculated on the basis of the existing table statistics.

  Selecting this option can increase *Refresh* execution time to more than an hour.

- Select *Checks* to display the screen Checking Consistency (DB2 UDB) [Seite 572]
- Space Statistics: Database (DB2 UDB) [Seite 574]
- State on Disk: Tablespaces (DB2 UDB) [Seite 577]
- Space Statistics: Tablespace (DB2 UDB) [Seite 575]
- State on Disk: Tables and Indexes (DB2 UDB) [Seite 579]
- Space Statistics: Tables and Indexes (DB2 UDB) [Seite 576]

Calculating table values with outdated statistics can result in inexact values. To calculate update statistics including calculation of table sizes, use the DBA Planning Calendar [Seite 1244] (Transaction DB13).

Space Statistics: Database (DB2 UDB)

To display the space statistics information for the database, choose *Tools* → *Administration* → *Monitoring* → *Performance* → *Database* → *Tables/Indexes* (or call Transaction DB02) and choose *Space statistics* in the *Database system* group box.

This function gives an overview of the growth of the database. It shows how the values for tablespaces and tables/indexes change.

The screen is divided into two tables. The first table contains the total space allocated for the database:

Field	Description
Scale	Daily, weekly or monthly overview
Values	Total or delta overview
DB Size	Space occupied in the database
DB Free	Free space in the database
%-Used	Occupied space as a percentage of the total size of the database

The second table gives an overview of the size of the net data in the database. The individual values correspond to the values on the screen *Database performance: tables and indexes*.

Field	Description
Scale	Daily, weekly or monthly overview
Values	Total or delta overview
Tables Total	Number of tables in the database
Tables Size	Total size of the tables
Indexes Total	Number of indexes in the database
Indexes Size	Total size of the indexes

To switch to a screen displaying the delta values, select *Db history* → *Delta values*. This screen shows how the values change over the course of a day. To switch back to the screen displaying the total values, select *Db history* → *Total values*. These statistics are recorded daily.

Space Statistics: Tablespaces (DB2 UDB)

To display the space statistics information for the tablespaces, choose *Tools* → *Administration* → *Monitoring* → *Performance* → *Database* → *Tables/Indexes* (or call Transaction DB02) and choose *Space statistics* in the *Tablespaces* group box.

The screen *History of tablespaces* gives information about the growth of the tablespaces. By default, only those tablespaces are shown where changes have occurred. To display all the tablespaces, choose *History* → *Tablespaces on/off*.

Field	Description
Scale	Daily, weekly or monthly overview
Tablespace	Name of the tablespace
Total Used Size	Space currently occupied
Chg/day	Average change in size per day (measured since recording started)
Total Free Size	Space currently free
Chg/day	Average change in size per day (measured since recording started)
Total %-Used	Percentage of the total size occupied
Chg	Percentage by which the amount used has changed.
Total Container	Number of containers belonging to the tablespace
Chg/day	Average change in number per day (measured since recording started)
Total Tables/Indexes	Number of tables/indexes belonging to the tablespace.
Chg/dy	Average change in number per day (measured since recording started).

To see more detailed information on a tablespace, select the tablespace you are interested in and choose *Select*. A table is displayed that shows the data collected for the tablespace arranged by day. To move through the list of tablespaces, select *Next tablespace* or *Prev. tablespace*.

These statistics are updated daily.

Space Statistics: Tables and Indexes (DB2 UDB)

To display the space statistics information for the tables and indexes, choose *Tools → Administration → Monitoring → Performance → Database → Tables/Indexes* (or call Transaction DB02) and choose *Space statistics* in the *Tables and indexes* group box.

This function allows you to select a number of tables. Information is then displayed about the growth of these tables. If you select all tables (by entering *), by default only those tables are listed whose size has changed. If you need to display all the tables, choose *History → All Objects on/off*.

Field	**Description**
Object Name	Name of the table or index
Type	Table or index
Total Size	Size of the table/index
Chg/day	Average change in size per day (measured since start of recording)
Tablespace	Tablespace to which the object belongs

To see more detailed information, choose an object and then select menu option *Select*. A second table is displayed. This table shows each day for which data was recorded separately together with the data that was recorded. To move within this display from one object to another, select *Next table/index* or *Prev. table/index*. These statistics are recorded on a weekly basis (on Sundays).

State on Disk: Tablespaces (DB2 UDB)

To display the state on disk information for the tablespaces, choose *Tools* → *Administration* → *Monitoring* → *Performance* → *Database* → *Tables/Indexes* (or call Transaction DB02) and choose *Detailed analysis* in the *Tablespaces* group box.

This function gives you more information about the tablespaces that have been created. You can get an up-to-date analysis at any time by selecting *DB Analysis* → *Refresh*. This only has a minimal effect on your system performance and takes less than a minute.

Field	Description
Tablespace	Name of the tablespace
Type	**D**atabase **M**anaged **S**pace or **S**ystem **M**anaged **S**pace. At present, SMS can only be used for PSAPTEMP.
Used Size	Space occupied
Free Size	Free space
%-Used	Percentage share of the occupied space compared to the total size of the tablespace.
Container	Number of containers belonging to the tablespace.
Status	Status of the tablespace when the analysis takes place.

The final line of the table contains the total values.

The total size of a tablespace is calculated by adding together the values for *Used Size* and *Free Size*.

For a more detailed analysis of a tablespace, position the cursor on the tablespace you are interested in and choose *Analysis*. The following options are available:

- *Containers*

  Lists all containers belonging to the selected tablespace. The information displayed includes the size and name of the tablespace.

  You can update this information online at any time by selecting *Refresh*. This only has a minimal effect on your system performance and takes less than a minute.

- *Tables and Indexes*

  Lists all tables or indexes belonging to a tablespace.

- *Detailed Analysis*

  Shows detailed information about the selected tablespace. This information includes:

  - Number of pages
  - *Page size*
  - *Extent size*
  - *Prefetch size*

- *Number of containers*
- Space occupied (*Used size*)
- Free space (*Free size*)
- *High water mark* (the greatest amount of space used to date)

You can get an up-to-date analysis at any time by selecting *DB Analysis* → *Refresh*. This only has a minimal effect on your system performance and takes less than a minute.

- *History*

Shows how the size of the selected tablespace has changed. For more information, see Space Statistics: Tablespace (DB2 UDB) [Seite 575].

State on Disk: Tables and Indexes (DB2 UDB)

To display the state on disk information for the tables and indexes, choose *Tools* → *Administration* → *Monitoring* → *Performance* → *Database* → *Tables/Indexes* (or call Transaction DB02) and choose *Detailed analysis* in the *Tables and Indexes* group box.

If you select all the tables (by accepting the default `*`), it may take some minutes for the data to be prepared. If you select *show flagged tables only*, only those tables proposed for reorganization are displayed.

Field	Description
Owner	Owner of the table, that is the user who created the table.
Name	Name of the table
Tablespace	Tablespace to which the table currently belongs.
Overflow Rows	An asterisk (*) in this field means that over 5% of the total number of rows has overflowed.
Freespace Problems	An asterisk (*) in this field means that the net table size is less than 70% of the total table size.
Empty Pages	An asterisk (*) in this field means that the proportion of empty pages is more than 20% of the total number of pages.
Index Reorg Recommend	An asterisk (*) in this field means that the primary index does not fulfill one of 3 check criteria. (See also the option *Table and its indexes* below.) `n/a` means that the table does not have an index.
Table Size	Net table size
Last Update Statistics	The date on which the table statistics were last updated.

Ensure that the table statistics are regularly updated. Out of date statistics can adversely affect system performance. In addition, the values that are displayed could be inaccurate. You can update the statistics using the jobs provided by the DBA Planning Calendar [Seite 1244].

To see more detailed information on a table, select the table you are interested in and choose *Detailed Analysis*.

You have the following options:

- *Table and its indexes*
- *History*
- *Structure*

Table and Its Indexes

Displays a detailed view on the table and its indexes. On the left side of the screen, the parameters for the table and index statistics are displayed. On the right side of the screen, the results of the reorganization check are displayed. If one of the three reorganization checks of the primary indexes is marked by an asterisk (*), then the column *Index Reorg Recommend* on the previous table is selected.

Table Statistics

Field	Description
Tablename	Name of the table
Tableowner	Owner of the table
Table space	Tablespace to which the table belongs
Cardinality	Number of data records in the table
Overflow records	Number of records that have overflowed. Records overflow when a data record is updated and the new data record is larger than the old one or when a column is added to a table.
Number of pages with data	Number of data "pages" containing data.
Total number of pages	Total number of "pages" in the table.
Table size (kB)	Size of the table. The size is calculated from the product of the average length of a data record * the number of records.

Index Statistics

Field	Description
Indexname	Name of the index
Owner	Owner of the index
Table space	Name of the tablespace to which the index belongs
Cardinality	Number of entries in the index
Number of leafs	Number of index leaves
Number of levels	Number of index levels
Distinct 1st key value	Number of different values in the first column of the index
First2keycard	Number of different values in the first two columns of the index
First3keycard	Number of different values in the first three columns of the index
First4keycard	Number of different values in the first four columns of the index
Fullkeycard	Number of different values in all columns of the index
Clusterratio	Degree of fragmentation of the index (100% means no fragmentation. This is the optimum value.)
Clusterfactor	Not currently calculated. The value is set to -1.

Sequential pages	Number of index leaves physically located on the hard disk sorted by index without large intervals between them.
Density	Relative density of the *Sequential pages* as a proportion of the total number of index pages *Number of leafs*. (100% is the optimum value.)
Index size	Net index size

If the value of *Fullkeycard* is the same as that of *Cardinality*, then the index is a "unique index", that is, every record in the table can be accessed via the index.

Field	Description
Overflown rows	An asterisk (*) in this field means that the rows that overflowed amount to more than 5% of the total number of rows.
tablesize / alloc. pages	An asterisk (*) in this field means that the net table size is less than 70% of the total table size.
full pages / alloc. pages	An asterisk (*) in this field means that the proportion of empty pages in the table amounts to more than 20% of the total number of pages in the table.
Clusterratio	An asterisk (*) in this field means that the fragmentation of the index has a value greater than 20% or that the cluster ratio is lower than 80%.
indexsize /alloc.space	An asterisk (*) in this field means that the index size is less than 50% of the total size.
number of entries/no. of possible entries	An asterisk (*) in this field means that the proportion of entries in the index actually occupied is less than 90% of the maximum possible number of entries. The value shows the free entries as a percentage. This value should be lower than 10%.

See Also

Run Statistics (DB2 UDB) [Seite 582]

Run Statistics (DB2 UDB)

To access the run statistics function, call Transaction DB02 and choose *Detailed analysis* in the *Tables and indexes* group box. In the dialog box that is displayed, enter the name of the table or tables you are interested in. A list is displayed corresponding to your selection. Double-click on a table in the list. In the dialog box, select *Table and its indexes*. In the next screen, choose *Run Statistics*.

You can use the function *Run Statistics* to update the table and index statistics. In the case of large tables, this can take some minutes. If applications try to access the table in the meantime, this can lead to wait times in the applications.

History

Shows how the size of the selected table changed in the past. See also Space Statistics: Tables/Indexes [Seite 576]

Structure

The first 3 columns show the table definition as created in the database. Columns 4 to 6 show the definition of the table in the ABAP Dictionary.

If you compare the total size of the tables and indexes with the total size of the database on the screen *Database performance: tables and indexes*, you may notice a difference. The reason for this is that the database size is calculated by adding the size of the tablespace containers whilst the total size of the tables and indexes is calculated on the basis on the average column length of the table and index data. Containers are preallocated in the file system with the result that free space is included in the size calculation for the database.

The difference between the two values can be used as an indicator for the fragmentation of the database.

Finding Storage Management Problems (DB2 UDB)

The DB2 UDB database monitor offers some functions for finding storage management problems in the database.

Using these functions, you can trace tablespaces of the database that need more disk space and find out whether tables or indexes are fragmented and require reorganization.

To access these functions, select *Tools* → *Administration* → *Monitoring* → *Performance* → *Database* → *Tables/Indexes* or call Transaction DB02.

Storage Management

Storage management problems generally develop so slowly that they are detected in time and can be eliminated. However, such problems can also turn up very quickly during the data transfer phase of the R/3 System implementation if you transfer mass data from an old system into a new one.

Tablespace Analysis

A tablespace consists of one or more tablespace containers in which the database objects are stored. A container can be a directory, a file or a raw device. Each container can be put on a different disk. The database manager attempts to balance the load of the data across the containers of one tablespace. All containers will be used to store data. The number of pages written to a container before the next container is used is called the *Extent Size*.

Two different tablespace types can be used with a DB2 UDB database:

- the System Managed Space (SMS) tablespace type, where the operating system's file manager controls the storage space
- the Database Managed Space (DMS) tablespace type, where the database manager controls the storage space.

For performance reasons, only the Database Managed Space (DMS) tablespace type using file containers is supported for R/3.

The DB2 UDB database monitor offers functions for examining the allocated and free storage space of the database.

The state overview screen, *Database performance: tables and indexes,* shows the total number of table spaces, the total allocated space and the freespace in the database. Furthermore, the amount of freespace of the fullest tablespace is shown (*Minimum free/kB*).

The button Detailed Analysis [Seite 577] for the *Tablespaces* lists all table spaces with type, current size, freespace, number of containers and current state.

SAP DB – Monitoring and Analysis

There are a number of database displays you can use to monitor the status of your database system.

You can call these displays from every application server in your R/3 System. All data displayed is identical on each R/3 application server as this information comes from the central database server.

See also:

Diagnosing Performance Problems [Seite 682]

Database Monitor (SAP DB)

You can use the database monitor to display the most important statistics for an analysis of database activity. This provides you with an overview of database performance and supports your monitoring of the database system.

Prerequisite

You are assigned the SAP DB administrator role.

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role: choose *Monitoring* → *Database assistant (*transaction `DB50`). The screen *SAP DB: System Monitoring Analysis* appears.

Then choose *Database: Performance and Configuration* → *Database Performance Analysis* (transaction **`ST04`**).

The statistics are arranged according to the following areas:

Cache Activity [Seite 586]

Commands [Seite 587]

I/O Activity [Seite 588]

Locking Activity [Seite 589]

Logging Activity [Seite 590]

Scan and Sort Activity [Seite 591]

You can also use the Database Alert Monitor [Seite 677] for further analysis.

See also:

DBA Action Logs [Seite 1289]

The Alert Monitor [Seite 812] [Seite 812] (Release 4)

Cache Activity (SAP DB)

Activity for the following R/3 cache types is displayed in the Database Monitor [Seite 585] :

Cache Type	Cache Function
Data	Cache for data pages already accessed
Converter	Cache for administering assignment of logical to physical data pages When data pages not in the data cache are accessed, the system first searches for their physical position in the data dev spaces located in the converter cache.
Rollbacks	Cache for the rollbacks
Catalog	Cache containing the table descriptions needed to parse SQL statements.

Cache Hits

The cache hit rate is:

The **number of times the requested information was found in the cache** divided by the **total number of requested accesses**.

In other words, the number of hits divided by the total number of accesses.

In a balanced system, the hit rata for the data and converter cache should be above 98%, for the catalog cache above 90%.

In general, the lower the hit rate, the less efficient the database.

See also:

Cache Statistics [Seite 596]

Commands (SAP DB)

The following is found in the Database Monitor [Seite 585] and provides you with an overview of SQL statement components.

Table: Commands

Command Type	Command Function
SQL command	Number of statements sent to the database for processing. This is a direct indicator of database activity.
Rollbacks	Indicate **unsuccessful** database actions, usually as a result of a terminated application.
Commits	Indicate **successful** database actions.
Prepares	Number of dynamic SQL statements
Executes	Number of dynamic SQL statements executed

The ratio of rollbacks to commits can indicate logical inconsistencies in one or more SQL statements.

See also:

Performance Database [Seite 641]

I/O Activity (SAP DB)

The I/O activity area in the Database Monitor [Seite 585] provides information about I/O activity initiated by the database. This includes details of both logical and physical read/write activity.

A physical read/write means the system directly accesses the hard disk, while a logical read/write accesses data stored in a data or other cache.

See also:

I/O Statistics [Seite 595]

Locking Activity (SAP DB)

The information about database locking activity found in the Database Monitor [Seite 585], gives the maximum number of available database locks or entries, as well as the number used since database startup. If the number of locks set is close to the number of entries available, you may need to increase the value set for the `MAXLOCKS` database configuration parameter.

The *Escalations* value displays the total number of rows locked by a single user session. If more than a certain percentage of the rows of a table are locked by a single user session, then the database system will lock the entire table. The threshold value for E*scalations* depends on the number of LOCKS available for the database instance.

A lock collision will occur when a request is issued for an object that is already locked. A high number of lock collisions indicates that either a lock is being held too long, or a database record is being accessed frequently by many users.

The *Lockholder* and *Lock request* values are indicators of the lock situation at the time the transaction was executed.

See also:

Exclusive Lockwaits [Seite 604]

Logging Activity (SAP DB)

This area in the Database Monitor [Seite 585], provides details about the number of log pages written by the log writer.

Counts greater than 0 for *Log I/O queue overflows* indicate that the log queue is not configured large enough.

See also:

Logging Statistics [Seite 601]

DBA Action Logs [Seite 1289]

Scan and Sort Activity (SAP DB)

The most important performance indicator in this area of the Database Monitor [Seite 585] is *Table scans*.

Scans should be kept to a minimum, as a high number of sequential table scans can be detrimental to database performance.

Use Bottleneck Analysis [Seite 611] to determine which tables are being scanned frequently.

To reduce the number of table scans, you can:

- Ensure that the required table columns are indexed
- Rewrite the requesting SQL statements so that full table scans are no longer required

Detail Analysis Menu (SAP DB)

You can display information for a database performance analysis and detailed database monitoring using the *Detail analysis menu*.

Prerequisite

You are assigned the SAP DB administrator role.

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role: choose *Monitoring* → *Database assistant (*transaction `DB50`). The screen *SAP DB: System Monitoring Analysis* appears.

Then choose *Database: Performance and Configuration* → *Database Performance Analysis*.

Choose *Detail Analysis Menu*.

The information is separated into the following categories:
- *Database activity analysis*
- *Exceptional conditions analysis*
- *Performance analysis tools*

Choose *Configuration* to display both the current configuration values of the database management system and the current amount of storage space.

Database Activity Analysis

Database Processes [Seite 594]	I/O Statistics [Seite 595]	Cache Statistics [Seite 596]
Region Statistics [Seite 597]	DB Filling Statistics [Seite 598]	Runtime Environment [Seite 599]
Parameter Changes [Seite 600]	Logging Statistics [Seite 601]	Storage Overview [Seite 603]
Database Assistant [Seite 642]	Exclusive Lock Waits [Seite 604]	Tables/Indexes [Seite 605]

Exceptional Conditions Analysis

Control Log File [Seite 606]	Database Message Log [Seite 607]	Overview [Seite 608]

Performance Analysis Tools

Diagnostic Monitor [Seite 609]	Bottleneck Analysis [Seite 611]	Performance Database [Seite 641]

See also:

Database Monitor [Seite 585]

Database Alert Monitor [Seite 677]

The Alert Monitor [Seite 812](Release 4)

DBA Action Logs [Seite 1289] [Seite 812]

DB Processes (SAP DB)

This display in the Detail Analysis Menu [Seite 592] provides you with details about the user kernel processes.

Information about the present status of the various database tasks is displayed . For example, if you frequently find that logwriter tasks have the status *I/O wait*, you may find an I/O bottleneck noted in your log.

A *wait* status means the task is waiting to process new data.

A *Vsuspend* status means that the coordinator process is waiting, for example, for other system resources.

I/O Statistics (SAP DB)

These statistics, found in the Detail Analysis Menu [Seite 592], are used to analyze how the database system devspaces are accessed, both physically and logically.

This also includes details about how the data stored as B*trees is accessed.

I/O Activity [Seite 588]

Cache Statistics (SAP DB)

These statistics are displayed in the Detail Analysis Menu [Seite 592] and provide you with information about the general cache efficiency on the database server.

In a balanced system, the hit rata for the data and converter cache should be above 98%, for the catalog cache above 90%.

In general, the lower the hit rate, the less efficient the database.

See also:

Cache Activity [Seite 586]

Region Statistics (SAP DB)

This area in the Detail Analysis Menu [Seite 592] contains statistics for semaphores and caches that monitor main memory access.

These statistics track cache activity and any resource conflicts. These conflicts occur when multiple requests are sent to the same resource simultaneously. Serious bottlenecks occur when the number of collisions is more than 10 percent of the total number of accesses.

DB Filling Statistics (SAP DB)

These statistics can be displayed using the Detail Analysis Menu [Seite 592] and provide you with information about how full the data areas of the database system are.

To display details about database growth in the last month, choose *History*. You can display these statistics graphically by choosing *Graphics*.

See also:

Checking Free Database Space [Seite 686]

Runtime Environment (SAP DB)

This area of the Detail Analysis Menu [Seite 592] provides details about the processes behind the database operation.

The screen displays the operating system process IDs for the database system and its resources. The upper section covers all processes, while the lower section only shows the processes and tasks responsible for hard disk access.

Parameter Changes (SAP DB)

This area in the Detail Analysis Menu [Seite 592] displays changes made to the database configuration file.

Select period	Select a day from which to start monitoring changes to database parameters
Active parameters	Displays current number of database parameters
History of file	Displays changes to database parameters

Logging Statistics (SAP DB)

This information in the Detail Analysis Menu [Seite 592] provides information about the log status of your database system.

- **Log**

  You can see the:

  - Total size of all log segments
  - Total unarchived used log area
  - Total free log area

  If the fill level is above 60%, it is important to backup the completed log segments to free up space for future transactions. You can automate this process by activating Automatic Log Backup [Seite 1280] (autosave log mechanism).

Log mode	Explanation
Demo	Transaction log is overwritten cyclically after reaching a certain size (not recommended for production systems)
Single	Log segments are backed up
Dual	Log segments are backed up and mirrored (recommended for production systems)

- **Since database restart**

  This section provides details about read/write log accesses. Read accesses are required for transaction rollbacks.

- **Since last backup of log or data (incremental or complete)**

  This contains information about savepoint and checkpoint activity since the last backup. Savepoints and checkpoints are used to ensure database system consistency and to write changed data pages from the data cache to the hard disk.

- **Queue**

  This provides, among other information, details about the *log queue* of the database system. The most important value in this list is the number of *queue overflows*. If the number of queue overflows is greater than zero, increase the LOG_IO_QUEUE database parameter.

See also:

Checking Free Log Space [Seite 687]

Logging Activity [Seite 590]

DBA Action Logs [Seite 1289]

Storage Overview (SAP DB)

This area in the Detail Analysis Menu [Seite 592] contains details about memory assignment for the database kernel, task administrator and various dynamic data pools.

Exclusive Lockwaits (SAP DB)

This area in the Detail Analysis Menu [Seite 592] provides information about all active and requested database locks.

Exclusive locks are used to prevent other users from accessing the locked entries. These locks can considerably reduce R/3 System and database system performance.

To access all the locked table entries, select *Complete lock list*.

- The upper section of this screen (*Lock List Statistics*) provides information about the total number of available locks as well as locks currently set.
- Below this information is more details about the individual locks currently set. The information includes the following:

  Lock Type

LOCK ROW excl	One row in the table is locked
LOCK TAB excl	The entire table is locked

  Timeout

  This displays the time left until the lock is automatically released by the database system.

Request Timeout	The transaction responsible for the lock is terminated and reset.
Lock Timeout	The lockholder has not performed an action using the locked entry within a certain amount of time (the length of time is specified by the database parameter LOCK TIMEOUT).

Procedure to Determine the User that Triggered the Lock

In a database system belonging to an R/3 System the name of the user will always be SAPR3. The *TASKID* can be used to determine which R/3 work process has initiated the lock request.

Using this information and comparing it to the list of Database Processes [Seite 594]), you can determine which user triggered the lock.

See also:

Locking Activit [Seite 589]y

Tables/Indexes (SAP DB)

This area in the Detail Analysis Menu [Seite 592] provides an overview of the database fill level and the tables and indexes contained in the database.

Checking the database fill level

Checking Free Database Space [Seite 686]

Checking consistency of database objects between the R/3 Repository and the database system.

Consistency Checks [Seite 688]

Control Log File (SAP DB)

When you choose this display in the Detail Analysis Menu [Seite 592], the system displays the log for the database manager or CONTROL tool (`control.log`). This log contains the command sequence of the DBA actions executed by the respective tool.

All database error messages and warnings – including system start and stop messages – are sent to this log.

See also:

DBA Action Logs [Seite 1289]

Database Message Log (SAP DB)

These entries in the Detail Analysis Menu [Seite 592] are used to interpret events that have occurred since the database was started.

The system messages are displayed here (kernel log `knldiag`). This log includes the following messages listed in chronological order:

- Database start and stop
- Information about the physical memory areas
- User processes
- System error messages

When the log file reaches a certain size, the system overwrites the beginning of the file with new information. A new log file is created after the database is restarted.

To display the old log (`knldiag.old`) created as a backup copy before the database was restarted, choose *Show Old File.*

To display the error log (`knldiag.err`), choose *Show Error Log*.

See also:

DBA Action Logs [Seite 1289]

Overview (SAP DB)

You can examine the most important system information for the database system by using the overview in the Detail Analysis Menu [Seite 592].

- *System information*
  Status of the database system, hit rate for several important caches
- *CONSOLE*
  Select the desired display and choose *Execute*
- *MONITOR*
  Select the desired display and choose *Execute*

Diagnostic Monitor (SAP DB)

This area in the Detail Analysis Menu [Seite 592] allows you to start the diagnostic monitor

The diagnostic monitor allows you to perform a runtime analysis of long-running SQL commands within an application.

Function	
Diagnostics ON	Switches on diagnostic monitor. Specify the diagnostic options you desire (see “diagnostic options”). Caution: All SQL statements are reparsed as soon as the diagnostic monitor is active.
Diagnostics OFF	Switches off diagnostic monitor Make sure the diagnostic monitor is switched off after the analysis to stop the system from reparsing the SQL statements.
Clear Monitor Tables	Initializes the monitor tables. SQL statements and values determined using the WHERE condition in the SQL statement are entered in the SYSMONITOR and SYSMONDATA tables for the corresponding command. These satisfy the diagnostic monitor options and are transmitted to the database system when the diagnostic monitor is switched on. These tables are overwritten cyclically. The SYSMONITOR and SYSMONDATA tables are not initialized when the diagnostic monitor is stopped. Therefore, use the *Clear Monitor Tables* function instead.
Start Analysis	Analysis of diagnostic monitor data.
Switch Monitor Data On/Off	Activate or deactivate logging of the values for the WHERE condition.

Diagnostic Options

After switching on the diagnostic monitor, you can specify criteria for determining how SQL statements are logged in the diagnostic monitor tables.

Option	Default Value	Explanation
SELECTIVITY	1.0 (%)	An SQL statement is logged in the diagnostic monitor tables when the ratio between qualified records and records read exceeds the percentage specified.
READ	10,000	The SQL statement is logged in the diagnostic monitor tables when the number of virtual reads exceeds the number specified.
TIME	1,000 (milliseconds)	The SQL statement is logged in the diagnostic monitor tables when the statement runtime exceeds the number of seconds specified.

Number	255	This value specifies the maximum number of entries held in the SYSMONITOR table before the table is cyclically overwritten.

Bottleneck Analysis (SAP DB)

For the results of the current bottleneck analysis on the database server, choose *Bottleneck analysis* in the Detail Analysis Menu [Seite 592].

To restart the bottleneck analysis on the database server, choose *Bottleneck analysis* → *Restart x_wizard.*

The bottleneck analysis tool `x_wizard` executes the bottleneck analysis. The tool looks for bottlenecks in the database system at regular 15-minute intervals. These bottlenecks could be low cache hit rates or negative values for table searches. The results of this analysis are displayed in text form to provide the database administrator with a quick overview of possible causes of performance problems.

Calling

If you call the bottleneck analysis using CCMS (*Bottleneck analysis*), all required settings for a routine bottleneck analysis of the database system are set.

If necessary, you can also start the bottleneck analysis on the operating system level. If this is required, contact local support.

Results

If bottlenecks are found, they are assigned an importance as follows:

- I: Information
- W1: Low-grade bottleneck warning
- W2: Medium-grade bottleneck warning
- W3: Severe bottleneck warning
- This classification of warnings is used for applications that have been running for a while (steady system). You can usually ignore warnings that appear just after starting the database system.

All results may not necessarily be caused by true bottlenecks (table searches may make sense in certain cases, long SQL statement runtimes may be necessary for large amounts of data, and so on). In particular for poor search strategies (rows read / rows qualified), you need to analyze the situation further.

Activities

For information about how to fix the bottleneck, see Bottleneck Analysis Messages [Seite 612].

Bottleneck Analysis Messages

Low data cache hit rate

`Low data cache hit rate: <percentage> % <number of> accesses, <number> successful, <number> not successful`

Explanation

The hit rate when accessing the data cache is too low. For a balanced database application, the hit rate should not be less than 99%, as otherwise too much data would have to physically be read. Hit rates may be temporarily low such as when reading a table for the first time, or when a tablescan is repeated and the table does not fit into 10% of the data cache (only for _LRU_FOR_SCAN =YES). The 15-minute average data cache hit rate should not be less than 99%.

User Reaction

In addition to increasing the data cache (watch out for paging problems in the operating system), you should also determine the cause of the high number of reads.

Single SQL statements are often a main cause of the total number of logical and physical reads. Increasing the cache only transfers the load from the hard disk to the CPU, although an additional index can turn a read-intensive table search into a less-intensive direct access.

Low catalog cache hitrate

```
Low catalog cache hitrate: <percentage> % <number of> accesses,
<number> successful, <number> not successful
```

Explanation

The hit rate for accessing the catalog cache, where the parsed SQL statements are administered, is too low. For a balanced database application the catalog cache hit rate should be around 90%. If programs or subprograms are started, the hit rate may temporarily drop to very low rates. However, the 15-minute average should not be below 85%.

User action

The catalog cache should be about 100 pages per database session. This can be checked using XPARAM and the parameters MAXUSERTASKS and CAT_CACHE_SUPPLY. The catalog cache is dynamically increased during the active database sessions, and released by a RELEASE statement. The current cache sizes can be determined using SELECT * FROM CONNECTEDUSERS. If sessions are using considerably more than 100 pages, the catalog cache should be increased for the medium-term, as long as the server's memory can support this.

Low converter cache hit rate

`Low converter cache hit rate_ <percentage> % <number of> accesses, <number> successful, <number> not successful`

Explanation

The hit rate for accessing the converter cache, where the assignment of logical to physical data pages is administered, is too low. For a balanced database application the converter cache hit rate should be at least 98%. If the converter cache is too small additional I/Os may be required as the system must search the converter cache to determine the physical position of data pages not in the data cache.

User action

Increase the size of the converter cache using the XPARAM parameter CONVERTER_CACHE.

Cache swaps

`Cache swaps:<number of> pages/second`

Explanation

Changed pages are being forced out of the data cache and on to the disk as the data the applications require cannot completely fit in the data cache. If the data cache were large enough, the physical write would be delayed until the next savepoint and then executed asynchronously. Cache displacements cause synchronous I/O and should be avoided at all costs. However, long data imports also always cause cache displacement as the imported volume of data is usually larger than the cache.

User action

Increase the size of the data cache and, if necessary, the size of the converter cache.

High read rate (physical)

```
High read rate (physical):<number of>pages per command <number of>
physical reads, <number of> commands
```

Explanation

The application contains statements that require a large number of physical database reads, as the requested data was not found in the data cache. A high read rate is normal if a table is accessed for the first time, or has not been used for a long time and therefore has been removed from the data cache.

User action

If the high read rate is not caused by the first read of the table, check the size of the data cache and the data cache hit rate. Also, determine whether the SQL statements in the application read much more data than is actually required for processing (tablescans or disadvantageous search strategies).

For tablescans, make sure for _LRU_FOR_SCAN=YES (XPARAM) only 10% of the cache is used for buffering, so that there is enough room in the cache for the entire table. Otherwise, the table will have to be physically read again for the next scan.

High read activity (physical)

`High read activity (physical)<number of> pages/second`

Explanation

There are a high number of physical database reads, as the data requested by the applications was not found in the data cache. High read activity is normal if a table is accessed for the first time, or has not been used for a long time and therefore has been removed from the data cache.

User action

If the high read activity is not caused by the first read of the table, check the data cache hit rate and increase the size of the data cache, if necessary. Also determine whether the SQL statements in the application read much more data than is actually required for processing (use the diagnostic monitor to look for unnecessary tablescans or disadvantageous search strategies).

For tablescans, make sure for _LRU_FOR_SCAN=YES (XPARAM) only 10% of the cache is used for buffering, so that there is enough room in the cache for the entire table. Otherwise, the table will have to be physically read again for the next scan.

High write activity (physical)

```
High write activity (physical):<number of> pages/second
```

Explanation

There are a high number of physical writes to the data devspaces as the data requested by the applications cannot be fully stored in the data cache. In addition, data is being forced out of the cache and on to the hard disk. During large data imports, data is almost always forced out of the cache, as the imported data volume is usually much larger than the cache.

There is a savepoint at regular intervals (default: every 10 minutes). This involves a flushing of the data cache with all changed pages being written to the disk to ensure a consistent database state in the devspaces. During this process, the I/O activity increases dramatically (disk load close to 100%), without this being a true bottleneck. During normal operation, there shouldn't be high amounts of write activity besides the savepoints.

User action

If high amounts of write activity are noticed during normal operation, first check to determine whether this is due to a savepoint that may have occurred during the check interval (which may also have been too short). Otherwise, only by increasing the size of the data cache can the cache displacement be avoided.

High read rate (virtual)

```
High read rate (virtual):<number of> pages per command, <number of>
virtual reads, <number of> commands
```

Explanation

The application contains statements that require a large number of logical database cache reads. The only way to determine if this is a problem is to examine the application profile. For example, an application containing a large number of mass selects with relatively unspecific WHERE clauses will lead to a high number of virtual reads.

User action

Check to determine whether the SQL statements executed in the application are actually reading much more data than is actually required for processing.

High parse activity

```
High parse activity:<number of> prepares per command, <number of>
commands (executes), <number of> prepares
```

Explanation

Compared to the total number of statements executed, the number of parses is too high. Before an SQL statement is executed for the first time, the database system analyzes (parses) the SQL command string, whereby it determines the possible access strategies and stores the statement in a compact form in the database, among other actions. After this, the database only needs to access this internal information and then directly execute the statement.

If the statement was created using static SQL and a precompiler, the precompiler ensures that the parse only occurs once for each statement. If dynamic SQL or the CALL interface was used, the developer is responsible for the administration of the parse and execute requests. High amounts of parse activity in production operation may mean the cursor cache was implemented incorrectly. High parse activity is normal the first time a program or subprogram is started.

User action

This cannot be influenced by any specific change to the database.

Low hit rate for table scans

Low hit rate for table scans: <percentage>%, <number of> scans, >number of> rows read, >number of> rows qualified

Explanation

During tablescans, there is a poor ratio of read lines to found (qualified) lines. In almost all cases, this means the search strategy is poor. This can be caused by the application (missing or insufficient indexes, and so on) or by a problem during the cost-based SELECT optimization in the database kernel. Scanning large tables can considerably reduce the performance of the entire system due to a number of negative affects (I/O, overwriting the data cache, CPU load, and so on).

User action

First, rebuild the internal database statistics to determine if the database optimizer can find a better search strategy and therefore avoid tablescans. You can use CONTROL to update statistics, or use the UPDCOL tool from an operating system command line. As large amounts of data may need to be checked, these statements may run for a long time and should therefore not be started during active applications, if possible, which will also help to avoid any lock conflicts. (stopsap r3 - Update Statistics - startsap r3).

If this is not successful, check the statement that triggered the tablescan. You can do this by starting the diagnostic monitor or an R/3 SQL trace, and then checking for any long-running statements where the EXPLAIN statement is used to check the search strategy used by the optimizer.

Low hit rate for optimizer strategy

```
Low hit rate for optimizer strategy:<percentage>%, <number of>accesses,
<number of>rows read, <number of> rows qualified
```

Explanation

There is a poor ratio of read lines to found (qualified) lines for the access strategy used by the database optimizer. See also Low hit rate for table scans [Seite 622].

User action

First rebuild the internal database statistics to see if the database optimizer can find a better search strategy. You can use CONTROL to update statistics, or use the UPDCOL tool from an operating system command line. As large amounts of data may need to be checked, these statements may run for a long time and should therefore not be started during active applications, if possible. This will also help to avoid any lock conflicts. (stopsap r3 - Update Statistics - startsap r3).

If this is not successful, check the statement that triggered the poor search strategy. You can do this by starting the diagnostic monitor or an R/3 SQL trace, and then checking for any long-running statements where the EXPLAIN statement is used to check the search strategy used by the optimizer.

Low hit rate on <deletes/updates>

```
Low hit rate on <deletes/updates>: <percentage>%, <number of> rows
read, <number of> rows qualified
```

Explanation

There is a poor ratio of read lines to changed lines for DELETES and UPDATES. Before UPDATES and DELETES can be used to change or delete lines, the lines must be localized in the corresponding table. For this, the system uses the same access strategies used for SELECT.

User action

First rebuild the internal database statistics to see if the database optimizer can find a better search strategy. You can use CONTROL to update statistics, or use the UPDCOL tool from an operating system command line. As large amounts of data may need to be checked, these statements may run for a long time and should therefore not be started during active applications, if possible. This will also help to avoid any lock conflicts. (stopsap r3 - Update Statistics - startsap r3).

If this is not successful, check the statement that triggered the poor hit rate. You can do this by starting the diagnostic monitor or an R/3 SQL trace, and then checking for any long-running UPDATE or DELETE statements.

'Physical Temp Page Writes' high

```
’Physical Temp Page Writes’ high: <number of Pages> per command
creating big result tables
```

Explanation

While creating temporary database pages for building temporary results (for example, for joins, ORDER BY statements, and so on), the data cache was not large enough to accept all the temporary pages, displacing data on to the disk. Physical writes of temporary pages should be avoided, as these pages must be read back in for further processing of the SQL statement,. This problem is often caused by problems with application design (for example, missing indexes) or the database optimizer. Collection of large result quantities can considerably reduce performance of the entire system due to a number of negative affects (I/O, overwriting of the data cache, CPU load, and so on).

User action

First rebuild the internal database statistics to see if the database optimizer can find a better search strategy. You can use CONTROL to update statistics, or use the UPDCOL tool from an operating system command line. As large amounts of data may need to be checked, these statements may run for a long time and should therefore not be started during active applications, if possible. This will also help to avoid any lock conflicts. (stopsap r3 - Update Statistics - startsap r3).

If this is not successful, check the statement that triggered the result creation. You can do this by starting the diagnostic monitor or an R/3 SQL trace, and then checking for any long-running statements where the EXPLAIN statement is used to check the search strategy used by the optimizer.

High collision rate on SQL locks

```
High collision rate on SQL locks: <average number> per write
transaction, <number of> write transactions, <number of> SQL collisions
```

Explanation

A large percentage of the write transactions are resulting in locks of SQL objects (lines and tables), which in turn are causing waits in the application, until the locking application tasks releases the lock by sending COMMIT. This problem is usually a problem with application design and not with the database. The database system attempts to execute locking tasks in the database kernel with higher priority if these locks are requested by other sessions (which are then in Vwaits) to avoid queues for SQL lock objects.

User action

Determine whether the application is suited for isolation level 0 (dirty read) to avoid read locks ((SAP R/3 runs on isolation level 0). Also, determine whether the time between the lock being set and the COMMIT can be reduced (avoid locks during dialog sessions).

Log writing may be the cause of another bottleneck, as the SQL locks for the transaction can only be released after the successful, physical log I/O of the COMMIT. Therefore, it makes sense to place the log in the fastest devspaces possible. By monitoring the maximum length of the log queue, you can determine whether bottlenecks occur when the system writes to the log.

Long waiting time with SQL collisions

```
Long waiting time with SQL collisions: <duration> seconds per Vwait
(<number of> vwaits)
```

Explanation

The wait time for the release of a SQL lock is very long for collisions of SQL objects. SQL locks are released by the locking application with COMMIT. Transactions that run too long, where the application sets a long SQL lock, are usually the cause of long wait times. Long wait times may also occur if many applications request locks on the same object, as there is a queue for the relevant SQL lock. Due to sequential processing, the queue is often reduced slowly (particularly for multi-CPU systems). The database system attempts to execute locking tasks in the database kernel with higher priority if these locks are requested by other sessions (which are then in Vwaits) to avoid queues for SQL lock objects. For information about the current lock situation in production operation, use SELECT * FROM TRANSACTIONS.

User action

Determine whether the application is suited for isolation level 0 (dirty read) to avoid read locks (SAP R/3 runs on isolation level 0). Also, determine whether the time between the lock being set and the COMMIT can be reduced (avoid locks during dialog sessions). If queues for SQL objects are occurring, check the application to determine whether tables might be split to avoid simultaneous locks on the same line.

Queue on SQL collisions

```
Queue on SQL collisions: coll/vwait = <relation>, <number of> lock list
collisions, <number of> vwaits
```

Explanation

Queues for SQL locks are occurring, that is, more than one task is waiting for these locks to be released. The locking application releases the SQL lock with COMMIT. Transactions that run too long, where the application sets a long SQL lock, are usually the cause of long wait times. Long wait times may also occur if many applications request locks on the same object, as there is a queue for the relevant SQL lock. Due to sequential processing, the queue is often reduced slowly (particularly for multi-CPU systems). The database system attempts to execute locking tasks in the database kernel with higher priority if these locks are requested by other sessions (which are then in Vwaits) to avoid queues for SQL lock objects. For information about the current lock situation in production operation, use SELECT * FROM TRANSACTIONS.

User action

Determine whether the application is suited for isolation level 0 (dirty read) to avoid read locks (SAP R/3 runs on isolation level 0). Also, determine whether the time between the lock being set and the COMMIT can be reduced (avoid locks during dialog sessions). Modify the application logic to avoid simultaneous locks on the same line.

Lock escalations (<number of> table locks)

`Lock escalations (<number of> table locks)`

Explanation

The number of SQL line locks on a table set by a transaction exceeds the threshold. Therefore the individual line locks are converted into a table lock. SQL locks are usually set on individual lines in a table. However, administration of a high number of individual line locks is expensive and only a limited number of locks can be administered in the database lock list. Therefore, a configurable threshold is set after which the system attempts to lock the entire table for the transaction. This means that until the COMMIT is set, other transactions cannot set any locks on lines in this table.

User action

You can set the maximum number of individual line locks that a database can administer using the XPARAM parameter MAXLOCKS. Escalation is attempted when a task holds more than 0.1*MAXLOCKS line locks in a table. If undesired lock escalations are occurring often, you should increase the value of the parameter (maximum 2.3 million). Whether lock escalations become a problem depends largely on the application. If lock escalations are occurring, check the application to determine whether changing transactions, which lock a large number of lines, can be staggered using COMMITS.

Log queue overflows

```
Log queue overflows (<number>), parameter ’LOG_IO_QUEUE’ (<number of
pages>) too small
```

Explanation

The queue used for accepting log entries overflowed. When change transactions write log entries, the log entries are first stored temporarily in a queue, before the logwriter writes them to the log devspace. This queue usually consists of a single page, but for mass statements (mass DELETES, array INSERTS, and so on), there may be more log entries to write than the system can simultaneously physically write to the disk. If the log queue overflows, it cannot accept any more log entries. This results in a large number of internal database wait situations (Vsuspend) within a short amount of time. Because transactions that write log entries hold SQL locks, other transactions will be affected.

User action

Increase the XPARAM parameter LOG_IO_QUEUE (maximum: 2000). Also, check whether the log devspaces can be moved to a faster disk, to accelerate the physical log I/O.

'Log I/O-Queue' too small

```
'Log I/O-Queue' too small: total <pages>, max. used <pages>
```

Explanation

The queue for log entries may be too small. When change transactions write log entries, the log entries are first stored temporarily in a queue, before the logwriter writes them to the log devspace. This queue usually consists of a single page, but for mass statements (mass DELETES, array INSERTS, and so on), there may be more log entries to write than the system can simultaneously physically write to the disk. If the log queue overflows, it cannot accept any more log entries. This results in a large number of internal database wait situations (Vsuspend) within a short amount of time. Because transactions that write log entries hold SQL locks, other transactions will be affected.

User action

Although the log queue has not overflowed, you should increase the XPARAM parameter LOG_IO_QUEUE (maximum: 2000) for the medium-term. Also, check whether the log devspaces can be moved to a faster disk, to accelerate the physical log I/O.

High log activity

`High log activity <number of>pages/second`

Explanation

The number of log pages written per time unit is very high. Depending on the capability of the current log disks, the physical writing of the logs may create a bottleneck. For each COMMIT, the entire 4KB log page must be written to the disk even if the page is not full. This means that if there are a lot of short transactions a log page may have to be physically written to the disk multiple times. In a multi-user system, the database system attempts to combine the COMMITS from several application tasks into group commits.

User action

If the measured I/O rate is close to the limit of the log disks, you should consider moving the logs to faster disks.

Long write transactions

```
Long write transactions: <number of> log pages per transaction <number
of> write transactions, <number of> log pages
```

Explanation

The write transactions from the application are very long causing a large number of physical writes to the log. This is not a problem for background applications. However, long write transactions can cause bottlenecks if other sessions are attempting to access SQL objects (lines and tables) locked by the long write transaction. In some cases, very long transactions may cause a delay when a backup requests a checkpoint, because at the time of the checkpoint the system must wait for a COMMIT from every open write transaction. No new write transactions are permitted until the checkpoint is finished, which almost always forces a temporary stop in database operation (all tasks with status Vwait).

User action

This bottleneck cannot be influenced using the database. You can avoid the wait times due to checkpoint requests by running data backups without checkpoints.

High collision rate on <name> region

```
High collision rate on <name> region: <percentage>%, <number of>
accesses (of a total of <number>), <number of> collisions
```

Explanation

The collision rate is very high when accessing protected areas of the database kernel memory. Access to critical areas in the database kernel memory used by multiple tasks is protected using regions. A database task is assigned exclusive access to a region to prevent multiple database processes / threads from simultaneously manipulating a global memory area. If only one processor is used (XPARAM parameter MAXCPU=1), there can almost never be collisions on this region due to internal tasking (exception: parallel data backup, CONNECTS, and SAVEPOINT). If multiple CPUs are used and there are a high number of collisions on this region, the entire database operation may be sequentialized for accessing this region. In this case, the use of additional CPUs can actually reduce performance due to this additional synchronization administration requirement.

User action

You need to take action if the collision rate is higher than 10%. The higher the number of processes (XPARAM parameter MAXCPU) the more likely there will be collisions. You should therefore check multiprocessor systems to determine whether the database might still be able to adequately handle the application requirements with fewer CPUs. In the R/3 environment the ratio of CPU usage between the database and R/3 is approximately 1:4; this means that on an eight-CPU machine, two CPUs will do. These calculations can also be made for client/server architectures.

If there are high region collision rates on centralized multiple-processor systems (database and application on the same server), check whether the machine is CPU-bound, where the database processes are blocked by the application. If this is the case, those database processes that contain user tasks should receive REAL TIME PRIORITY from the operating system. Make sure, however, that MAXCPU is at least one less than the actual number of CPUs.

Additional measures:

- DATAn, SPLITn, TREEn regions:

  You can use the XPARAM parameters _DATA_CACHE_RGNS and _TREE_RGNS to increase the segmentation of the data cache, thus reducing the likelihood of collisions. At the same time, you should slightly increase the data cache (DATA_CACHE) so that the subcaches are not too small. Problems can arise if there is a high collision rate on only one of the subregions of the DATA, SPLIT or TREE structures. This means that multiple applications are obviously simultaneously processing the same page or at least frequently processing the same table (rootpage). Only a change to the application logic can improve the situation in this case.

- TRACE, BUFWRTR regions:

  Vtrace should be activated only temporarily to find a database problem.

High TAS collision rate

```
High TAS collision rate: <number> per region accesses, <number of> TAS
collisions, <number of> region accesses
```

Explanation

There is a very high collision rate when accessing internal database semaphores during region access. See also High collision rate on <name> region [Seite 634]. If the parameters are set correctly, this problem is only observed on multi-CPU machines or with a high UKP number (XPARAM parameter MAXCPU > 4).

User action

The likelihood of TAS collision increases as the number of the UKPs increases (XPARAM parameter MAXCPU). You should therefore check multiprocessor systems to determine whether the database might still be able to adequately handle the application requirements with fewer CPUs. In the R/3 environment the ratio of CPU usage between the database and R/3 is approximately 1:4; this means that on an eight-CPU machine, two CPUs will be sufficient for SAP DB. These calculations can also be made for client/server architectures.

If the database is running on a pure database server (client/server), the number of UKPs should be at least one less than the number of CPUs on systems with four or more CPUs. If TAS collisions still occur, set the XPARAM parameter _MP_RGN_DIRTY_READ to YES.

If a large number of TAS collisions occur in centralized multiprocessor systems with less than four UKPs (XPARAM parameter MAXCPU), check whether the machine is CPU-bound with the database processes blocked by the application. If this is the case, those database processes that contain user tasks should receive REAL TIME PRIORITY from the operating system. Make sure, however, that MAXCPU is at least one less than the actual number of CPUs.

Large number of time-consuming commands

```
Large number of time-consuming commands (>1 second): <percentage>%,
<number of> long commands, <number of> commands
```

Explanation

A high percentage of SQL statements have runtimes longer than one second in the database kernel. Whether this is an actual bottleneck depends on the structure of the application. For example, mass statements in background processing often lead to long runtimes. Wait times may also be caused by locks on SQL objects, thus increasing processing time. The fact that long-running statements are occurring is therefore only a warning.

User action

If there is no additional information from x_wizard, check whether the database server is CPU-bound.

Long command runtime in DB kernel

`Long command runtime in DB kernel (receive/reply): <duration> seconds`

Explanation

The average time the database kernel processes SQL statements is longer than 100 milliseconds. Whether this is an actual bottleneck depends on the structure of the application. For example, mass statements in background processing often lead to long runtimes. Kernel wait times may also be caused by locks on SQL objects, physical I/O, dispatching due to prioritization of other tasks, and so on, thus increasing processing time.

User action

If there is no additional information from x_wizard, check whether the database server is CPU-bound.

Large number of self suspends (dispatches)

`Large number of self suspends (dispatches): <number> per command`

Explanation

There are a high number of task displacements in the database. The processing of long-running applications is terminated after a set time (XPARAM parameter _MAXRGN_REQUEST), to prevent other transactions from being blocked in the database (similar to timeslices in operating systems). To determine whether this is actually a problem, check the application profile. For example, complex searches in the data cache almost always dramatically increase the number of self-suspends. A high number of self-suspends is definitely indicative of a high number of long-running statements.

A task can execute a self-suspend when another task with higher priority changes from status waiting to status ready (only for XPARAM _MP_DISP_PRIO=YES).

User action

For background applications, the number of self-suspends can be reduced by increasing the XPARAM parameter _MAXRGN_REQUEST. This can improve throughput because the system will now sequentially process the statements, but creates a disadvantage for short-running statements due to higher dialog response times.

If your analysis of the database application does not reveal any indication of complex statements, check whether the SQL statements are requiring much more data to be read than is actually required for processing (for example, due to tablescans or poor search strategies).

Long vsuspend time (user tasks)

```
Long vsuspend time (user tasks): <duration>seconds per vsuspend
(<number of> vsuspends)
```

Explanation

Wait times in the database are too long. This does not refer to collisions on SQL lock objects, which lead to Vwait, but rather to waiting states while waiting for various events, such as the writing of a log entry, the release of a B* tree after a structure change, and so on.

User action

Contact local support, as only they are able to analyze the situation in more detail.

Large number of vsleeps (user tasks)

```
Large number of vsleeps (user tasks): <number> per command, <number of>
vsleeps, <number of> commands
```

Explanation

The internal database wait state Vsleep is occurring too frequently.

User action

Contact local support, as only they are able to analyze the situation in more detail.

Performance Database (SAP DB)

Daily performance statistics are created to compare database activity with that from previous days. You can display these statistics using *Performance database* in the Detail Analysis Menu [Seite 592].

The information includes the number of SQL statements, commit, rollback and prepare operations.

See also:

Commands [Seite 587]

Database Assistant (SAP DB)

Use

The database assistant (available as of R/3 Release 4.6A) is a new tool for monitoring the database system.

Prerequisite

You are assigned the SAP DB administrator role.

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role: choose *Monitoring* → *Database assistant (*transaction `DB50`). The screen *SAP DB: System Monitoring Analysis* appears.

The following views and functions are available:

Current System Status (SAP DB) [Seite 643]

Problem Analysis (SAP DB) [Seite 658]

Configuration (SAP DB) [Seite 670]

System Checks (SAP DB) [Seite 675]

Statistics (SAP DB) [Seite 679]

Current System Status (SAP DB)

Processes (SAP DB)

This provides you with an overview of the active activities for the individual tasks. You can define your own task view.
Use this process overview to analyze the type of activities in your database system and to recognize any bottlenecks by examining the wait and lock situation data.
Also use the display of current **B* tree locks**. This allows you to recognize any locks on individual data structures.

Task Manager (SAP DB) [Seite 645]

DB Threads (SAP DB) [Seite 647]

Task Manager (SAP DB)

Use

Display all database tasks and their current statuses.

Terminate running SQL commands and tasks.

Prerequisites

You are assigned the SAP DB administrator role in the CCMS (SAP_BC_DB_ADMIN_SAPDB).

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role, choose *Database: Monitoring* → *Database assistant* (transaction `DB50`).

Choose SAP DB *Assistant* → *Processes* → *Task manager*.

Result

An overview of the database tasks appears as well as information about the current status of each task.

You can choose between the following views:

Active tasks

User tasks (task type User)

System tasks

All tasks

The following data (among other information) appears:

`Task ID`	Internal task ID
`Thread/Process ID`	Thread ID of corresponding internal task.
`Application PID`	Process ID of application program assigned to task. If there is an asterisk (*) in front of the PID, the PID is on a different server.
Task Status [Seite 708]	Information about the current statuses of the database tasks.

If logwriter tasks often have the status `I/O wait`, this may be caused by an I/O bottleneck in your log.

You can execute the following functions:

To terminate the action of an active task, select the task.

By choosing *Edit* → *SQL command*, you terminate the command just executed.

If you choose *Edit* → *Terminate SQL session* instead, you terminate the task itself. If an R/3 System work process was assigned to this task, the work process will immediately be assigned to a new task.

Use one of these two possibilities for ending an active action, if you want to add the relevant SQL command to the performance trace after the command is terminated. If you terminate an action by terminating the relevant work process (transaction `SM50`), the command is not added to the trace.

If you observe that certain tasks have the status `Vsuspend` for a long time, choose *B* tree locks*, to check whether the tasks are waiting for B* tree locks.

If you observe that certain tasks have the status `Vwait` for a long time, choose *Current system status* → *SQL lock* → *Waits* for information about the current waits.

To refresh the display, choose *Edit* → *Refresh.*

DB Threads (SAP DB)

This section contains detailed information about the basic database processes. This includes the operation system thread and process IDs of the database and resources.

I/O Operations (SAP DB)

This provides you with an overview of the activities on the disks where the data devspaces, system devspace, and the log devspaces are stored. Use the read and write access information to determine how many I/O operations are being executed in your database system.

When the kernel trace is activated (*SAP DB: System Monitoring and Analysis* screen: Choose *Edit* → *DB kernel trace* and then *DB kernel trace* → *Activate*), you can determine whether entries are being written to the `knltrace` trace file.

You can run a **detailed analysis** of the processes that are triggering the I/O operations on the devspaces. See: Overview (SAP DB) [Seite 649]

If there are *Queues* for accessing the disk area, this indicates a disk access bottleneck.

You can monitor **asynchronous I/O operations** triggered by backups. You can refresh the display while a backup is running. This gives an indication of the progress of the backup.

See: Backup (SAP DB) [Seite 650]

Overview (SAP DB)

Displays the number of reads and writes in the devspaces and in the `knltrace` file (where the kernel trace is written to) since the last database restart. One access equals a read or write of one page.

To display which thread or process accessed the individual devspaces and for information about the I/O queue of each devspace, choose **Display I/O details**. This information allows you to recognize I/O bottlenecks. The *Queue length* and *Maximum queue length* columns indicate whether there is a large amount of I/O requests queued currently or whether there were a large number of queued requests in the past. To update the display, choose **Refresh**.

Backup (SAP DB)

Displays the number of devspace reads and writes due to the current backup action. One access equals one I/O operation, which can include a varying number of pages. The actual number of pages backed up is displayed under *Backup.*

Synchronization Objects (SAP DB)

This provides you with an overview of access to the exclusive main memory structures (synchronization objects or regions), including information about which process (task) accessed the main memory structure.

These statistics are useful for analyzing the cache activity or any resource conflicts. Conflicts occur when several requests are sent to one resource simultaneously. If the number of conflicts is more than 10% of the total accesses, this indicates severe bottlenecks in your system.

SQL Locks (SAP DB)

This displays all active and requested database locks. Exclusive locks mean that other users cannot access the locked entry, but these locks can severely reduce performance of your R/3 System and database system.

Procedure for Determining Which User Triggered the Lock:

In a database system that is part of an R/3 System, this user is always SAPR3. Using the *TASKID*, you can determine the corresponding *Application PID* and the application server through **Current system status / Task manager**. Then use transaction `SM50` to determine which R/3 work process triggered the lock or lock request.

Waits (SAP DB) [Seite 653]

Total (SAP DB) [Seite 654]

Waits (SAP DB)

List of all current database waits for SQL locks.

Total (SAP DB)

This shows detailed information about the locks currently set. In a running R/3 System, this list can be very long. Therefore, you should always start the analysis of SQL locks through Waits [Seite 653].

Space Areas (SAP DB)

Overview of the use of database space.

Data Area

This provides information about the current database fill level.

This area contains the following information:

- Total data area size
- Size of the data area used by permanent and temporary data pages
- Total free data area
- Number of data pages changed since the last data backup. This provides you with the number of data pages an incremental backup would include at this point.

Log Area

This provides information about the status of the database log and the number of savepoints and checkpoints written since the database was installed.

This area contains the following information:

- Log mode
- Total log area size
- Log segment size
- Reserved area for REDO operations in the data area
- Size of the log area used that is not backed up
- Total free log area

If more than 60% is used, backup the log segments to free up space for additional transactions. To automate this process, activate automatic log backup.

Log mode	Explanation
`Demo`	When log reaches a certain size, log will be overwritten (not recommended for production systems).
`Single`	Log segments are backed up
`Dual`	Log segments are backed up and mirrored (recommended for production systems)

The reserved data area is used during a database restore for loading log entries. Besides this use, the area can only be used by temporary data pages.

Caches

Displays size of most important caches (data, converter, catalog, and rollback caches) as well as successful accesses and the cache hit rate for all caches. This provides you with an overview of the general cache efficiency on your database server. In a balanced system, the hit rate for the data and converter caches should be above 98%; for the catalog cache, above 90%. The lower the hit rate the less efficient the database system.

System Settings (SAP DB)

This displays the current version of the system and the current options set.

This indicates, for example, whether join optimization, monitoring, or a trace is activated, as well as whether a checkpoint was requested or there is an error in an index.

Problemenanalyse (SAP DB)

Hier finden Sie die wichtigsten Werkzeuge zur Analyse von Datenbankproblemen.

Neben der Ausgabe der wichtigsten Protokolldateien und detaillierten Informationen zu einzelnen Tabellen können Sie hier den Datenbanktrace einschalten und Performanceengpässen auf die Spur kommen.

Für eine allgemeine Analyse der Performance wählen Sie den Einsteig über die Engpassanalyse. Als Ergänzung hierzu erstellen Sie mit der Analyse des SQL-Ressourcenverbrauchs eine Übersicht der SQL-Kommandos, die im Beobachtungszeitraum (ca. ein Tag) die meisten Ressourcen verbraucht haben.

Werden von den Anwendern spezielle langlaufende Transaktionen gemeldet, führen Sie eine Kurzzeitanalyse mit der Kommandoanalyse durch.

Bottlenecks (SAP DB)

Calling the database bottleneck analysis tool x_wizard

Use the x_wizard program to start every database performance analysis. If your system seems to have long database response times (average response time is longer than one second), call x_wizard and run it over a long time period (such as a day). The program provides a database performance analysis at 15-minute intervals. Bottlenecks found are indicated in color; general information has a gray background.

Start the program at the operating system level using **`Restart x_wizard`** and display the results using **`Refresh`**.

Analyzing the results

In the *Detail Menu* you can display a detailed log of the performance-related database information, arranged according to sub-areas. To be able to analyze the results, you need a thorough understanding of SAP DB.

SQL Performance (SAP DB)

Command Analysis (SAP DB)

You use this tool when the bottleneck analysis indicates there are inefficient database accesses. This allows you to identify long-running SQL commands. This tool is designed for a short-term analysis, as only a limited number of commands can be recorded. You enter recording criteria to restrict the number and type of recorded commands.

Activate monitoring just before a transaction to be analyzed is started and deactivate recording immediately after the transaction is completed. To activate and deactivate monitoring and to change the recording criteria, choose *Change monitor settings*. SAP delivers default settings, which you can later change, if needed. To use the default settings, choose *Apply SAP default settings*.

To display the logged SQL statements, choose *Update*.

If a logged SQL statement was executed from an ABAP program, you can analyze the statement down to the source code. Select the statement and choose *Calling point in ABAP program*.

Double-click to go a detailed view of the command including the complete command. If during the recording of the command the recording option *Save parameter values* was active, you can determine which access strategy the SQL optimizer would use to process the command by choosing *Display processing plan for SQL statement*. Choose *Trace processing plan for SQL statement* to create an optimizer trace for the support team.

Resource Use (SAP DB)

Analyzing the use of resources allows you to identify the most expensive SQL statements. This measures the resources used by the SQL statement (for example, runtime, I/O accesses, and so on). If an SQL statement is executed several times, the total costs are determined. This allows you to identify commands that may have a short runtime, but create a high load on the database as they are executed multiple times. This tool is a monitoring tool you can use to analyze load for a specific time period, such as a day.

Messages (SAP DB)

This displays the system messages (kernel log `knldiag`) and is used to interpret events that have occurred since the database system was started. This logs includes a chronological list of the following messages:

- Database start and stop
- Information about the physical space
- User processes
- System error messages

This log is a cyclical file, that is, it is overwritten once it has reached a specific size. The first part of the file, which logs the last restart of the database, is not overwritten. A new log is created every time the database is restarted. To update the list, choose *Import message file*. Use the paging icons to go to the beginning of the file, end of the file, beginning of the write cycle, or the current write position in the file.

Current Messages (SAP DB)

To display the current log (kernel log `knldiag`), choose *Current*.

Old Messages (SAP DB)

The log file is backed up before the database is restarted. To display this old log (`knldiag.err`), choose *Old.*

Errors (SAP DB)

As of SAP DB Version 6.2, all errors and warnings from the database kernel are logged in an error log. This file is written sequentially. To display the error log (`knldiag.err`), choose *Error*.

DBA Logs (SAP DB)

Backups

Overview of all backup and restore actions since the last database installation. To update the display, choose *Import backup logs*. For a detailed log of an action, double-click the relevant line in the table.

Consistency Checks

Overview of all consistency checks. To update the overview, choose *Update display*. For a detailed log of an action, double-click the relevant line in the table.

Tables (SAP DB)

Information about database tables.

Enter the name and owner of the table and choose *Display table information.*

You can also enter an asterisk (*) with a character string to display a list of table names, such as by entering “T0*” for all table names beginning with “TO”. Choose a table name from the list and then *Display table information.*

You can choose from the following displays:

- Attributes: Type, access rights, creation and change date as well as the date of the last run for determining optimizer statistics for this table.
- Table definition stored in the database (this is not the table definition stored in the ABAP Dictionary, but rather the table definition from the system tables in the database).
- Indexes defined for this table.
- The values last determined for the optimizer statistics.

Kernel Trace (SAP DB)

The kernel trace is an extended trace of the database kernel. As the writing of this trace is very performance-intensive, only activate it for problem analysis.

If you want to ensure that only the succeeding database actions are logged, choose *Initialize trace*.

To activate the trace, first choose the trace options you want to use (usually these are the default values in the system) and then choose *Activate trace*.

While the trace is running you can add additional trace options by selecting them and then choosing *Activate trace* again.

Current trace status indicates which trace options were selected for the current trace running.

To deactivate the trace, choose *Deactivate trace*. This also deactivates the selected trace options.

To specify that the trace be automatically deactivated if a certain error occurs, choose *Deactivate trace after error* and enter the database error number.

To analyze the trace, first choose *Flush trace bu*ffer, then *Prepare and display trace*. Save the displayed trace by choosing *Save to local file*, so that the trace display is not overwritten during the next trace analysis.

To display a previous trace, choose *Display trace.*

Configuration (SAP DB)

Parameters (SAP DB)

This displays changes to database parameters.

Current (SAP DB) [Seite 672]

Change History (SAP DB) [Seite 673]

Current (SAP DB)

This displays the current database parameters.

You can choose from the following display variants:

- General parameters
- Extended parameters
- Support parameters
- All database parameters

To display previous changes to a database parameter, click the symbol at the beginning of the line containing the relevant parameter, or select the parameter name and choose *Expand subtree*. A list of parameter changes and values appears.

The parameter history data is logged once a day by the collector. If you have changed database parameters, the correct change date is displayed only after the collector has logged the change.

Change History (SAP DB)

List of all database parameter changes sorted by change date.

To display changes for a specific date, click the symbol at the beginning of the line containing the relevant date, or select the date and choose *Expand subtree*. A list of parameter changes for that day appears as well as the previous and new parameter values. Parameters no longer used by the database after a specific date have the value <<parameter inactive>>.

The parameter history data is logged once a day by the collector. If you have changed database parameters, the correct change date is displayed only after the collector has logged the change.

Devspaces (SAP DB)

Overview of database devspace configuration (log, data, and system devspaces).

System Checks (SAP DB)

Database Operations Monitor (SAP DB)

Use

Use the database operations monitor for online monitoring of database operations of internal and external database tools. (See: Using the Database Operations Monitor [Seite 1235]).

Database Alert Monitor (SAP DB)

For information about the alert monitor (Release 4), see The Alert Monitor [Seite 812] (Release 4).

Prerequisite

You are assigned the SAP DB administrator role.

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role: choose *Monitoring* → *Database assistant (*transaction `DB50`).

Choose *Utilities* → *Database Alert Monitor*.

Result

The database alert monitor information is displayed under *SAP DB*.

- **Green** means that the database is OK. The database administrator does not need to intervene.
- **Yellow** or **red** means that this state puts the production operation of the database at risk. The database administrator must determine the cause of the alert and fix the problem.

	Explanation
SpaceManagement	Displays the free memory space in your database system. Checking Database Free Space [Seite 686] *Data Area*: Displays the amount of free data space (%). The delivered preset threshold is set to generate a yellow alert when the free space is less than 30%. A red alert is generated when free space is less than 10%. To change these settings, use transaction `RZ21`. *Log Area*: Displays the amount of free log space (%).The delivered preset threshold is set to generate a yellow alert when the free space is less than 33%. A red alert is generated when free space is less than 5%. If a yellow or red alert is displayed, back up the log area to ensure enough space is available in the log area for new log entries. If no log entries can be written, the database crashes.

Performance	Displays performance-relevant measurements. If yellow or red alerts are displayed, performance will be worse. *Optimizer Statistics*: Indicates whether optimizer statistics need to be updated. This checks both the last optimizer check and the current update statistics. If the last check is older than 30 days, a yellow alert appears; if older than 90 days, a red alert appears. If the last update statistics is older than 30 days, a yellow alert appears; if older than 60 days, a red alert appears. *Caches*: The data cache and converter cache hit rates are determined every 120 seconds. If the data cache hit rate falls below 99%, a yellow alert appears; below 96%, a red alert appears. If the converter cache hit rate falls below 98%, a yellow alert appears; below 94%, a red alert appears. Use transaction `RZ21` to change threshold values. *Bad Indexes*: Checks whether there are any corrupt indexes in the database. Corrupt indexes cannot be used for the strategy selection and must be re-created using the administration tool. If there is a defective index in the system, a red alert appears.
Backup/Restore	Indicates whether the existing data backups satisfy the security criteria for your database system. *Data Backup*: *Last Data Backup*: Indicates how old the last successful data backup is (in days). If the backup is more than seven days old, a yellow alert appears; if older than 14 days, a red alert appears. You can also use the DBA action log display. Double-clicking an alert in this transaction brings you directly to the display in transaction `DB12`. If necessary, schedule a backup in the DBA Planning Calendar. *Backup Return Code*: Checks whether the last backup was successful. If the last backup had errors, a red alert appears.
Health	Indicates whether the system is error-free. *Data Devspaces*: This alert is not yet supported.

See also:

Database Monitor [Seite 585]

Detail Analysis Menu [Seite 592]

Statistics (SAP DB)

You can display database fill level and activity information.

Database Fill Level (SAP DB)

These statistics provide you with information about the current database fill level and the level over the last four weeks. A collector logs the database fill level once a day providing you with an overview of database growth over the last month. To display the statistics graphically, choose *Graphics*.

Database Activity (SAP DB)

Overview of database activity over the last four weeks (collected daily by the collector).

Choose *More* to switch between the following displays:

- Total number of SQL statements, commits, rollbacks, and prepares
- Number of database reads and writes
- Cache hit rates
- Number of table scans as well as log activity

The values indicate the numbers since the last database restart.

To compare the number to that of the previous day, choose *Delta*.

To display the statistics graphically, choose *Graphics*.

Diagnosing Performance Problems (SAP DB)

A Transaction is Running Very Slowly (SAP DB)

Use

As a database administrator, end users will often ask you to investigate why a particular transaction or set of transactions is running slow. There are many factors to take into account when tracking down these types of problems.

Prerequisites

Your search may take a long time and you should therefore make sure you have as much background information as possible before starting.

Ask the users for the following information:

- Has the transaction always been slow or did you only recently notice the slowdown?
- Is it a new program or transaction?
- Is the slowdown only during peak periods or is it fairly constant?
- Has the user workload changed recently?
- Does it appear that just this one transaction is slow or are other transactions or applications also now performing poorly?

Procedure

Attempt to localize the performance bottleneck using the information you obtained.

If you feel this issue may be isolated to one particular transaction, program or application, you may also need assistance from the application developers. They will better understand the process flow of the application and can help to change and test statements in the program.

Use the following analyzes:

- Determine if Exclusive Lock Waits [Seite 604] have occurred
- Check the tables involved in the transaction for possible Missing Indexes [Seite 684].
- Check the cache hit rates and table scan speeds using Bottleneck Analysis [Seite 611].
- Use the Diagnostic Monitor [Seite 609] to perform runtime analyzes of any long–running SQL statements.

Missing Indexes (SAP DB)

Use

Indexes are essential for fast efficient access to the serverdb. You should make regular checks to determine whether indexes are missing from the database system. This is particularly important after major changes, such as an upgrade.

There are two types of inconsistencies:

- Indexes are found in the serverdb, but not in the R/3 Repository
- Indexes are defined in the R/3 Repository but are not found in the server database

Procedure

Use the Tables/Indexes [Seite 605] section in the Detail Analysis Menu [Seite 592] to determine whether any indexes are missing.

The available functions are:

Checks	Updates the comparison between the R/3 Repository and the database system
Detailed analysis	Detailed index analysis for a table
Roots	Detailed index analysis for a table. The system reports the root page number for the B*tree created for the index.
Missing indexes	Detailed list of the missing indexes

Always recreate missing indexes immediately as they are very important for database system function.

- Primary indexes (ending with 0) ensure that the line keys (row keys) are unique. Missing primary indexes is therefore a critical problem.
- Secondary indexes (do not end with 0) are used for special scans and are only relevant to performance.

To create missing secondary indexes, use the database utility. For indexes that are defined in the ABAP Dictionary but are not in the server database, you can start the database utility directly from the list by choosing *Create in DB* (Creating Objects in the Database [Seite 375])

To create missing primary indexes, you can also use the database utility. However, you should make a detailed analysis of the situation before you use the database utility. If duplicate keys have already been inserted in the table, for example, it is no longer possible to create the index. You must identify the incorrect keys and delete them. In difficult cases, contact SAP regarding the restoration of the index.

See also:

Database Alert Monitor [Seite 677]

Checking Database Free Space (SAP DB)

Regularly check the available free space in the data area of the database system.

Suggestion 1

Monitor *SpaceManagement* using the Database Alert Monitor [Seite 677] to recognize space problems in time.

Suggestion 2

1. Go to the Detail Analysis Menu [Seite 592].
2. Choose Tables/Indexes [Seite 605].
3. Analyze the entries in the *DB filling* area. This will tell you how many pages in your database system are used.

Suggestion Three

1. Go to the Detail Analysis Menu [Seite 592].
2. Choose DB Filling Statistics [Seite 598].
3. Analyze the database fill level entries.
4. Select *History*. The system displays daily database fill level statistics for the last month.
5. You can also display this information graphically. To do this, choose *Graphics*.

Activities

Provide enough memory space for the affected devspaces; use the database manager to set this.

Also regularly monitor the log area. Proceed as described in Checking Free Log Space [Seite 687] .

Checking Free Log Space (SAP DB)

Regularly check the available free space in the log area of the database system.

Suggestion 1

Monitor *SpaceManagement* using the Database Alert Monitor [Seite 677] to recognize space problems in time.

Suggestion 2

6. Go to the Detail Analysis Menu [Seite 592].
7. Choose Logging Statistics [Seite 601].
8. Analyze the log area fill level information.

Suggestion 3

1. Go to the DBA Action Logs [Seite 1289] display.
2. In the display area *Log area*, choose *State*.
3. Analyze the log area fill level information.

Activities

Provide enough memory space for the log area. Schedule log backups using the DBA Planning Calendar [Seite 1273].

If possible, activate automatic log backup (Automatic Log Backup [Seite 1280]). This ensures regular backup of the log area entries.

Also regularly monitor the data area. Proceed as described in Checking Available Database Free Space [Seite 686] .

Consistency Checks (SAP DB)

There are two types of inconsistency:

- Objects exist in the serverdb, but not in the R/3 Repository
- Objects are defined in the R/3 Repository but are not found in the server database

Procedure

Use the Tables/Indexes [Seite 605] section in the Detail Analysis Menu [Seite 592] to determine whether any database objects are inconsistent.

The available functions are:

Checks	*Database* <-> *ABAP Dictionary*: updates the comparison between the R/3 Repository and the database system *R/3 Kernel:* consistency check in the kernel
Detailed analysis	Detailed index analysis for a table
Roots	Detailed index analysis for a table. The system reports the root page number for the B*tree created for the index.
Missing indexes	Detailed list of the missing indexes

Choose *Checks* → *Database* <-> *ABAP Dictionary*

You can use already existing consistency checks or create a new one.

The system provides you with the following information:

- A list of objects missing from the database
- Objects that the ABAP Dictionary does not recognize
- Results of other checks
- Optional indexes

See also:

Missing Indexes [Seite 684]

Database - ABAP Dictionary Consistency [Seite 371]

Creating Objects in the Database [Seite 375]

Displaying Object Definitions [Seite 376]

Consistency Check for Structures in the Database System

You can schedule the following consistency checks in the DBA Planning Calendar [Seite 1273]:

- Consistency check for database and index structures (*Check database structure (all objects)*
- Consistency check for database table structures (*Check database structure (only tables)*

For information about planning consistency checks, see Adding an Action [Seite 1282]

To display the logs for the consistency checks, use the display in DBA Action Logs [Seite 1289]

SAP *live*Cache – Monitoring and Administration

Use the Computing Center Management System (CCMS) to display information about the status of the individual *live*Caches within an SAP application system (for example, APO system).

You can display the various *live*Caches assigned to the application server and choose one of them. After you choose a *live*Cache, you can do the following:

liveCache: Connection [Seite 691]

liveCache: Administration [Seite 693]

liveCache: Configuration [Seite 699]

liveCache: Console [Seite 701]

liveCache: Monitor [Seite 717]

liveCache: Performance [Seite 724]

*live*Cache: Connection

Use

*live*Cache connections are usually created during installation, but can be created, changed, or deleted later.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

`Logical liveCache name`	Name of the *live*Cache; enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
`Physical liveCache name`	Freely-selectable database name; caution: Do not use hyphens in the names.
`liveCache server`	Name of the server the *live*Cache is installed on.

Creating connections

Choose *liveCache* → *Create/Change/Delete connection*.

Fill in all the required fields in the *liveCache Connection Information* window.

Choose *Connection* → *Save.*
A message indicating the connection was successfully created appears.

Changing connection

You can change the physical *live*Cache name and the name of the *live*Cache server.

After changing a connection, you must stop all the R/3 application servers that access the *live*Cache and then restart these servers. Otherwise, there may be severe connection problems.

Choose *liveCache* → *Create/Change/Delete connection*.

Fill in all the required fields in the *liveCache Connection Information* window.

Choose *Connection* → *Save.*
A message indicating the connection was successfully changed appears.

Stop and then restart all R/3 application servers that access the *live*Cache.

Deleting connection

Choose *liveCache* → *Create/Change/Delete connection*.

Choose *Connection* → *Delete.*
After confirming, a message indicating the connection was deleted appears.
The *live*Cache can no longer be accessed from the R/3 System.

Configuring application-specific connection reports

You can configure application-specific reports before or after starting, initializing, or stopping the *live*Cache.

Choose *liveCache* → *Create/Change/Delete connection*.
Fill in all the required fields in the *Configuration of Application-Specific Functions* window.

Report names for pre- and postprocessing must be unique.

Choose *Connection* → *save.*
The report configuration is saved.

*live*Cache: Administration

Use

Start, stop, and initialize the *live*Cache.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Administration*.

You can analyze the selected *live*Cache using the following functions:

System Messages [Seite 725]

liveCache Attributes [Seite 726]

Starting the liveCache [Seite 694]

Stopping the liveCache [Seite 695]

Initializing the liveCache [Seite 696]

Starting the *live*Cache

Use

Start the *live*Cache (change to condition `WARM`)

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Administration*.
4. Choose *Administration* → *Start liveCache*

To schedule a start of the *live*Cache in the background, proceed as follows:

1. Call transaction `SE38`.
2. Enter report **`RSLVCSTART`**.
3. Choose *Execute*.
4. Enter the following:

   ```
   liveCache name
   liveCache server
   ```
5. Choose *Execute*.

Example

`liveCache name`	LCAVC
`liveCache server`	p12345
`Background mode`	*X*

Result

The *liveCache* status is changed to `WARM`.

Stopping the *live*Cache

Use

Stop the *live*Cache (change to status `OFFLINE`).

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Administration*.
4. Choose *Administration* → *Stop liveCache*

To schedule a stop of the *live*Cache in the background, proceed as follows:

1. Call transaction `SE38`.
2. Enter the report name **`RSLVCSTOP`**.
3. Choose *Execute.*
4. Enter the following information:
   `liveCache name`
   `liveCache server`
   Confirm background processing (*Bmode=X*)
5. Choose *Execute*.

Result

All data currently in the data cache is saved to the data devspaces. This ensures the *live*Cache is always consistent.

The *liveCache* status is changed to `OFFLINE`.

Initializing the *live*Cache

Use

Initialize the *live*Cache.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache* / *COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Administration*.
4. Choose *Administration* → *Initialize liveCache*.

To schedule an initialization of the *live*Cache in the background, proceed as follows:

1. Call transaction `SE38`.
2. Enter the report name **`RSLVCINIT`**.
3. Choose *Execute*.
4. Enter the following information:
   `liveCache name`
   `liveCache server`
   Confirm background processing (*Bmode*=*X*)
5. Choose *Execute*.

Process Flow

1. The system checks the operation status of the *live*Cache If the *live*Cache has the status `WARM` or `COLD`, the system sets the *live*Cache to `OFFLINE`.
2. The system then sets the *live*Cache operation status to `COLD` and reinitializes it.
3. The system loads the *live*Cache system tables and activates *live*Cache monitoring.
4. The system loads the COM routines
5. The system installs the routines for the R/3 application system (for example, APO)

Result

The log `LCINIT.LOG` appears.

If an error occurs during the initialization process, the process is terminated and the log appears.

If the *live*Cache initialization fails, the *live*Cache is not activated. Fix the error and **restart the initialization process**.

System Messages

Use

Display *live*Cache log files.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools → APO Administration → liveCache / COM routines → Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache → Performance | Configuration | Administration* and then *System messages.* The *live*Cache logs appear.
4. To switch between displaying the old kernel log and the current kernel log, choose *Goto → liveCache messages → Current / Old*

`System messages`	Kernel log (`knldiag`) listing all messages in chronological order (for example, *live*Cache starts and stops, system error messages) When the log file reaches a certain size, the system overwrites the beginning of the file with new information. A new log file is created every time the *live*Cache is started.
`Old system messages`	Old *live*Cache log (`knldiag.old`). This log is created as a backup copy every time the *live*Cache is started
`System error messages`	Log of all errors (`knldiag.err`) occurring in the *live*Cache
`Initialization log`	Log (`lcinit.log`) of the last initialization

*live*Cache: Configuration

Use

Change and display the current configuration values of the *live*Cache data and log devspaces.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Configuration*.

After selecting a *live*Cache, you can use the following views and functions:

System Messages [Seite 725]

liveCache Attributes [Seite 726]

Extending the Data Area [Seite 700]

Extending the Data Area

Use

Extend the current data area configuration.

If the size of the data area is insufficient, you can add an additional data devspace. After making this change, the system adjusts all dependent parameters.

Prerequisites

You are assigned the APO administrator role in the CCMS.

The *live*Cache is in `WARM` or `COLD` mode.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Configuration*.

Choose *Configuration* → *Data area* and then *Edit* → *Change configuration*.

If the current configuration corresponds to the *live*Cache naming convention, the new devspace name appears in the dialog box.

Enter the following data:

`Size in KB`	Size of the new data devspace.
`Path`	Full path of the new data devspace. If you do not specify a path, the system creates it in the *live*Cache software directory.

Confirm your entries.

After confirming again, the system creates the new data devspace.

*live*Cache: Console

Use

Display information from the runtime analysis of *live*Cache bottlenecks.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Console*.

After selecting a *live*Cache, you can use the following views:

liveCache Attributes [Seite 726]

I/O Operations in the liveCache [Seite 702]

Process Analysis [Seite 703]

Space Management [Seite 710]

I/O Operations in the *live*Cache

Use

Display information about I/O activity in the individual *live*Cache devspaces.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Console*.

In the *liveCache Console* dialog box, choose the *Disk access* option and then *Console* → *Execute command*.

Process Analysis

Use

Display the current statuses of tasks.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Console*.

The following views for process analysis are displayed in the *liveCache Console* dialog box:

Process Overview [Seite 704]

Active Tasks [Seite 705]

Runnable Tasks [Seite 706]

Sleeping Tasks [Seite 707]

Task Activities [Seite 708]

Causes of Suspends [Seite 709]

Process Overview

Use

Details about the tasks in the user kernel processes.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Console*.
4. One the *liveCache Console* screen, choose `Process overview`.
5. Choose *Console* → *Execute command*.

Information about the current status of the database tasks is displayed .

`Running`	Running, using CPU
`Runnable`	Can run, but priority was lowered due to long runtime or higher priority of another task
`IO wait`	Waiting for I/O (`W`: write, `R`: read)
`Vwait`	Waiting for SQL lock
`Vbegexcl`	Waiting for access to protected memory
`Vendexcl`	Runnable after protected memory access (lock collision occurred)
`Vsuspend`	Waiting for B* tree lock (very short) or log I/O, explicitly activated
`Vsleep`	Short wait, automatically continues after a specified time interval
`Vopmsg`	A message is being written to the `knldiag`, `knldiag.err` and / or `opmsg[n]`files
`command wait`	Task is a database session without a request
`connect wait`	Free database session

See also: *R/3 database administration (CONTROL)* → *Performance analysis and tuning* → *SAP DB bottleneck analysis*

Active Tasks

Use

Display the status of all database tasks.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Console* and select the `Active tasks` option on the *liveCache Console* screen.

Choose *Console* → *Execute command*.

Detailed task information about every database task appears.

`running`	Running, using CPU.
`runnable`	Can be run, but its priority was reduced due to long runtime or higher priority of another task.
`IO wait`	Waiting for I/O (`W`:write, `R`:read).
`Vwait`	Waiting for SQL lock.
`Vbegexcl`	Waiting for access to protected memory.
`Vendexcl`	Can be run after protected memory access where lock collision occurred.
`Vsuspend`	Waiting for B* tree lock (very short) or log I/O, explicitly activated.
`Vsleep`	Short wait, will automatically continue after the specified time interval.
`Vopmsg`	Message written to `knldiag`, `knldiag.err` and / or `opmsg[n]` files.

Runnable Tasks

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Console* and select the `Runnable tasks` option on the *liveCache Console* screen.

Choose *Console* → *Execute command*.

Detailed task information about every runnable task appears.

`runnable`	Can be run, but its priority was reduced due to long runtime or higher priority of another task.

Sleeping Tasks

Use

Display all sleeping tasks.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Console* and select the `Sleeping tasks` option on the *liveCache Console* screen.

Choose *Console* → *Execute command*.

Detailed task information about every sleeping task appears.

`Vsleep`	Short wait, will automatically continue after the specified time interval.

Task Activities

Use

Display all available tasks.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Console* and select the `Task activities` option on the *liveCache Console* screen.

Choose *Console* → *Execute command*.

Detailed task information appears, including information about special tasks.

`running`	Running, using CPU.
`runnable`	Can be run, but its priority was reduced due to long runtime or higher priority of another task.
`IO wait`	Waiting for I/O (`W`:write, `R`:read).
`Vwait`	Waiting for SQL lock.
`Vbegexcl`	Waiting for access to protected memory.
`Vendexcl`	Can be run after protected memory access where lock collision occurred.
`Vsuspend`	Waiting for B* tree lock (very short) or log I/O, explicitly activated.
`Sus...`	Special suspend situations.
`Vsleep`	Short wait, will automatically continue after the specified time interval.
`Vopmsg`	Message written to `knldiag`, `knldiag.err` and / or `opmsg[n]` files.
`Command wait`	Task is a database session without request.
`Connect wait`	Free database sessions.
`Asyn Wait`	Task waiting for I/O save/restore activity.

Causes of Suspends

Use

Detailed analysis to determine the causes of the Vsuspend.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Console* and select the `Causes of suspends` option on the *liveCache Console* screen.

Choose *Console* → *Execute command*.

Detailed task information about every task in Vsuspend appears.

`Vsuspend`	Waiting for B* tree lock (very short) or log I/O, explicitly activated.

Space Management

Use

Display data relevant for space management.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Console*.

The following views for space management are displayed in the *liveCache Console* dialog box:

Space Use [Seite 711]

PSE Statistics [Seite 712]

PSE Data [Seite 713]

Synchronization Objects [Seite 714]

Runtime Environment [Seite 715]

Time Use [Seite 716]

Space Use

Use

Display information about memory allocation to the database kernel, the task administrator, and various dynamic data pools.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Console.*
4. One the *liveCache Console* screen, choose `Memory use`.
5. Choose *Console* → *Execute command*.

PSE Statistics

Use

Display PSE statistics.

Prerequisites

You are assigned the APO administrator role in the CCMS.

You are using Intel's PSE36.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Console* and select the `PSE statistics` option on the *liveCache Console* screen.

Choose *Console* → *Execute command*.

Statistics about the use of this extended memory area appear.

PSE Data

Use

Display PSE data.

Prerequisites

You are assigned the APO administrator role in the CCMS.

You are using Intel's PSE36.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Console* and select the `PSE data` option on the *liveCache Console* screen.

Choose *Console* → *Execute command*.

Information about the use of this extended memory area appear.

Synchronization Objects

Use

Displays statistics for semaphores and caches, which control access to main memory.

These statistics track cache activity and any resource conflicts. These conflicts occur when multiple requests are sent to the same resource simultaneously.

Serious bottlenecks occur when the number of collisions is more than 10% percent of the total number of accesses.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Console* and select the desired information.
4. Choose *Console* → *Execute command*.

Runtime Environment

Use

Display details about basic *live*Cache processes.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache* / *COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Console*.
4. One the *liveCache Console* screen, choose `Runtime environment`.
5. Choose *Console* → *Execute command*.

The operating system process IDs for the *live*Cache and its resources appear. All processes are listed in the upper section. The lower section contains only those processes and tasks required for hard disk access.

Time Use

Use

This view is used by SAP DB support and is only activated in special cases.

*live*Cache Monitor

Use

Analyze the current status of the *live*Cache.

Integration

On the *liveCache: Initial Screen*, choose *liveCache monitor* (SAP/liveCache – Monitoring and Administration [Seite 690]).

Activities

OMS/liveCache Base Layer [Seite 718]

Class Container [Seite 720]

Caches [Seite 722]

SQL Statistics [Seite 723]

OMS/liveCache Base Layer

Use

Display information about OMS events and information about the base *live*Cache layer for each database procedure.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Monitor*.
4. Choose *liveCache*→ *OMS/liveCache base layer*.

The following information appears:

Column	**Explanation**
DB Procedure Name	Name of the DB procedure called
Calls	Number of calls of the DB procedure
Runtime (average – in ms)	Average runtime in milliseconds
OMS Dereferencing	Number of dereferences, that is, the number of accesses of object instances using OID
LC Dereferencing	Number of dereferences sent from the *live*Cache base level (that is, could not be satisfied by the OMS)
Key Dereferencing	Number of dereferencings through keys
OMS Save Operations	Number of save operations by object instances
LC Save Operations	Number of save operations executed by the *live*Cache base layer
OMS Delete Operations	Number of deleted object instances
LC Delete Operations	Number of object instances deleted by the *live*Cache base layer
LC Lock Operations	Number of lock operations
LC Log Accesses (by dereferencing)	Number of dereferencings that requested access to the log
Exceptions	Number of C++ exceptions that occurred

OutOfDateExceptions	Number of out-of-date exceptions. These occur when a lock is requested for an instance, which has already been changed in the interim
Timeouts	Number of timeouts that occurred due to waiting for locks
Rollbacks (Subtransactions)	Number of subtransactions rolled back
Commits (Subtransactions)	Number of subtransactions ended using COMMIT
Maximum Nesting (Subtransactions)	Maximum nesting of subtransactions
Loaded object instances of variable length	Number of loaded object instances of variable length
Saved object instances of variable length	Number of saved object instances of variable length
Average length of variable object instances	Average length of a variable length object instance
OMS Cache Size	Size of the OMS cache in bytes
Interface GUID	GUID of the interface that defines the methods

Class Container

Use

Display information about class containers.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Monitor*.
4. Choose *liveCache*→ *Class container*.

Result

- Information about class container appears. Each class container contains instances of exactly one persistent class.
- Information about the actions of the garbage collector for each container appears.

Column	Explanation
Class ID	Internal *live*Cache class number
Size of instance (in bytes)	Size of an instance within the class (in bytes)
Container size (in pages)	Size of the container in pages
Free pages (pages containing deleted instances)	Number of pages that can accept instances
Empty pages (release possible)	Number of empty pages in the container. These can be released by the garbage collector.
Free EOT pages (pages containing instances marked as deleted)	Number of pages containing instances marked as deleted
Objects per page (average)	Percentage of page filled
Checked objects (garbage collector)	Number of instances checked by the garbage collector
Deleted objects (garbage collector)	Number of instances deleted by the garbage collector
Class GUID	GUID of the class

Caches

Use

Provides information about the caches in the *live*Cache instance.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Monitor*.
4. Choose *liveCache*→ *Caches*.

Result

The system displays the sizes of the most important caches as well as access statistics since the *live*Cache instance was started.

Some of the parameters	Hit rate to achieve
DATA CACHE SQL HIT RATE (%)	100
DATA CACHE OMS HIT RATE (%)	100
DATA CACHE OMS LOG HIT RATE (%)	0
OMS LOG CACHE HIT RATE (%)	100
CONVERTER CACHE HIT RATE (%)	100

See also:

Data Cache [Seite 731]

SQL Statistics

Use

Display statistics of SQL activities.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache* / *COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Monitor*.
4. Choose *liveCache*→ *SQL statistics*.

Result

The information includes the following:

- Number of commands executed (SQL COMMANDS)
- Number of commits and rollbacks (COMMITS, ROLLBACKS)
- Number of insert, update, and delete operations (INSERTS, UPDATES, DELETES)

*live*Cache: Performance

Use

Display statistics for analysis of *live*Cache activities.

This provides you with an overview of *live*Cache performance and supports your monitoring of the *live*Cache.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Performance*.

You can analyze the selected *live*Cache using the following functions:

System Messages [Seite 725]

liveCache Attributes [Seite 726]

Cache Hit Rates [Seite 727]

Disk Access [Seite 728]

Status [Seite 729]

Detail Analysis [Seite 730]

System Messages

Use

Display *live*Cache log files.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

5. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
6. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
7. Choose *liveCache* → *Performance* | *Configuration* | *Administration* and then *System messages.* The *live*Cache logs appear.
8. To switch between displaying the old kernel log and the current kernel log, choose *Goto* → *liveCache messages* → *Current / Old*

`System messages`	Kernel log (`knldiag`) listing all messages in chronological order (for example, *live*Cache starts and stops, system error messages) When the log file reaches a certain size, the system overwrites the beginning of the file with new information. A new log file is created every time the *live*Cache is started.
`Old system messages`	Old *live*Cache log (`knldiag.old`). This log is created as a backup copy every time the *live*Cache is started
`System error messages`	Log of all errors (`knldiag.err`) occurring in the *live*Cache
`Initialization log`	Log (`lcinit.log`) of the last initialization

*live*Cache Attributes

Use

Display the most important *live*Cache attributes, used for performance and error analysis.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Performance* | *Administration* | *Console* | *Configuration*.

After selecting a *live*Cache, a list of attributes is displayed for the selected *live*Cache:

`liveCache name`	Name of the *live*Cache
`liveCache server`	Name of the *live*Cache server
`Installation directory`	Directory (DBROOT) where the *live*Cache software is installed
`DBMServer version`	Version of the DBMServer
`liveCache version`	Version of the *live*Cache software
`liveCache mode`	Current operation mode of the *live*Cache Green: *live*Cache is active (mode `WARM`) Yellow: *live*Cache can only be used by the administrator (mode `COLD`) Red: *live*Cache is inactive (mode `OFFLINE`)

Cache Hit Rates

Use

Display the important cache hit rates since the last system start.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Performance*.

To display the OMS data cache and converter cache hit rates for the selected *live*Cache, choose *Cache hit rates*.

Disk Access

Use

Display *live*Cache disk access information.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Performance*.

To display the data and log devspace read and write access information of the selected *live*Cache, choose *Disk access*.

Status

Use

Display performance-critical variables for analysis purposes.

Prerequisites

You are assigned the APO administrator role in the CCMS.

The *live*Cache monitor is activated.

Expect performance bottlenecks when the monitor is activated.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Performance*.

Status displays the performance-critical variables of the selected *live*Cache:

`Data cache use = 100 %`	Indicates performance bottlenecks; requires detailed data cache analysis.
`Checkpoint requested = Yes`	Can have negative impact on performance as all caches are written to the disks

Detail Analysis

Use

Detailed analysis of the *live*Cache.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Performance*.

You can analyze the following areas:

Data Cache [Seite 731]

Log Caches [Seite 732]

Devspace Statistics in the liveCache [Seite 733]

liveCache Configuration [Seite 735]

Active Parameters [Seite 736]

OMS Versions [Seite 737]

Transactions [Seite 738]

Database Procedures [Seite 739]

Data Cache

Use

Information about the data cache.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Performance*.
4. Choose *Performance* → *Data cache analysis*.

The following information is displayed:

`Data cache configuration`	• Size (specified using page size of 8 KB) and load (used and free area) of the data cache. The size of the data cache depends on the amount of data in the *live*Cache and on the application. Size of the data cache (in the case of a *live*Cache installation on a server separate from the application server): – Approximately 70% of the available physical memory space – 200 <= <Size of the data cache> <= MAXDATAPAGES <= 2,147,483,640 – MAXUSERTASKS*10 <= <Size of the data cache>
`Data cache use`	Distribution of data cache use by OMS and SQL data and OMS and SQL logs. Display of pages changed since last *live*Cache restart.
`Data cache access`	Statistics about *live*Cache access. The OMS data and SQL data hit rates should be 100%.

See also:

Caches [Seite 722]

Log Caches

The *live*Cache uses the following log caches to store logging information:

- OMS rollback cache
  Object logging information
- SQL rollback cache
  SQL logging information

Both rollback caches use a log queue. Log information is stored in the log queues before it is written to the log disk.

Use

Information about the log caches (rollback caches).

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Performance*.
4. Choose *Performance* → *Log cache analysis.*

This provides you with the following rollback cache information:

`OMS log cache configuration`	Size of the OMS rollback cache and the OMS log queue
`OMS rollback cache accesses`	Statistics about accesses to the OMS rollback cache

For information about the SQL rollback cache, choose *Edit* → *SQL log analysis.*

Devspace Statistics in the *live*Cache

Use

Display information about the fill level of the disk areas (data and log devspaces).

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache* / *COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Performance*.
4. Choose *Performance* → *Devspace statistics*.

The following information is displayed:

Data Space Statistics

`Total Data Space`	Configured size of all data devspaces
`Max Allowed Perm Page No`	Highest possible data page number that can store permanent data
`Used Data Space`	Used data devspace
`Used Perm. Data Space`	Amount of data devspace used for permanent data
`Used Temp. Data Space`	Amount of data devspace used for temporary data
`Free Data Space`	Free space in data devspace
`Last Data Page No`	Highest page number used up to this point

Log Space Statistics

`Total Log Space`	Size of the total log area
`Total SQL Log Space`	Amount of the entire log area used for SQL logging

SQL Log Space Statistics

`Total SQL Log Space`	Amount of the entire log area used for SQL logging
`Used SQL Log Space`	Amount of SQL log space used
`Free SQL Log Space`	Amount of free space in SQL log

To analyze the fill level of log devspace, choose *Edit* → *Detail Log*.

*live*Cache Configuration

Use

Display the current *live*Cache configuration.

Prerequisites

You are assigned the APO administrator role in the CCMS.

Procedure

In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).

In the *live*Cache dialog box, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.

Choose *liveCache* → *Performance*.

Choose *Performance* → *Configuration*.

An overview of the *live*Cache configuration appears.

Change Configuration [Seite 699].

Active Parameters

Use

Display *live*Cache configuration parameters.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Performance*.
4. Choose *Performance* → *Active parameters*.

A list of the active parameters for the selected *live*Cache appears.

Displaying the OMS Versions

Use

When a version is defined, the system creates a named copy of the consistent *live*Cache state valid at that time. The version can then be activated in the Object Management System (OMS) at any time. After the version is activated, all succeeding OMS operations will use the activated copy. Changes can be made. However, the changes will only be visible within the version.

The *live*Cache does not copy all the data, but only the data changed in the copy.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Performance*.
4. Choose *Performance* → *OMS versions.*

Result

An overview of the existing versions including their age (in hours) and the creation date appears.

Transactions

Use

Display active transactions.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Performance.*
4. Choose *Performance* → *Transactions*.

A list of all open transactions in the *live*Cache appears.

Database Procedures

Use

Display available database procedures.

Prerequisites

You are assigned the role of APO administrator in the CCMS.

Procedure

1. In the */SAPAPO/LCAD_SUPERUSER* role, choose *Tools* → *APO Administration* → *liveCache / COM routines* → *Monitor* (transaction `LC10`).
2. In the *live*Cache selection screen that appears, enter the name of the *live*Cache (case-sensitive) or choose one from the list of existing *live*Cache names.
3. Choose *liveCache* → *Performance.*
4. Choose *Performance* → *Database procedures*.

A list of all the database procedures loaded in the *live*Cache appears including their input and output parameters. You are also provided with information about the corresponding COM routine.

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:

Reset	Temporarily sets the displays in the Server Engine/Elapsed and SQL Requests sections to zero.
Since Reset	Displays the values in the Server Engine/Elapsed and SQL Requests sections since you reset the display.
Refresh	Returns the main screen to the current values.
Totals per Second/Absolute Totals	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 [Seite 741]	Overview of memory configuration and performance.
Space Usage [Seite 743]	Shows disk space availability for the R/3 database.
Server Engine/Elapsed [Seite 744]	Shows CPU consumption and disk I/O.
SQL Requests [Seite 746]	Overview of SQL Server request statistics.

See also Detail Analysis Menu [Seite 747].

Memory Usage

Use

In the SAP/SQL Server Database Monitor, the Memory Usage section is divided into the following areas:

Current memory

The current size of the virtual memory used by SQL Server is shown here in KB. This memory includes areas for the data cache, the procedure cache, open objects, locks, and connections. If the parameter *Set working set size* is set to 1, the value displayed is the physical memory used by SQL Server.

Maximum memory

The maximum amount of virtual memory used for the SQL Server since its start.

Memory setting

Memory usage is determined by the SQL Server parameters *min server memory* and *max server memory*.

This area shows the memory allocation strategy that SQL Server uses. There are three possible values

FIXED:	The memory setting is FIXED if *min server memory (MB)* is equal to *max server memory (MB)*. SQL Server will use a constant amount of memory, specified by these two parameters. This is the recommended setting for R/3. If FIXED is not set, and an R/3 instance is running on the same server, R/3 and SQL Server will compete for memory.
AUTO:	SQL Server will dynamically allocate memory it requires between 4 MB and 2 PB. *min server memory (MB) = 0* and *max server memory (MB) = 2147483647*. AUTO is the recommended setting for a standalone database server, that is, a server without an R/3 instance.
RANGE:	SQL Server can dynamically allocate memory between *min server memory (MB)* and *max server memory (MB)*. SQL Server will use memory in a range between these 2 parameter settings.

We recommend that you set the SQL Server parameter *set working set size* to 1, so that the allocated memory will always reside in the physical memory.

Total SQL connections

All the connections on the SQL Server database. This includes the R/3 connections, the SQL Agent, and potential Query Analyzer connections.

R/3 connections

The number of connections from the SAP work processes to the SQL Server database.

Each work process has multiple connections to the SQL Server database.

Free buffers

The number of free data buffers available. These buffers can be used either by the data cache or by the procedure cache.

Procedure cache

The size in KB of the procedure cache, which contains stored procedures, execution plans, etc. In SQL Server 70, this value is adjusted automatically.

Data cache size

The size in KB of the data cache memory. Data and index pages are stored in the data cache so that they do not need to be stored on disk.

Data cache hit ratio

The data hit ratio is the main performance indicator for the data cache.

The hit ratio is the percentage of pages found in the cache without having to read from the disk. The ratio shows the average percentage of data pages found in the cache since SQL Server was started.

The hit ratio is a mean value that gives a snapshot of a short period of time before the analysis. This value should always be above 95, even during periods of heavy workload. If the hit ratio is significantly below 95, the data cache could be too small.

Always check the history of these values. Choose *Detail analysis menu* → *Performance database*. A snapshot is collected every 2 hours. Check the column *Data cache hit %.*

Space Usage

Use

Database space is organized in several database files on disk.

In the SAP/SQL Server Database Monitor, the Space Usage section is divided into the following areas:

Total data size

The size of the R/3 database in MB.

Total log size

The size in MB of the logfiles for the R/3 database.

Free space

The free space is displayed in the R/3 database and its log files in MB.

Server Engine/Elapsed

Use

The time elapsed since the last monitor reset is displayed in the format *seconds (days : hours : minutes : seconds)*. To be sure that enough throughput was processed by the database, check that *CPU busy* displays a minimum of 500 seconds.

The elapsed time in the display can be greater than the time displayed for *CPU idle*, which is the sum of idle time for all the CPUs used by SQL Server.

The system displays the statistics cumulated since SQL Server startup. These cumulated values are normally not very informative as the high amount of idle time during the night balances out potential bottleneck situations during the day when the workload is high. You should therefore refresh the display to show values from a representative time frame. Initially and after a refresh, the system displays the time elapsed since the SQL Server started.

In the SAP/SQL Server Database Monitor, the Server Engine/Elapsed section is divided into the following areas:

CPU busy

The number of seconds used by all available CPUs on the database server. That is, the number of seconds that the CPUs have been running SQL Server threads. This value is measured starting from the last time SQL Server was started, or the last time the statistics were reset.

Each CPU counts separately. For example, if during 2 seconds, 3 CPUs are used, CPU busy will be 6.

The CPU busy time should be always less than 70%. CPU idle should be greater than CPU busy divided by 2.

IO busy

Part of CPU busy. Shows the CPU time that was used by all available CPUs for I/O operations issued by SQL Server. That is, the number of seconds that SQL Server has spent doing I/O operations since SQL Server was started, or the last time the statistics were reset.

CPU idle

The number of seconds that all available CPUs in SQL Server were not running SQL Server threads. The CPU idle time is measured from the last time SQL Server was started, or the last time the statistics were reset. For this time, the CPUs may not have been strictly idle, as they may have been used for R/3 work processes.

Data About Disk I/O

The disk I/O is the most time-intensive operation for the database system. You can make significant performance gains by using fast disks and I/O controllers, and by physically separating files with high I/O activity so that they can operate concurrently.

Physical reads

The number of disk reads done by SQL Server since it was last started, or since the statistics were reset. This number is incremented every time a disk read operation is initiated. This includes both pages read for the data cache and procedure cache.

writes

The number of disk write operations by SQL Server since last startup, or since statistics were reset. Each time a disk write operation is initiated, this counter is incremented. The disk writes can be to the database files or to the log files.

errors

The total number of disk I/O errors encountered by SQL Server since last startup, or since the statistics were reset.

SQL Requests

Use

In the SAP/SQL Server Database Monitor, the SQL Requests section is divided into the following areas:

SQL batches	The number of Transact SQL command batches received by SQL Server. For the SAP R/3 work processes, a batch is created every time a database commit is required. This number is usually much smaller than the number of database requests issued by ABAP.
Read ahead pages	The number of requests that are read ahead, that is, asynchronously prefetched from disk.
Request buffer pages	The total number of requests for buffer pages, or data cache pages, issued by SQL Server. This is a logical request, so if the page exists in the cache, no physical read/write takes place. If the page is not in cache, then a physical page read will be recorded.
reads	The number of physical page reads into the data cache. The total number of pages, across all databases, is recorded.
writes	The number of physical page writes from the data cache. This is the total number of pages written across all databases.
Full table/index scans	The number of full table or index scans. These scans can be either base-table or full-index scans.
Index range scans	The number of qualified range scans through indexes. This number represents logical, as opposed to physical, access.
Index searches	The number of index searches. Index searches are used to start range scans and single index record fetches.
Probe scans	The number of probe scans that occur in a database. Probe scans are used to find rows in an index or base table directly.
Lazy write	The number of buffers written by the buffer manager's Lazy Writer. The Lazy Writer is a system process whose task is to write dirty buffers to disk when I/O activity is low. The goal of the Lazy Writer is to minimize the time needed by checkpoints to write dirty data pages to disk.

Detail Analysis Menu

Use

In the Detail Analysis Menu of the SAP/SQL Server Database Monitor, you can display information on the following areas:

Server Detail

This screen shows the general database information with greater detail than is displayed in the main screen.

SQL Processes

Shows all the threads needed by SQL Server, including all connections made by R/3 work processes.

To sort the display by work process, choose *Process monitor* → *Grouped output*. The work process ID is displayed in the column *Host PID*.

You can display the currently executed statement for active threads (threads with status *runnable*). Mark a row, then choose SQL Statement.

If a blocking situation occurs, the column *Blcked* shows the SQL Server process connection that is blocked and the process that is blocking it All processes that are not displayed as zero in this column are being blocked by another process.

As in the main screen, you can *Refresh* and *Reset* and display changes since reset.

SAP Stats on SPs

Displays the collected statistics for name cache and stored procedure execution.

R/3 with SQL Server executes all ABAP SQL statements as stored procedures.

Each R/3 server has a stored procedure name cache in R/3. This stored procedure name cache can contain statistics about stored procedures.

If an R/3 System consists of more than one server, you can display information for your local server or for all servers in the R/3 System. In a central R/3 System, the name cache statistics are displayed first. In a non-central R/3 System, choose *Name Cache Stats* to display them. To switch name cache statistics collection on or off, choose *Set statistics* → *On* or *Off*.

Choose *SP statistics* to display the selection screen for stored procedure statistics. You can restrict your selection or choose *Execute* to display all the statistics. Each stored procedure executed is displayed in one row, showing the duration and frequency of execution. Refer to the F1 Help to find out what each column means.

To display the source text of a stored procedure, select the line with the stored procedure, then choose *Single detail*. Choose *Explain* to display the execution plan. Choose *Table detail* to display details of the table used by the stored procedure.

Exclusive Lockwaits

Normally no data is displayed here. You see the following message: *Currently no exclusive lockwaits found*.

Exclusive lockwaits are wait situations that are currently being caused by database locks.

A user holding a lock occupies an R/3 work process. If other users attempt to apply the same lock, these users will have to wait. During this time, these users occupy their own R/3 work process. This waiting for a lock is known as a lockwait. As the number of lockwaits increases, fewer and fewer R/3 user requests can be processed by the available R/3 work processes. In the worst case, that is when the number of lockwaits equals the number of R/3 work processes, a small number of users can cause the entire R/3 System to freeze.

If exclusive lockwaits occur, both blocking and blocked processes are displayed. For each blocking situation, a tree of waiting processes is shown Column Hpid shows the process IDs of the host processes. Usually, this is the process ID of the R/3 work process. The lock holders are displayed furthest left, that is, they are not indented.

To display the SQL statement on the database, mark a row, then choose *SQL statement*.

Blocking Lockstats

Displays a history of blocking situations. This function shows blocking situations in the past as opposed to *Exclusive Lockwaits*, which shows the current situation. To collect this history, a SQL Server task called *SAP CCMS Blocking Lockstats* runs once a minute. If blocking lockstats is switched off, this parameter is switched off.

We recommend that you keep blocking lockstats switched on, as the effect on system performance is negligible.

The system displays the name of the table where the blocking lock is held, the duration in minutes, and the start date of the blocking situation.

A regular weekly job, *SAP CCMS Cleanup Saplocks* deletes all blocking lockstats data that is older than 7 days.

Deadlock

Displays a selection screen to search for deadlocks.

The section *Execution selection* allows you to select *Single statistics* or *Count statistics*. *Single statistics* displays the detailed history of database deadlocks collected by R/3 for the last 7 days. *Count statistics* displays a summary of the statistics for all deadlocks since R/3 was installed.

Deadlocks occur far less frequently with SQL Server 7.0 than with SQL Server 6.5. This is because SQL Server 7.0 uses row level locking.

Error Logs

The SQL Server error log provides additional information about bottleneck situations and their causes.

You can alternatively use the Enterprise Manager to display the SQL Server error log, or display the error log directly from the file system directory *\mssql7\log*. For example, if SQL Server fails to start, you need to check the error log in that directory.

State On Disk

Displays information to help you monitor the database growth. This button calls the Database Performance Monitor (transaction DB02).

System Tables

Displays the content of important system tables.

SQL Parameters

Displays all the SQL Server parameters and change history.

Performance Database

You can check the history of the values displayed here. A snapshot is collected every 2 hours.

DB Utilities

From here, you can execute SQL Server stored procedures and DBCC commands.

DB Backup History

Calls the CCMS Backup Monitor (transaction DB12).

Here, you can also display the SQL Agent error log.

Operating System Monitor

Purpose

An R/3 instance runs within an operating system. The operating system provides the R/3 instance with the following resources:

- Virtual memory
- Physical memory
- CPU
- File system management
- Physical disk
- Network

Bottlenecks in these areas can significantly affect R/3 System performance. The CCMS operating system monitor allows you to monitor these resources.

The operating system monitor helps you to identify the cause of a performance problem. If the source of the problem is in the operating system, it can be analyzed further and resolved using external tools or other external means.

The performance indicators are

- CPU load average and utilization
- Memory utilization
- System swapping information (replaced by pool data in the OS/400 operating system monitor.)
- Disk utilization information
- LAN activity
- OS configuration parameters

See also:

Checking CPU Statistics [Seite 754]

Checking Memory Utilization [Seite 756]

Operating System Collector [Seite 769]

Calling the Operating System Monitor

Use

You use the operating system monitor to monitor system resources that are provided by the operating system. You can call the operating system monitor for the server that you are currently logged on to, or for another server.

Procedure

To Call the Operating System Monitor for Your Server:

1. Choose *CCMS* → *Control/Monitoring* → *Performance menu* → *Operating system* → *Local* → *Activity.* Alternatively, call Transaction ST06.

To Call the Operating System Monitor for Another Server:

1. Choose *CCMS* → *Control/Monitoring* → *Performance menu* → *Operating system* → *Local* → *Activity.* Alternatively, call Transaction ST06.
2. In the *SAPOSCOL Destination* window, select a server. The system displays performance indicators of the operating system:

   For information specific to the OS/400 Operating System Monitor see Pool Data in the OS/400 Operating System [Seite 766]

3. To display more details on the individual areas, select the appropriate line.

For the CPU, memory and swap space, the data displayed is from the last 24 hours. For the disk and LAN the system displays a list of the current data for every disk or LAN interface. The system indicates any problems that may have occurred.

If you have logged on to the SAP instance running on host HS0001, the system monitors this host. You should normally start the monitor on the central host system. This is usually the host system on which the database system is running.

Checking CPU Statistics

Use

The CPU statistics are important to ensure smooth system performance. The table below displays any CPU bottlenecks that have occurred.

Display	Procedure
Idle CPU 0%	Check the analysis for previous hours.
Load average >3	An average of more than 3 processes waiting in front of the CPU.

Procedure

1. To display the CPU processes that consume the most resources, choose *CCMS* → *Control/Monitoring* → *Performance menu* → *Operating system* → *Local* → *Activity.*
2. Then choose *Goto* → *Current data* → *Snapshot* → *Top CPU processes.*
3. Check the following additional performance factors:
    - Which processes are causing the load?

      Check whether all other processes with high CPU utilization (memlog, r3trans, nwengine, brbackup...) are necessary.
    - If high values for CPU time are displayed, the process may be in a loop.
    - The resident size of an SAP work process should be approx. 9-12 MB, depending on the size of the ztta/roll_area.

      SAP work processes with high CPU utilization can also be displayed with the SAP process overview. The SAP process overview shows the ABAP program that is using the CPU.

      SAPGUI should not run on the application server.

See also:

Operating system monitor [Seite 750]

Checking Memory Utilization

Use

The memory utilization statistics are important to ensure that system performance is running smoothly.

The system displays the following information for the last twenty four hours:

- Pages in per hour
- Pages out per hour
- Paged in (kb/hour)
- Paged out (kb/hour)
- Free memory per hour (the minimum, maximum and average kb)

As general rule of thumb in a small hardware configuration, CPU bottlenecks occur when free physical memory is less than 10 MB and/ or when page out activity is greater than 0.

These values may vary for different operating systems and system sizes.

Procedure

1. To check memory utilization, choose *CCMS* → *Control/Monitoring* → *Performance menu* → *Operating system* → *Local* → *Activity*.
2. Choose *Detail analysis menu* and then *Goto* → *Current data* → *Previous hours* → *Memory.*

See also:

Operating System Monitor [Seite 750]

OS/400 Wait States

Definition

When a job is waiting for a resource, it may wait in main storage or else be removed from main storage until the resource becomes available. A job waiting for a system resource may have the following wait states:

- Short wait
- Short wait extended
- Long wait

Short Wait (OS/400 Wait States)

A job in this wait state holds an available activity level while waiting for an activity to occur. A job can remain in short wait for a maximum of two seconds. Actions that cause short waits include:

- Sending a WRITE instruction to a display when `*NO` is specified on the *Defer Write* (`DFRWRT`) parameter.
- Sending break messages to workstations.
- Specifying `*YES` on the *Restore Display* (`RSTDSP`) parameter on display files.

When using remote lines, avoid actions that cause short waits since they can cause the wait time in main storage to be much longer for a job than if the job was waiting for resources at a local workstation.

Short Wait Extended (OS/400 Wait States)

A job moves to this wait state after being in a short wait for the maximum two seconds. If no activity occurs within this time, the system cancels the short wait, removes the job from the activity level, and puts the job into a long wait. In the performance reports, this job state transition is called a short wait extended.

Long Wait (OS/400 Wait States)

A job that immediately leaves the activity level is in what is called a long wait. During a long wait, the job leaves the activity level.

Examples of long waits are:

- **Key/think waits**
  This type of long wait occurs outside the activity level when a job completes its work assignment and returns to request more work. It is a user-defined time period allowing the user to decide what is to be entered and to type the data. When the job receives the new assignment, it attempts to run again. If no activity level space is available, the job becomes *ineligible*.
- **record lock conflicts**
  These occur when two or more jobs attempt to lock the same record of a file.

- **Distributed Data Management data requests**

See also:

Activity Level (OS/400) [Seite 761]

Activity Level (OS/400)

Definition

An activity level assures system resources for a job. A job that cannot get an available activity level is ineligible.

In interactive environments, there are usually more jobs running than there is space for them to run in. Before a job can become active, there must be sufficient space in main storage. Activity levels are set for each pool in the system, which restrict the number of jobs that can be in main storage at any one time.

If an activity level is available, the job becomes active and starts processing in main storage. If no activity level is available, the job becomes ineligible. When this happens, the job is placed in the ineligible queue [Seite 763] until an activity level becomes available.

See also:

OS/400 Wait States [Seite 758]

Ineligible Queue (OS/400)

Definition

Jobs waiting for an activity level [Seite 761] are placed in the ineligible queue.

You can avoid unnecessary job transitions and disk operations by correctly managing the ineligible queue. Without proper tuning, you can keep down the number of jobs in the queue and improve throughput and response time.

See also:

OS/400 Wait States [Seite 758]

Job Priority (OS/400)

Definition

Jobs can gain priority if they are held by locks.

If a job enters the long wait state as a result of something other than a lock conflict, it is normally placed behind all other jobs of equal priority already in the ineligible queue. A queue of this type is called a “first-in, first-out priority queue”.

If a job becomes ineligible after a short wait extended or long wait caused by a lock conflict, it is placed in front of jobs of equal priority already in the queue.

The most common causes of this change to the normal queue placement are:

- A job entered a long wait as a result of a lock conflict because it was active before the conflict occurred. If the wait was short, you may be able to get the job back into an activity level before all the objects the job was using are removed from main storage.
- When a job has been granted a lock, it leaves the wait state. If other jobs in the ineligible queue are to use the same object, they must wait until the object is once again available. Therefore, you want jobs holding locks on objects to use them and make them available for other jobs to use. To accomplish this, the job moves ahead of any potential requesters.

See also:

OS/400 Wait States [Seite 758]

Pool Data in the OS/400 Operating System

Definition

In the OS/400 operating system monitor, the swap space information is replaced by pool data.

The table below displays the states of jobs running in the system:

Job State	Description
Active	The job is in the main memory. It processes tasks requested by the application.
Wait	The job is waiting for a resource that is currently not available (see also OS/400 Wait States [Seite 758])
Ineligible	The job has tasks to process, but the system is currently unable to process any more requests.

See also:

Activity Level (OS/400) [Seite 761]

Ineligible Queue (OS/400) [Seite 763]

Job Priority (OS/400) [Seite 765]

PTF Check in the OS/400 Operating System Monitor

Use

The function *PTF check* compares all the PTFs installed on AS/400 with the Information APAR.

Features

This function shows a list of all machines on which you can perform the PTF check.

Activities

- You can start the PTF check from the screen *Analysis report for PTF* by double-clicking the respective host name.

  All the PTFs installed in the system (AS/400) are compared with the Information APAR stored in the directory `/usr/sap/trans/config`.

  See also:

  – Getting the Latest IBM Information APAR [Extern]

  – PTF Status Check [Extern].

- You can upload the Information APAR from a local file of the frontend PC to the respective AS/400 machine.

  a. Choose *Upload Information APAR*.

     The dialog box *Import Information APAR from a Local File* appears.

  b. Enter the path of the local file of the frontend PC from which the Information APAR is uploaded to the AS/400 machine.

  c. Confirm with *Transfer*.

     Do **not** use the HTML version. Use the version that is available from the sapserv (x) instead.

Operating System Collector

Definition

The operating system collector (OS collector), a program called `saposcol`, is a standalone program that runs in the operating system background independently of the R/3 instances. It collects data on the following operating system resources:

- Virtual and physical memory
- CPU
- Storage management
- Physical disk
- Network

The OS collector makes this data available to all R/3 instances on a single host through shared memory. A dialog work process reads the data from the shared memory.

This data can be viewed using the CCMS Operating System Monitor [Seite 750].

The SAPOSCOL Program

The `saposcol` program is part of the standard R/3 System. The two most significant differences between `saposcol` and other R/3 programs are:

- you need special authorization to use `saposcol`.
- a single `saposcol` per host can run independently of R/3 instances.

  `saposcol` can be started before the R/3 instance is started and can continue to run after the instance is stopped.

`saposcol` collects and records current data every 10 seconds and records hourly averages over the preceding 24 hours.

Another background job, `SAP_COLLECTOR_FOR_PERFORMANCE`, extracts performance data about the previous 24 hours out of the shared memory and writes it to a performance database. This data can be compared for one or more hosts.

Program RFCOSCOL

While `saposcol` collects operating system data while R/3 work processes are running, `rfcoscol` collects OS data from hosts **where no R/3 work process is running** as, for example, on dedicated database servers.

R/3 uses a Remote Function Call to remotely start `rfcoscol`, which then reads the `saposcol` data from the shared memory and uses a Remote Function Call (RFC) to forward it to the current transaction. The system administrator must maintain Special Remote Function Call destinations for this in the R/3 System.

See also:

SAPOSCOL on a UNIX Host [Seite 775]

Starting SAPOSCOL [Seite 798]

Stopping SAPOSCOL [Seite 800]

Installing SAPOSCOL and RFCOSCOL for a Dedicated DB Server [Seite 792]

Setting Up RFC Communication to RFCOSCOL [Seite 789]

Displaying Collector Data [Seite 787]

Controlling SAPOSCOL from the R/3 System [Seite 806]

Controlling SAPOSCOL from the Operating System [Seite 777]

OS Collector: Troubleshooting [Seite 803]

Special Features of Operating Systems

Definition

`saposcol` collects data in different ways depending on the operating system.

This section outlines the various prerequisites and special features of each platform, for example, which data is collected according to the *Details* flag. Note that some data is not available on all platforms and is marked as N/A in the operating system monitor, or with a 0 or a 1 in dialog mode.

AIX

There are no particular prerequisites for `saposcol` in AIX versions up to 3.2.5. However, from AIX version 4.1, `saposcol` requires an IBM interface for performance data. Alternatively, you can install one of the following IBM packages:

- Performance aid 2.1.3 for AIX 4.1 IBM 5696-899
- Performance toolbox 2.1.3 for AIX 4.1 IBM 5696-900 (includes additional programs which are not necessary for operating R/3.)

The following descriptions apply to AIX 4.1.

Data collected according to the details flag:

Not all *Top CPU consumer* data is collected in the *Details not required* mode. The program name is displayed, not the command. Therefore, you only see `disp+work` instead of `dw.sapC11_D00 pf=/usr/sap/C11/SYS.....`

OSF/1

Details/idle mode

In *Details required* mode, the command lines for the processes are displayed, otherwise only the names of the executables are displayed.

Paging

All memory space managed by the virtual memory system is paged. This means that executable code (text, bss, data), memory, shared memory, and so on is only mapped on actual pages when necessary. If a maximum number of pages is reached (default is 128 pages), then the *Page stealer* process is started using inactive pages (pages which have not been accessed since a specified point in time).

The swapper starts swapping whole processes if the number of free pages gets too small.

Working set size (resident set size)

This gives details of all physical pages stored in the main memory at that point in time (including pages in the shared memory). This is a variation on the SAP definition.

Windows NT

`saposcol` on a Windows NT operating system is generally based on the same data as the Microsoft program `perfmon.exe`. Similarly, `saposcol` can only collect data available in `perfmon.exe`. You must activate the function which determines the disks performance, using the Windows NT command:

```
diskperf -y
```

All data is collected, regardless of the details or idle mode.

SAPOSCOL on a UNIX Host

Use

The `saposcol` program is normally installed with the R/3 System in the R/3 executables directory. The program requires “root” authorization since it reads areas of the operating system that are normally reserved for kernel processes.

Procedure

1. To confirm how `saposcol` is installed, enter the following command from the UNIX prompt in the R/3 executables directory: `ls -l saposcol.`

   This will display the following line: `-rwsr-xr-x 1 root system 779332 Aug 1 10:14 saposcol`

   For security reasons, it is important that the authorizations `-rwsr-xr-x` and the owner `root` are configured. The owner `root` authorizations can be reconfigured using the following commands (as `root` user): `chown root saposcol` and `chmod 4755 saposcol`

   Remember, `saposcol` should be started at or before the time you call up the first R/3 instance.

   If R/3 instances with different release levels will be running on a single host, the most current `saposcol` must be used. To ensure that the correct `saposcol` is started before each R/3 instance, you can modify every start profile so that the most recent `saposcol` version is called. Also, be sure that the host is started with the same time zone as the R/3 instances. Doing so will prevent discrepancies in the performance data and the time at which it was generated.

See also:

Starting SAPOSCOL [Seite 798]

Stopping SAPOSCOL [Seite 800]

Displaying Collector Data [Seite 787]

Installing SAPOSCOL and RFCOSCOL for a Dedicated DB Server [Seite 792]

Controlling SAPOSCOL from the Operating System

Use

You can start **saposcol** from the operating system command prompt with the following options:

-i <interval(sec)> | -u [<interval(sec)>] | -e [<interval(sec)>]

For all three of these options, SAPOSCOL must already be running. Alternatively, SAPOSCOL can be started using the -l option placed before any of these options.

For example, if SAPOSCOL is not already running, -l -i 20 -e 400 -u 90 will set the intervals successfully. Without the -l option, SAPOSCOL must already be running.

Option	Argument
-d	Dialog mode
-k	Kills the running collector
-r	Kills the running collector and print results
-l	Launches the collector
-f	Launches the collector in all cases
-s	Displays the status of the collector
-i	Sets a new interval for collecting data in normal mode (default is every 10 seconds)
-v	Displays the version of the collector
-u	Sets a new interval so that if no data is collected during that interval, the collector mode changes from normal to idle mode. The default interval is 300 seconds or 5 minutes. Example: SAPOSCOL – u 600 means that if after ten minutes there is no input to get data (either from a user or a process), the collector goes into idle mode.
-e	Sets a new interval for collecting data in idle mode (default is every 60 seconds)
-c	Cleans shared memory
-p	Puts shared memory to file
-g	Gets shared memory from file
-t	Sets debug trace level
-n	Sets normal trace level
-o	Prints all data
-m	Prints all snapshot data
-x	Tests background process running and answers

Dialog Mode

Running `saposcol` in dialog mode allows you to display from the OS prompt any data in shared memory. This data is collected by a different `saposcol` running in the background. You can run `saposcol` in dialog mode both from the command prompt and from the CCMS operating system monitor [Seite 750].

Procedure

1. To start `saposcol` in dialog mode from the command prompt, use the following command: `saposcol -d.`

   `saposcol` does not attempt to start a second collector. `saposcol` then displays a new command prompt: `Collector>`

2. You can enter several `saposcol` commands at this prompt. To display a list of possible commands, use the `help` command.

3. In dialog mode, you can control and monitor `saposcol` running in the background. To display the status of the `saposcol` that collects data in the background, use the command `status` or `stat` You can display the data that `saposcol` has collected directly from shared memory.

   To display OS performance data stored in shared memory, at the `Collector>` prompt, enter the command:

   `dump memory all`

   The following information is displayed:

```
Collector > dump memory all
Pages paged in / sec        1
Pages paged out / sec       0
KB    paged in / sec        4
KB    paged out / sec       0
freemem [KB]            13312
physmem [KB]            65536
swap configured   [KB]    76348
swap total size   [KB]    76348
swap free inside  [KB]    72556
```

   Unlike in the SAP System, data that is not available is **NOT** marked by N/A.

These are the most important combinations of the command `dump`:

`dump cpu single`	snapshot details for each CPU
`dump cpu all`	general snapshot details for all CPUs
`dump cpu sum`	mean values from data collected every hour for all CPUs

`dump memory all`	snapshot details of the memory
`dump memory sum`	mean values from data collected every hour for the whole memory

`dump top`	snapshot details of top cpu consuming processes

`dump disk single`	snapshot details of each individual disk
`dump disk sum`	mean values for every hour for every disk

`dump filesystem single`	snapshot details of each file system
`dump filesystem sum`	mean values for every hour for each file system

`dump lan single`	snapshot details of every network interface
`dump lan sum`	mean values for every hour for every network interface

`dump parameter configured`	operating system parameters configured
`dump parameter used`	operating system parameters currently being used

`dump hour`	Displays the following list, which shows the hour (0 to 23) and the day when the data was collected. The format is `year/month/day` with no separator. Data collected at a certain hour marked with the number 2 is inconsistent. Data marked with the number `1` is for the current hour. Data marked with the number 0 is unavailable.

```
-------------------------------------------------------
 hour:  0 of day 19950814    hour:  1 of day 19950814
 hour:  2 of day 19950814    hour:  3 of day 19950814
 hour:  4 of day        2    hour:  5 of day 19950814
 hour:  6 of day 19950814    hour:  7 of day 19950814
 hour:  8 of day 19950814    hour:  9 of day 19950814
 hour: 10 of day 19950814    hour: 11 of day 19950814
 hour: 12 of day        1    hour: 13 of day 19950813
 hour: 14 of day 19950813    hour: 15 of day 19950813
 hour: 16 of day 19950813    hour: 17 of day 19950813
 hour: 18 of day 19950813    hour: 19 of day 19950813
 hour: 20 of day        2    hour: 21 of day        0
 hour: 22 of day        2    hour: 23 of day 19950813
-------------------------------------------------------
```

Controlling SAPOSCOL from the Operating System

You can also control the background `saposcol` in dialog mode at the `Collector>` prompt using the following commands:

`detailson`	sets the details flag
`detailsoff`	reverses the details flag
`interval 5`	changes the collection interval to 5 seconds (default 10)
`kill`	stops the background process
`launch`	starts a new collector
`force`	forces a new collector to start (only in emergencies)

The changed values are written to the shared memory and `saposcol` reads them before more data is collected. For example, if the collection interval is changed from 10 to 2 seconds, it can take up to 8 seconds before the collector switches to the new interval.

You can also add comments in lines beginning with #.

4. To exit `saposcol`'s dialog mode, use the `quit` command.

See also:

Starting SAPOSCOL [Seite 798]

Stopping SAPOSCOL [Seite 800]

Reducing CPU Load [Seite 810]

Collector Home Directory, Paths and Profile Parameters [Seite 782]

Collector Status [Seite 784]

Displaying Collector Data [Seite 787]

Interpreting the Data [Seite 794]

Collector Home Directory Path & Profile Parameters

Definition

The collector home directory is the directory into which `saposcol` writes its data and looks for the `dev_coll` trace file and the shared memory backup file `coll.put`. This directory also contains configuration files from the RMON-MIB `saposcol` variant. By setting a profile parameter, you can use a different directory. In this case, `saposcol` must be started using a profile. For example:

```
saposcol -l pf=/usr/sap/C11/SYS/profile/saposcol.pfl
```

where the file `/usr/sap/C11/SYS/profile/saposcol.pfl` contains only the following lines:

```
DIR_PERF = /usr/sap/my_saposcol_dir
```

You can also access the `dev_coll` trace file so you can view it from the R/3 System. This profile parameter should also be maintained synchronously for the R/3 instances.

The profile parameter `exe/saposcol` (not a `saposcol` profile parameter itself) is also important for the R/3 instance. To start, stop or view the status of `saposcol` from R/3, the parameter must point to a correctly installed `saposcol`.

If several R/3 instances are running, for example, each with a different R/3 Release, you must ensure that each instance is using the same (latest) version of `saposcol`. For example, if the XYZ System is a more recent release than C11, and both systems are running an instance on one host, then set the instance profile parameters in both systems as follows:

```
exe/saposcol = /usr/sap/XYZ/SYS/exe/run/saposcol
```

This ensures that each instance on this one host uses the same `saposcol`. The Operating System Monitor [Seite 750] status displays whether the current `saposcol` is the one which you started.

See also:

Collector Status [Seite 784]

Displaying the Collector Status

Use

You can display the collector status using the operating system monitor [Seite 750].

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Performance menu* → *Operating system* → *Local* → *Activity* → *OS Collector* → (equivalent to the command saposcol -s). Then choose *Details on*.

Result

The system displays the following information:

```
Status report --------------------------------------------------------

Versions of collectors

running                 COLL 20.30 95/07/05 - @(#)hpuxcoll 20.19 HP
95/04

dialog                  COLL 20.30 95/07/05 - @(#)hpuxcoll 20.19 HP
95/04

Shared Memory           attached

Number of records       1958

Active Flag             active (01)

operating system        HP-UX hp01 A.09.00 E

                        9000/817 1431332232

collector PID           29270 (00007256)

Collector               running

starttime coll.         Mon Aug 14 15:10:01 1998

current time            Mon Aug 14 15:10:35 1998

Last write access       Mon Aug 14 15:10:26 1998

Last read  access       Mon Aug 14 15:10:24 1998

collecting intervall    10 sec (next delay).

collecting intervall    14 sec (last).

status                  free

details coll.mode       required

refresh                 required


Header extension structure
```

```
 number of x-header      records        1
 number of communication records        40
 number of free com.     records        40
 resulting offset to 1.data rec.        41
Trace level                             2
Collector in IDLE - mode ?              NO
  become idle after 300 sec without read access.
  length of idle interval               60 sec
  length of norm.interval               10 sec
**************************************************************
```

The most important details displayed here are:

- The time `saposcol` was started (`starttime coll.`)
- The collector's PID, whether the collector is active (`Active Flag  active(01)`) or whether the operating system has recognized that this collector is available (`Collector running`)
- When the collector last wrote data into the shared memory (`Last write access`)
- The current setting of the details flag (`details coll.mode required`)
- The length of the intervals

One of these versions is the collector that runs in the background, and the other is the collector that prepared the status (dialogs). If these vary from each other, then an old collector may be running.

Displaying Collector Data

Use

You can use the operating system monitor [Seite 750] to display data in the shared memory segment that was collected by the `saposcol` program.

Prerequisites

`saposcol` must be running so that the data is available.

Possible Error Messages

Shared memory not available	`saposcol` has not created a shared memory segment. This is usually because the `saposcol` program has not been started.
Collector not running	`saposcol` was started and created a shared memory segment, but was later terminated.

Data that cannot be collected by `saposcol` on this platform is labeled N/A ('not available').

Procedure

1. To call the operating system monitor choose *CCMS* → *Control/Monitoring* → *Performance menu* → *Operating system* → *Local* **or** *Global* → *Activity*.

   To update the screen with the most recent data from the shared memory, choose *Refresh*. Note that `saposcol` collects data at 10 second intervals (default). You do not always obtain new data with refresh.

See also:

Controlling SAPOSCOL from the R/3 System [Seite 806]

Setting Up RFC Communication to RFCOSCOL

Purpose

You must set up RFC communication when you use a dedicated database server.

Process Flow

This section provides examples of the steps that are normally required to set up RFC communication. The procedure described may, however, not apply to all scenarios.

There are three main steps. Step 1 is carried out in the operating system of the system you are working in. Steps 2 and 3 are carried out in the R/3 System.

1. Set up a remote shell access

   R/3 uses a remote shell to call `rfcoscol` via the gateway. Note that the gateway runs on a different host than `rfcoscol`. You must maintain the `rhosts` file so that the gateway can start a program on the target host without a password (you may have to create this user on the target host first). The gateway either calls the remote shell so that the user name is determined by the gateway user ID, or, if the environment variable “USER” has been preset, then this name is used.

   The user `c11adm` starts the gateway on the host `r3c11000`, and the environment variable is not preset. When an R/3 work process needs the gateway to start the program `/usr/sap/bin/rfcoscol` using TCP/IP on the host `c11dbsrv`, then the gateway executes the following command:

   ```
   remsh c11dbsrv /usr/sap/bin/rfcoscol.....
   ```

   This is only possible if the `rhosts` file in the home directory of user `c11adm` on host `c11dbsrv` contains the following line:

   ```
   r3c11000 c11adm
   ```

   Using the same names as above, but where the environment variable USER has been preset as `osreader`, the gateway executes the following command:

   ```
   remsh c11dbsrv -l osreader /usr/sap/bin/rfcoscol
   ```

   Here, the `rhosts` file in the home directory of the user `osreader` on the host `c11dbsrv` must contain the following lines:

   ```
   r3c11000 osreader
   ```

2. Set up a logical destination

   Create a destination with the following attributes using the RFC destination maintenance tool (Transaction SM59):

Name:	`SAPOSCOL_<HOSTNAME>`, where `<HOSTNAME>` is replaced by the target host name
Type:	T for TCP/IP
Area:	Explicit
Target host:	Target host name (`c11dbsrv` in the above example)
Program:	`/usr/sap/bin/rfcoscol`

Call Transaction AL15. From the menu, choose *SAPOSCOL destinations*. Choose *New RFC destinations*. This calls the RFC destination maintenance tool (Transaction SM59). From here, you can test the connection to `rfcoscol`.

3. Save this logical destination as a remote `saposcol`.

   You can use Transaction AL15 to manage existing logical destinations that call up `rfcoscol`. Specify a logical destination and then save it as a `saposcol` destination using *Add Destination*. This logical destination can then be used under *saposcol remote*.

   The field *Database Server* specifies whether this destination leads to the R/3 System's database server. The remote `saposcol` can be used for both dedicated database servers and other non-R/3 System servers.

Installing SAPOSCOL/RFCOSCOL for a Dedicated DB Server

Use

The `saposcol` and `rfcoscol` programs are normally installed with the R/3 System. If you are using a **dedicated database server**, you may need to install the programs separately.

Procedure

1. To install the `saposcol` and `rfcoscol` programs in the operating system, copy the programs `saposcol` and `rfcoscol` into the directory `/usr/sap/bin` on the host that you want to monitor remotely. Ensure that you have the correct versions for the operating system to be monitored.
2. As user `root`, change the executable files' authorizations as follows:

   ```
   chmod 4755 saposcol
   chown root saposcol
   chmod 4755 rfcoscol
   chown root rfcoscol
   ```
3. Enter `saposcol` to start the program.

   `saposcol` creates a shared memory segment and periodically writes the performance data to it. You should also start `saposcol` every time you start the operating system so that the performance data is always available. This can be done automatically by entering it into the appropriate shell scripts in the `rc` directory or file. Make sure that the time zone is set correctly and that the performance data is summarized hourly. The installation is now complete.
4. Install the program `rfcoscol` on the remote host.

See also:

Setting Up RFC Communication to RFCOSCOL [Seite 789]

Documentation for the “Gateway” and “RFC for External Programs” that you are using.

Interpreting the Data

Definition

If the monitors display N/A, the data is not available on that platform. Earlier R/3 Releases (up to Release 2.2D) often display a **0** or **-1** instead of N/A to show that the data is invalid.

CPU (process management)

For each CPU, the following data is displayed:

- User
- System
- Idle

These values show how much CPU is being used as it is being conveyed to the UNIX or `vmstat`. Some operating systems also display a wait value. From R/3 Release 3.0C, the wait values will be separate from the idle value. You can determine from this data whether any CPU resources are still available.

A process could loop if a CPU user is permanently busy.

Further values include:

- Number of CPUs
- Load average over 1 min, over 5 min, over 15 min
- Interrupts per second
- System calls per second
- Context switches per second

Load average is the average number of processes per CPU which have been waiting to be assigned to a free CPU for over 1, 5 or 15 minutes. If the average is 1 process for each free CPU, then no problem has occurred. However, if the average is more than 3, there is a bottleneck in the CPU resources.

Top CPU Consumers

- Process ID
- Process owner
- Command
- CPU utilization of the process
- Used CPU time in minutes and seconds
- Resident size of process in KB
- Priority of the process

Memory

- Main memory
- KB of free memory
- Physical configured memory kilobyte

Paging

- Number of pages paged in per second
- Number of pages paged out per second
- KB paged in per second
- KB paged out per second

For most operating systems, the page out rates are more important than Page in rates. A Page in also takes place when a program is started, and does not necessarily lead to performance problems. However, if the actual Virtual Memory Management must store a page in the swap space several times (Pages out), then there is a memory bottleneck which significantly reduces performance. Which Pages out rates are critical depends on the current operating system.

Swap space

- Configured in KB
- Free in KB
- Actual size of total swap in KB
- Maximum swap size in KB

The most important details displayed in the *Swap* section are the *Free in swap space* and the *Actual swap space*.

You can provide the virtual memory management with swap space in various ways, depending on the operating system's configuration. Firstly, whole partitions or disks can be reserved for the swap, and this space is then marked as *Configured*. Swap space can also be a dynamically increasing file in a file system, and its values are displayed in *Current*. As the size of the file can increase, remember the limits of the swap space. All types of swap space are not available on all operating systems. Therefore, in some cases the actual size of the swap space must correspond to the configured and maximum swap space.

Disk

- Device file name of the disk
- Disk utilization in percent of the time the disk is busy
- Average queue length of request for an I/O
- Wait time in milliseconds in which a request remains in the queue
- Service time in milliseconds for an I/O
- Kilobytes transferred per second
- Number of disk operations per second

The *average response time* is also displayed in the Operating System Monitor. This is the sum of the *wait time* and the *service time*. Some of the values are redundant, for example, the *average wait time* cannot be calculated from the number of operations and average queue length. Note that all values have been included here, even though they are not all available on some platforms.

File System

- Name of the file system
- Capacity
- Free

LAN

- Name of the LAN interface
- In packets per second
- Out packets per second
- Errors for receiving data packets per second
- Errors for sending data packets per second
- Collisions per second

  Some values are not given in some network interfaces, for example, there may not be any 'collisions per second' in a token ring architecture. The values given here do not describe the actual network traffic. It describes the transfers made via this interface, for example, the errors displayed here are to do with the interface and not the actual network segment.

- Data summarized hourly

  You can also view the mean values of data collected by the hour during the last 24 hours. The values under 'Recent hour' are taken directly from the shared memory. `saposcol` automatically determines whether or not the data is complete, and does not display incomplete data.

  In addition, the background job `SAP_COLLECTOR_FOR_PERFORMANCE` reads the data from the shared memory, and then stores it in the performance database `MONI`.

See also:

Special Features of Operating Systems [Seite 772]

Starting SAPOSCOL

Purpose

You start the `saposcol` program if you want to begin collecting data on your operating system resources.

Process Flow

Start `saposcol` no later than when you start the first R/3 instance on a host. A new `saposcol` will not start if another `saposcol` is already running.

1. To start `saposcol`, use either the Operating System Monitor *OS Collector* option or, at the command line, use the command `saposcol`.

   Normally, `saposcol` is called without any additional parameters or profiles. This call is equivalent to `saposcol -l` (where `l` stands for *Launch*). For more information, see Controlling SAPOSCOL from the Operating System [Seite 777].

2. If this call finds a `saposcol` shared memory segment, it takes on the PID of a `saposcol`, which may already be running, from data stored in shared memory.
3. If no other `saposcol` is collecting data, this PID equals 0 and `saposcol` will start, using the existing shared memory segment. The `saposcol` program will also start if there is no shared memory available.
4. However, if the newly called `saposcol` recognizes that another `saposcol` is already collecting data, it will not start.
5. Once `saposcol` has started, it performs an initialization, setting aside the required amount of shared memory based on such things as the number of available disks, CPUs, and file systems.
6. The program then executes a fork, after which the child process runs in the background and the original program ends.

Error and status messages can be found in the `dev_coll` file in the directory `/usr/sap/tmp`. When it starts, `saposcol` reads the data in the file `coll.put`. You can change the default path by changing a profile parameter in `sapdefault.pfl`.

As `saposcol` collects data constantly, even when no R/3 components are running, you should not normally stop `saposcol`.

See also:

Stopping SAPOSCOL [Seite 800]

Operating System Monitor [Seite 750]

Stopping SAPOSCOL

Purpose

Stop `saposcol` only if you want to restart the host or if you want to swap to a different version of `saposcol` itself.

Because `saposcol` collects data constantly, even when no R/3 components are running, you should not normally stop `saposcol`.

Process Flow

7. Stop `saposcol` with the command `saposcol -k`.

   A new `saposcol` is started which stops the active `saposcol`. The process that is collecting data should then stop working after one second.

8. The new `saposcol` connects to the shared memory.
9. From the shared memory, it determines the PID of the `saposcol` that is collecting data, and the status of the active flag.
10. If the new `saposcol` finds a valid PID, then a stop flag is set in the shared memory. When the old `saposcol` finds this flag, it resets the active flag, and deletes the PID from the shared memory. If this does not happen within one second, the new `saposcol` sends a signal to the background process to stop working.
11. The new `saposcol` checks this several times against the active flag in the shared memory.

    If necessary, the process clears the data in the shared memory itself.

12. Before the old `saposcol` stops, it writes the data in the shared memory to the file `coll.put` in the home directory of `saposcol`.

The file `coll.put` is imported when the host is restarted, so that the summarized data in the shared memory is available to R/3. For example, if `saposcol` is stopped at 12:03 and restarted at 14:49, then the data from hours before 12 is still available to the R/3 System. To avoid confusion, invalid data for hours 12 to 14 is not displayed in the recent hours overview in the operating system monitor.

The time for which a shared memory segment exists depends on the operating system. On a UNIX operating system, it is retained until `saposcol` deletes it. On Windows NT, the shared memory is deleted by the operating system if there is no process connected to it.

Do not stop `saposcol` with other operating system commands. Doing so corrupts the data in the shared memory segment. You would also not be able to start a new `saposcol` and would be notified that another `saposcol` is already running.

See also:

Starting SAPOSCOL [Seite 798]

Controlling SAPOSCOL from the R/3 System [Seite 806]
Displaying Collector Data [Seite 787]

Troubleshooting: Operating System Collector

Purpose

No values displayed

There are various ways to check your R/3 installation if:

- the operating system monitor does not display any values
- the system displays messages such as *Shared memory not available* or *Collector not running*
- the values displayed do not change.

Process Flow

1. The quickest way is to log on to the R/3 System as the system administrator and view the `saposcol` status to verify that it is correctly installed and running properly.
2. Check the entry for the last write access. This entry should not be much greater than the length of the idle interval in seconds or than the current time.
3. To test further, call `saposcol` in dialog mode with the command `saposcol -x`
4. The following message should then be displayed `OK: actual time of background saposcol [Mon Aug 14 12:20:17 1995].`
5. A request is sent from the foreground program via the shared memory to the background collector, asking it to give its current time. If an error has occurred, the following message appears: `ERROR: background saposcol do not answer for 15 sec.`
6. If you get this error, check your installation and start the correctly installed `saposcol`. You do **not** have to restart the R/3 instances.

Result

If the test was successful, but you are still not receiving any new data in the Operating System Monitor [Seite 750], it generally means that the work processes cannot access the shared memory. This is particularly common on AIX systems that have incorrect instance profiles.

See also:

R/3 Notes on `saposcol` in SAPNet. R/3 Note 5076, in particular, is important for AIX.

Incorrect or Implausible Data

In certain circumstances some values are incorrect even if you have carried out a thorough function check. Possible reasons for this are:

- The message *N/A* instead of a value means that the collector cannot collect the data. Older `saposcol` versions often use **0** or negative values instead of the message *N/A*.
- On some platforms, the collector itself determines the load average. The mean values are greater than 1, 5, or 15 minutes, and `saposcol` must already be running during this

time. If, for example, a `saposcol` has not been running for the whole 15 minutes, this will result in a negative value to avoid misinterpretation.

Process Flow

1. Check the `saposcol` home directory for messages in the `dev_coll` file that indicate the cause of possible problems.
2. Check whether the values were listed incorrectly in R/3, or if they are incorrect in the shared memory. To do this, compare the values listed in R/3 with the values from the shared memory by using the `dump` command.

Controlling SAPOSCOL from the R/3 System

Purpose

You can use the R/3 System to control and monitor saposcol with the CCMS operating system monitor.

From the R/3 initial screen, choose *Tools → CCMS, Control/Monitoring → Performance Menu, Operating system → Local → Activity → OS Collector*.

From the OS Monitor, you can:

- start or stop `saposcol`
- display the `saposcol` trace file (default: `/usr/sap/tmp/dev_coll`)
- view the current status of `saposcol`
- switch *details mode* on and off

The profile parameters must be set correctly. Otherwise the data displayed will not be meaningful.

Process Flow

Detail analysis menu

In the OS Monitor main menu, choose snapshot values from the last 24 hours and access the data on the performance database.

Available functions include:

- Analyze the operating system
- Performance Database
- Additional Functions

The *Detail analysis menu* offers the following options:

Analyze the operating system

Snapshot analysis / Previous hours

Use the options in this area to display data from the shared memory on the CPU, memory, swap space, disk, LAN and file system. Analysis possibilities available include:

- Snapshot
- Data of the previous hours

  If there is no valid data available for a particular hour, that hour is not displayed.

Operating system log

This function lists part of the operating system log. The request is sent to `saposcol`, which transfers part of the operating system log to the file `/usr/sap/tmp/os_sys.log`. A response to this request is returned to the ABAP program that called it.

Performance database

Compare recent days / Compare all servers

The background job `SAP_COLLECTOR_FOR_PERFORMANCE` regularly collects data from `saposcol`'s shared memory and stores it in a performance database. You can use this data to compare a server's load over the last 30 days with the load of other servers. Only the most important data is actually stored in the database. Calculations based on that data are done on the fly. Mean values from the hour are summarized again and averaged out for the time when the load is greatest (10 AM to 1 PM, and 2 PM to 4 PM).

Additional functions

System Configuration

Defined values for operating system kernel parameters are displayed as they appear in the configuration files together with the values that are currently active. These active values will differ from the predefined values if the system was booted using a different kernel parameter.

Parameter Changes

Here, you can view all changes made to the system parameters.

LAN Check by Ping

A `ping` operation over a range of servers gives you the following information:

- minimum packet travel time
- maximum packet travel time
- average packet travel time
- number of lost `ping` packets

With a multiple `ping` operation, you can also obtain results for every packet that was sent.

The R/3 `ping` utility is an enhancement of the standard UNIX `ping` command. The R/3 `ping` utility determines the time taken by a network packet to travel between the home and a target system. This time is a good indicator of network traffic efficiency. It takes into account the physical network distance and the type of the network technology used.

The utility reports the number of presentation servers, application servers, and database servers that belong to the R/3 System. The administrator can select the servers of interest and execute the `ping` operation. It is possible to perform a `ping` operation on the target systems once or multiple times, using different `ping` packet sizes.

See also:

Controlling SAPOSCOL from the Operating System [Seite 777]

Reducing CPU Load [Seite 810]

Collector Home Directory, Paths and Profile Parameters [Seite 782]

Collector Status [Seite 784]

Controlling SAPOSCOL from the R/3 System

Displaying Collector Data [Seite 787]

Interpreting the Data [Seite 794]

Operating System Monitor [Seite 750]

Reducing CPU Load

Use

`saposcol` may use a large amount of the operating systems resources because it periodically collects data on the operating system. Which data uses up the most resources varies from one operating system to another.

Procedure

There are two ways to reduce the CPU load.

- Choose *Details Off*.

  Depending on the operating system, some data is not always collected during every cycle. The default is *Details required*. You can change this setting from the OS Monitor.

- Use idle mode.

  When `saposcol` is in idle mode, it only collects data every minute.

  If a process does not read data from the shared memory, the data does not need to be collected every 10 seconds. To get an average value for every hour, it is sufficient for data to be collected every minute. `saposcol` switches to idle mode if the data has not been read for 5 minutes. If a process reads data from the shared memory while the collector is in idle mode, then the `saposcol` switches back to normal mode and collects the data every 10 seconds.

`saposcol` normally requires up to 2% of memory when it is collecting data in the background. In idle mode, the CPU time requirement is less than 0.5%, and data is collected frequently enough to supply good statistics for each hour.

See also:

Interpreting the Data [Seite 794]

Controlling SAPOSCOL from the Operating System [Seite 777]

Controlling SAPOSCOL from the R/3 System [Seite 806]

The Alert Monitor

The CCMS (Computing Center Management System) provides an alert monitor for helping you to monitor and run your R/3 System. The alert monitor uses the advanced object-based technology of the CCMS monitoring architecture.

With the alert monitor, you can manage and monitor your R/3 System efficiently. The monitor offers you:

- Complete, detailed monitoring of the R/3 System, host systems, and database
- Status indicators (green, yellow, red) for all components
- Alerts if a status indicator is not in the green range
- Easy access to methods for analyzing alerts
- Alert tracking and management

This section describes how to use the alert monitor.

The Release 4.0 alert monitor has replaced the previous monitoring and alert system in the CCMS.

The new monitor offers all of the functionality that was available in the old alert monitor as well as new, more reliable alerts and more advanced and powerful features.

See also:

Using the Alert Monitor: Tutorial [Seite 827]

Using the Alert Monitor: Detailed Tutorial [Seite 829]

The Monitoring Architecture: Concept [Seite 814]

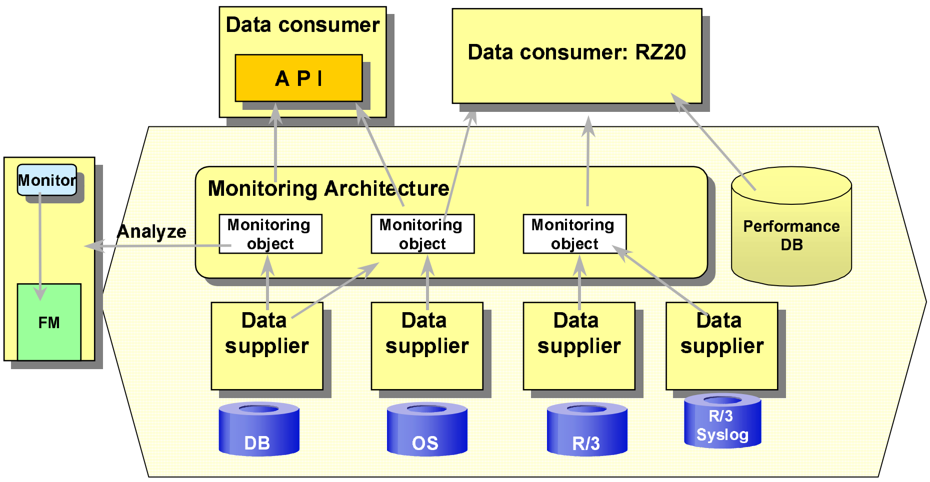
Monitoring: How-To Instructions [Seite 829]

Creating Your Own Monitors [Seite 884]

The Monitoring Architecture: Concept

Purpose

The R/3 alert monitor is based on an innovative new monitoring architecture that was introduced with Release 4.0. This section describes the main parts of the monitoring architecture and how they work together.



The Monitoring Architecture.

Here is how the monitoring architecture and the alert monitor work:

Collection methods: These are programs that gather information on different parts of the R/3 System and its environment. They then register this information with the monitoring infrastructure.

The monitoring architecture comes ready to use with collection methods for all of the most important components of your R/3 System and environment. When you start the alert monitor, you will see that collection methods are already active for reporting on all of the following:

- The host systems on which your R/3 System is running
- The R/3 database
- R/3 instances (application servers) and their services and components
- Components outside the R/3 System

There is nothing that you have to do to prepare or activate the monitoring architecture. All collection methods in your system are started automatically when the system starts or as they are needed.

Monitoring architecture: Collection methods pass their information to the monitoring architecture. The monitoring architecture provides an infrastructure for gathering and managing system information.

The infrastructure concept means that collection methods are independent of the monitoring architecture. Collection methods "plug into" the monitoring architecture and use its services for displaying and managing system information. But collection methods are not programmed directly into the monitoring architecture.

Collection methods are independent of the monitoring architecture, and so can be easily added or modified. The monitoring architecture offers ABAP and C interfaces which any R/3 or external programmer can use to create a collection method.

Monitoring objects and attributes: Collection methods report their information with respect to one or more monitoring objects (as shown by the arrows in the illustration). A monitoring object represents something in the R/3 System or its environment that should be monitored. A monitoring attribute represents one type of information that is to be reported on a monitoring object.

Example: Monitoring objects include the CPU in your host system, the database, and R/3 services, such as background processing. Monitoring attributes for a CPU object might include the five-minute average CPU load and the CPU utilization.

Monitoring objects and their attributes are displayed to you in the alert monitor. In the alert monitor, objects and their attributes are called nodes or monitoring tree elements (MTEs).

Monitoring objects are created by collection methods. All the objects that you can currently monitor in your R/3 System are automatically available when you start the alert monitor.

Data consumers: This is the layer of the monitoring architecture for displaying alerts and status data. Data consumers are supplied with the information that collection methods pass to the monitoring architecture. SAP delivers a standard "data consumer" (the alert monitor) with your R/3 System. SAP also provides other, more specialized monitors that use the data provided by the monitoring architecture.

The data consumer layer (the alert monitor) is your workplace for displaying the current state of your system and responding to alerts that are triggered by warnings or problems. The alert monitor displays its information and alerts to you in a hierarchical monitoring tree. See also a schematic view of the monitoring tree [Seite 860].

If a collection method reports a problem, then the monitoring architecture triggers an alert [Seite 833]. This is also visible to you in the alert monitor. For example, if the CPU's five-minute load average is too high, then the alert monitor triggers a "yellow" (warning) or "red" (problem) alert. The color-coding in the alert monitor lets you see an alert fast. And the alert monitor can then provide you with detailed information on the alert as well as access to a method for analyzing the alert.

The alert monitor also provides the management methods that you need to monitor the system. You can adjust alert thresholds. You can also add or customize auto-reaction and analysis methods. Auto-rection methods respond automatically when an alert is triggered. An analysis method lets you investigate the cause of an alert without leaving the alert monitor.

The monitoring architecture also offers an interface so that external "data consumers" (external programs) can receive the monitoring architecture's information.

What is Predefined in the Alert Monitor?

Use

The alert monitor is designed to be ready to use the first time that you start your R/3 System. There should be no need to do any extensive Customizing or preparation of the monitor before you start using it.

Exception: If you want to monitor multiple R/3 Systems, you will have to set this up yourself. The alert monitor cannot be predefined for monitoring multiple R/3 Systems [Seite 848].

The following is standard and predefined in the alert monitor:

- **Monitoring functionality.** All monitoring functionality that is available in this Release of the R/3 System is automatically available in the alert monitor. You do not have to do anything to ensure that all available monitoring functionality is active in the alert monitor.

  When you start the R/3 System, the basic set of monitoring objects and attributes (monitoring functionality) is automatically added to the alert monitor. This basic set includes monitoring functionality for critical R/3 functions, for your host servers, and for the R/3 database system. Additional basic functionality will be added to the alert monitor with each R/3 Release.

  R/3 applications and other R/3 components dynamically add their own monitoring objects and attributes to the alert monitor as they are used.

- **Method definitions, method assignments, and MTE class assignments.** All of the methods (data collection methods, analysis methods, and automatic reaction methods) that are needed for standard monitoring functionality are predefined in the alert monitor. You can display the definitions of these methods using Transaction RZ21.

  Method assignments to monitoring tree elements are also predefined. For every MTE, the best available data collection and analysis methods are assigned. Where available, automatic reaction methods are also predefined.

  Further, monitoring tree elements have been assigned to MTE classes according to a best-practices model for alert monitor Customizing. An MTE class represents the general properties and method assignments shared by a particular group of monitoring tree elements. The MTE class is also used in rule-based monitor definitions.

  For example, the *Space Management* monitoring object and the *Free Space* monitoring attribute share the *CCMS_DB_Freespace_MT* MTE class. This means that the two MTEs share general properties and method assignments.

- **Alert thresholds and attribute group assignments.**

  "Best practices" values for alert thresholds and severity weightings have been predefined for all monitoring attributes. These settings are based on SAP's wide experience with production R/3 Systems.

  SAP has also predefined the attribute groups to which monitoring attributes are assigned. An attribute group represents the alert thresholds shared by a particular type of monitoring attribute.

  For example, all instance-specific exemplars of the *Response Time* monitoring attribute share by default a single attribute group. That means that by default the same response time thresholds are in effect in all instances of an R/3 System.

- **Monitor set and monitor definitions.** SAP has predefined two default monitor sets, the *SAP CCMS Monitor Templates* and the *SAP CCMS Technical Expert Monitors*. The monitors in the *SAP CCMS Monitor Templates* set show the monitoring objects and attributes that are essential for monitoring the R/3 System and are used for normal system administration.

  You use the monitors in the *SAP CCMS Technical Expert Monitors* for problem analysis and for monitoring the monitoring architecture itself. For example, you could use the *CCMS Selfmonitoring* monitor for internal monitoring, or *All Monitoring Contexts* to see the available contexts in your R/3 System, that is, what you can include in your monitor.

  The monitors in the *SAP CCMS Monitor Templates* are set to the overview level of detail and the monitors in the *SAP CCMS Technical Expert Monitors* are set to the expert analysis level. However, the alert monitor automatically shows any MTEs that have alerts, no matter what level of detail [Seite 842]they are normally assigned to.

See also:

Working with the Tree [Seite 820]

Working with the Alert Monitor Tree

Use

You can use the alert monitor to edit and select MTEs and choose different display options. Regarding the selection of MTEs, the operations can be divided into two groups:

- Operations that are simultaneously valid for several MTEs without restriction (displaying details, for example)
- Operations that can only be executed meaningfully if the nodes have been selected in a certain way.

You can only access threshold Customizing once you have selected a node.

Procedures

To select MTEs in the tree:

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor*.
3. Choose *Edit* → *Selections* → *Select* → *Node MTE* or *Sub-tree* or one of the other available options.

To display an overview of the selected MTEs in the tree:

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor*.
3. Choose *Edit* → *Selections* → *Display selections.*

To expand a subtree:

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor*.
3. Select the subtree and choose *Edit* → *Tree* → *Expand tree* → *Expand subtree*.

Tree Components

Use

The monitoring architecture displays a tree with all monitor tree elements of the R/3 System. These monitoring tree elements (MTEs) help visualization when creating a new tree. The elements are uniquely structured and ordered within the R/3 System in the tree.

MTEs are ordered under a summary node. They are logically combined in groups, which are called the monitoring context. The monitoring context is named after the root element of the monitoring tree.

There are three types of MTE:

- Monitoring summary MTEs are labels in the alert monitor that do not have any monitoring functionality. No alerts can be triggered for summary MTEs. However, alert statuses and messages can be displayed in these nodes.

  Virtual nodes let the programmers of monitoring functionality structure the alert monitor. For example, *Application Server* and *R/3 Services* are standard labels (virtual nodes) in the alert monitor.

  There are two kinds of monitoring summary node:

  – Real: saved in the monitoring segment

  – Virtual: are not saved in the monitoring segment, and merely serve to visually group of real monitoring tree elements in a monitor

- Monitoring objects represent particular components of your R/3 System or its environment that can be monitored.

  Monitoring objects tie together different monitoring attributes that pertain to the same component or aspect of the R/3 System.

  For example, *SpaceManagement* and *CPU* are standard monitoring objects in the alert monitor.

- Monitoring attributes represent types of data that can be reported for a particular monitoring object. The monitoring object *CPU*, for example, has the attributes *CPU utilization* and *5MinuteLoadAverage*. The R/3 System reports data for these attributes, and the alert monitor triggers alerts with respect to these attributes, should the data violate the defined alert thresholds.

  There are various types of monitoring attribute:

  – Performance attribute

  – Status attribute

  – Heartbeat attribute

  – Log attribute

  Each of these attributes has specific characteristics, however they all have one thing in common: alerts can be triggered from the values reported for an attribute.

  Text attributes are another type of monitoring attribute. They cannot trigger alerts, and serve only to provide an exact description of a monitoring tree element.

Several monitoring attributes are grouped under one monitoring object.

A monitoring tree element is uniquely indicated within the monitoring context by means of the following:

- Number range.

  Unique number within the monitoring architecture. A logically related group of monitoring tree elements is identified with the help of the UID (Unique Identifier).

  Temporary number ranges are automatically generated by the system. With all others the uniqueness of the ID has to be guaranteed.

- Unique Identifier (UID)

  Unique number within the number range. Identifies an object within the number range (monitoring summary node, monitoring object, and monitoring attribute, for example).

See also:

Working with Log Attributes [Seite 825]

Working with Log Attributes

Use

Log attributes are monitoring tree elements (MTEs) that capture messages in context. This means that the collection method does not just write stand-alone messages into the alert monitor, rather the contents of a log or trace.

Procedures

Log attributes in the detailed data display have some special features. These are described here.

Displaying a log attribute:

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor*.
3. Mark one or more log attribute MTEs or alerts (choose *Legend* to display a list of the icons used) and choose *Display details*. The system displays the log attribute(s) for the MTE(s) that you selected.

Adjust the time period for the display:

The time period for the messages that are displayed in a log attribute is set independently of any alert or message shown in the alert monitor. If the log attribute is not showing the time period you need, then you can adjust the time period.

4. To set the time period for the log attribute, choose *Time interval*. You can either have the most recent <*nnn*> minutes displayed, or you can display entries that were created during a specific time period.

Merging log attributes:

If you are displaying detailed information on more than one log attribute, you can combine the contents of the log attributes.

5. To merge log attributes, find the table marked *Log attribute (overview)*. Mark the log attributes that should be merged together and choose *Merge*.
6. The system refreshes the display with the merged log attributes displayed in a single table. The messages are sorted by time stamp. You can resort them by marking the column heading to use for sorting and choosing either *Sort in ascending order* or *Sort in descending order.*
7. To display the log attributes separately again, choose *Individual display*.

Tutorial

Purpose

This section provides an overview of using the alert monitor. For more information, see the detailed tutorial for getting started with the alert monitor.

Process Flow

Assume that you want to make a periodic check of your R/3 System.

1. First, check the current condition of your R/3 System.

   Start the alert monitor by choosing *Administration* → *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.

   Open a monitor from the monitor sets tree and display the *Current system status* view to see whether everything is green.

   The alert monitor is ready to go when you install your R/3 System. There are no configuration or activation prerequisites that you have to fulfill before you use the monitor.

2. Second, check to see what has happened in the system since you last looked. Use the *Open alerts* view in the monitor to see whether any yellow (warning) or red (problem) alerts have occurred.

3. Thirdly, if there are open (unprocessed) alerts, you should resolve them. Switch to the *Open alerts* view and double-click on any red or yellow element in the monitoring tree (MTE) to open the alert browser and analyze the alerts.

   Once you have analyzed an alert and taken any required action, set the alert to *Complete* to remove it from the open alerts display.

See also:

Detailed Tutorial [Seite 829]

Customizing the Alert Monitor [Seite 882]

Creating Your Own Monitors [Seite 884]

Detailed Tutorial

Use

The following section gives you a detailed overview of using the alert monitor.

Procedure

1. **Choose *CCMS* → *Control/Monitoring* → *Alert monitor.*** Alternatively, call Transaction RZ20.

The system displays the *CCMS Monitor Sets*.

2. **Expand the monitor sets by positioning the cursor on the *CCMS Monitor sets* and choosing *Edit* → *Expand tree.***

SAP provides default monitor sets, the *SAP CCMS Monitor Templates* and the *SAP CCMS Technical Expert Monitors.* These monitor sets are always available to you.

3. **Display the *Entire System* monitor in the *SAP CCMS Monitor Templates* monitors by positioning the cursor on it and choosing *Monitor* → *Start monitor*.**

The monitor displays the tree as it was last used. The monitoring tree is a hierarchical display of the monitoring objects (system components) and monitoring attributes (types of information on objects) in the system. The *Entire System* monitoring tree shows all objects and attributes that are visible at the *Expert* level or for which there are alerts.

For the purposes of this tutorial, expand the tree if the whole hierarchy is not displayed. Position the cursor on the *Entire system* line, and choose *Edit* → *Tree* → *Expand Tree* → *Expand subtree*.

Click here for a schematic illustration of the monitoring tree [Seite 860].

In addition to the *Entire System* monitor, there are other predefined monitors for special purposes. For example, if you are the database administrator, you may wish to open the *Database* monitor instead of *Entire System*.

One special monitor is the *CCMS Selfmonitoring* in the *SAP CCMS Technical Expert Monitors.* This monitor displays any problems with the alert monitor and the monitoring architecture itself. Check in this monitor to ensure that all data collection methods that the alert monitor starts are running normally.

4. **Check the current state of your R/3 System**.

a) **Set the monitor to show the current system status.**

Display the *Current system status* if this is not displayed already.

The *Current system status* reflects the most recent performance values and status messages reported to the alert monitor. Older alerts that are still open (that is, not processed) are not reflected in the color-coding.

b) **Check the color-coding in the monitoring tree.**

The colors of the nodes, or MTEs in the tree have the following meaning:

Green: The component is running normally. Everything is ok.

Yellow: A warning or "yellow-alert" has been issued.

Red: A problem or critical condition has been reported, a "red alert."

Gray: No data is being supplied for an MTE. (Check the *Self-monitoring* monitor to see why the collection method for this MTE is not available.)

To display a legend of the colors and icons used in the alert monitor, choose *Extras → Legend*.

The alert monitor propagates the highest alert level up the monitoring tree. For example, if the MTE with the name of your R/3 System is green, that means that all components in the R/3 System's monitoring tree have the "green" status. There is no problem with the system.

Double click on any MTE to start the analysis method [Seite 896] associated with the MTE. The analysis method displays more detailed information on the current status of the MTE.

Optionally, you can choose to automatically refresh the display. To do this, choose *Extras → Display options* and switch to the *General* tab. In the *Refresh display* box, mark Y*es, interval* and enter the refresh interval. The suggested value is 300 seconds or longer. If the automatic refresh is deactivated, the alert monitor displays the data that was available when you started it.

5. **Check to see what has happened recently in the system by choosing *Open alerts*.**

   Instead of showing the current status of the system, the color coding now shows where open alerts exist. Open alerts are those that have not yet been analyzed.

   If you are just coming in to work or returning from lunch, you can use the *Open alerts* view to see if anything happened in the system while you were away. The monitor saves alerts for you to review even if the condition that caused the alert has since improved.

6. **Respond to an alert.**

   If you see yellow or red entries in the monitoring tree, that means that a warning (yellow) or error (red) condition exists.

   Proceed as follows:

   Ensure that you are displaying the *Open alerts* view.

   The monitor now displays how many alerts there are for each MTE. It also shows the most serious of the alert messages that are waiting.

   Position the cursor on a yellow or red monitoring tree element and choose *Display alerts*.

   The system opens the alert browser and displays the open alerts for that MTE. The alert browser includes all alerts in the branch of the tree that you have marked. Position the cursor "higher" up in the monitoring tree to display a broader range of alerts. If you position the cursor on an MTE at the lowest level, you display only alerts that pertain to that MTE.

   Analyze an alert.

   Each line in the alert browser provides you with summary information on an alert, including the alert message.

The browser also offers two further sources of information. Mark an alert and choose:

– ***Start analysis method*** to start the problem analysis transaction or method that is associated with an alert. (Analysis methods are still being added to the system; not all monitoring tree elements have such methods as yet.)

  For example, with buffer freespace problems, you can start the *R/3 Buffer Tuning Summary* from the alert monitor.

– ***Display details*** to display details on the monitoring tree element. These include the most current values or status messages, the alert thresholds, and the performance data for the last measurement period (only for performance monitoring tree elements). You can graph performance data by marking the appropriate row and choosing *Display performance values graphically*.

7. Has the alert been resolved? Then set it to *Complete*.

Once you have analyzed the problem and have either resolved it or can safely ignore it, you can set the problem to completed.

Mark the alert and choose *Complete alert*. The alert monitor deletes the alert from the open alerts.

See also:

Starting the alert monitor [Seite 837]

Switching views [Seite 841] to see the current state of the system or what has happened in the system in the past

Displaying detailed information [Seite 869] (in addition to that shown in the monitoring tree)

Graphing data [Seite 919]

Changing the level of detail [Seite 842] in the monitor

Turning automatic display refreshing on or off [Seite 852]

Adjusting display options [Seite 903]

Switching between monitors [Seite 844]

Resetting alerts [Seite 867]

Opening and using the alert browser [Seite 835] to respond to alerts

Starting methods [Seite 896]

Completing alerts [Seite 861]

Viewing completed alerts [Seite 863]

Temporarily stopping alert generation [Seite 846]

Tutorial [Seite 827]

Creating Your Own Monitors [Seite 884]

Alerts

Definition

The alert monitor uses thresholds and rules to generate alerts if anything abnormal occurs in your R/3 System or its environment. Alerts direct your attention to critical situations so that you do not have to discover these for yourself. With auto-reaction methods, the alert monitor can notify you of a problem or take action even if you are not currently working in the alert monitor.

For the system administrator, an alert is a "trouble ticket" or a "service request." You should respond to every alert. Use the alert browser and the analysis methods provided by the alert monitor to investigate the alert. When you have resolved the problem or know that you can safely ignore it, you can set the alert to complete. It is then deleted from the display and saved in the R/3 database.

The alert monitor reports alerts up through the monitoring tree. This means that the color-coding for a monitoring tree element (MTE) always represents the highest alert in all MTEs in its branch. For example, if a host system is color-coded red, then one or more of the components in the host system monitoring tree has a red alert.

An alert is uniquely assigned to a node in the monitoring tree (a monitor tree element, MTE), and created when values are sent by a collection method to a monitor tree element. The alert monitor evaluates this incoming information against alert thresholds. If the incoming data exceeds or violates a threshold, then the alert monitor triggers an alert. The alert monitor generates an alert for every abnormal event and keeps track of every alert until it is set to status "complete".

Any MTE that produces an alert is automatically visible in the alert monitor. This is true even if the offending MTE is intended for display only at a higher display level than is currently active (overview, analysis, expert analysis).

Alert colors and their meanings

Color	Meaning
Red	Problem or Error
Yellow	Warning
Green	Everything OK
Gray	No information available

See also:

The Alert Browser: Working with Alerts [Seite 835]

The Alert Browser: Working with Alerts

Purpose

Alerts [Seite 833] report warnings and problems in your R/3 System and its environment. For a system administrator or operator, an alert is a service request. It records the fact that a problem has occurred and provides an opportunity to investigate and resolve the problem. The alert monitor stores alerts that have been resolved in the R/3 database, to provide a record of the problem.

The main tool for working with alerts is the **alert browser**, which you can start from the alert monitor. The alert browser displays a list of alerts that have not yet been completed.

Process Flow

1. Display any alerts that occurred in your alert monitor in the alert browser.
2. Analyze the alert. Was the problem that the alert signals transient, and is it already resolved? If not, what caused the alert and what should be done to resolve the problem?

   Analyzing an alert requires that you collect information on the alert. From the alert browser, you can display more detailed information [Seite 869] that has been collected by the alert monitor.

   You can also start the analysis method [Seite 896]that is associated with the alert and the MTE. This method lets you check the current status of and collect information on the component affected by an alert.
3. When you have resolved the problem, or have determined that the problem was transient or can safely be ignored, select it and choose *Complete alerts.* The alert monitor records the alert in the R/3 database. The alert record is retained until you delete it by reorganizing the alerts database.

See also:

Detailed Tutorial [Seite 829]

Tutorial [Seite 827]

Starting the Alert Browser and Displaying Alerts [Seite 839]

Defining Automatic Alert Notification [Seite 900]

Completing an Alert [Seite 861]

Displaying Completed Alerts [Seite 863]

Reorganizing Completed Alerts [Seite 865]

Stopping and Restarting Alert Generation [Seite 846]

Starting the Alert Monitor

Use

You use the alert monitor to monitor your R/3 System(s).

Procedure

Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.

Starting the Alert Browser and Displaying Alerts

Use

The alert browser is used for analyzing and managing alerts. If the alert monitor is showing alerts, then you should switch to the alert browser to work on them.

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor.* Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor.*
3. To display a legend of the various colors and symbols used in the alert monitor, choose choose *Extras* → *Legend*.
4. Display the *Open alerts* view, if this is not already displayed.
5. Select one or more monitoring tree elements (MTEs) that are color-coded yellow or red for alerts [Seite 833] and open the alert browser by choosing *Display alerts*.

   The alert browser displays all alerts that are "lower" in the monitoring tree than the MTE on which you clicked. Clicking high up in the tree lets you view all alerts in a whole branch of the tree.

   For example, if you click on the top line in the monitor tree (the name of the monitor), then the alert browser shows all alerts that have been triggered for the MTEs in that monitor.

See also:

Completing Alerts [Seite 861]

Reorganizing Completed Alerts [Seite 865]

Displaying Completed Alerts [Seite 863]

Switching Views: System Status, Open Alerts

Use

The alert monitor offers you two views of your system. These are as follows:

- **Current system status** shows the latest reported data on each monitoring tree element (MTE). The alert [Seite 833] color-coding and alert message text reflect this data. They show the most serious current problem.
- **Open alerts** shows where there are alerts that have not yet been analyzed and set to complete. The color-coding for alerts is set according to the most serious unprocessed alert. This view does not necessarily reflect the current status of the system.

You can use these two views to get a complete overview of the system's status:

- When you check the system, start with the *Current system status* view, so that you can see the current state of the system.
- Switch to the *Open alerts* view to see whether there have been any problems in the system since you last used the alert monitor.

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor.* Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor.*
3. The system displays the current system status view. To switch between views, choose *Open alerts*. The top line of the display shows which view is currently active.

See also:

Switching Levels of Detail in Views [Seite 842]

Switching Levels of Detail in Views

Use

The alert monitor offers three levels of detail in the *Current status* and *Open alerts* views. These are as follows:

- **Overview**: Displays only the monitoring tree elements (MTEs) that are essential for monitoring the status of the system. Use this view for routine monitoring of the system during normal operation.
- **Analysis**: This view adds MTEs that are intended for problem analysis or performance tuning to the display. Switch to this display or to the *Expert analysis* view if you need to examine a problem or wish to analyze the system in detail.
- **Expert analysis**: This view adds SAP-internal MTEs. These MTEs are probably useful only when you are working with an SAP consultant or with the SAP hotline or EarlyWatch service. Specialist knowledge may be required to interpret the expert analysis MTEs.

The levels of detail build upon one another. That is, *Analysis and Expert analysis* simply add more MTEs to the *Overview* level of detail.

The alert monitor always shows MTEs for which there are alerts, regardless of the level of detail to which the MTE is assigned. Example: an MTE with a current alert is shown in the *Current status* overview even if the MTE is assigned to the *Expert analysis* level of detail.

Procedure

To switch the level of detail, proceed as follows:

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor*.
3. Choose *Extras* → *Display options* and choose the *Dsply tree* tab.
4. Mark the level of detail that you wish to see in the *View level of the tree* group box.
5. Choose *Continue*. The change is effective immediately.

See also:

Switching Views: System Status, Open Alerts [Seite 841]

Switching Between Monitors

Use

Most monitor sets, including the predefined *SAP CCMS Monitor Templates* and the *SAP Technical Expert Monitors*, contain more than one monitor.

You can skip backwards and forwards between the monitors within a single monitor set, without first having to close the monitor you are in.

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitors you require and choose *Monitor (set)* → *Load monitor.*
3. To switch to the next monitor in the monitor set, choose *Goto* → *Previous monitor* or *Next monitor.*

Stopping and Restarting Alert Generation

Use

You can prevent alerts from being generated for specific monitoring tree elements (MTEs).

For example, assume that you are working on a particular component that has failed. You can prevent additional alerts from being generated for this component while you are working on it.

Prerequisites

To change the alert monitor, you must first activate the maintenance functions.

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor*.
3. To change the alert monitor, choose *Activate maintenance functions*.
4. Select one or more MTEs in the alert monitor and choose *Edit* → *Alerts* → *Configure generation*.
5. The system displays the *Alert Generation* dialog box. Choose *Hide* and confirm your entry. The alert monitor marks the affected MTEs as inactivated.

Result

No alerts will be generated in the affected MTEs until you reactivate alert generation. To do this, repeat the procedure above and choose *Permit* in the *Alert Generation* dialog box.

Monitoring Multiple R/3 Systems

Use

By default, the alert monitor watches over the R/3 System in which you start it.

However, you can monitor multiple R/3 Systems from a single alert monitor. For example, you can make one R/3 System your central system for monitoring. You then simply add your other R/3 Systems to the alert monitor in this central monitoring system.

You can monitor as many R/3 Systems from a single system as you require. Technical factors, such as the speed of and the amount of traffic on your network limit the number of systems that can be monitored. Experience suggests that such practical limits become meaningful only when monitoring multiple tens of R/3 Systems.

Prerequisites

You can only monitor a remote RFC system once you have defined an RFC destination for each R/3 System that you wish to monitor remotely. A remote R/3 System is any R/3 System other than the one in which you are logged on.

Define an RFC destination for a remote R/3 System by calling Transaction SM59. The procedure for defining an RFC destination is described below.

Procedures

Monitoring an Additional R/3 System

1. Choose *CCMS* → *Configuration* → *Alert monitor*. Alternatively, call Transaction RZ21.
2. Choose *Technical infrastructure* → *Create entry for remote monitoring*.
3. Fill out the definition screen as follows:

   Target system: Enter the name of the R/3 System that is to be monitored. Examples: C11, R3P.

   Target system RFC destination: Enter the name of the RFC connection to the remote R/3 System. Choose *Goto* → *RFC connections* to display a list of the connections that have been defined. To define a new connection, see the procedure below.

   What should be monitored?: Select *The whole R/3 System*.
4. Choose *Save*. When you save, the R/3 System automatically tests the connection to the remote system. The system will display an error message if the RFC connection does not work. If there is an error, call Transaction SM59 to test and correct the connection.
5. Display the alert monitor by choosing *Control /Monitoring* → *Alert monitor* or by calling Transaction RZ20.
6. Add the new R/3 System to one or more of your monitors. To do this, load a monitor from a monitor set. Activate the maintenance functions and choose *Change monitor*. The newly added remote R/3 System will appear in the list of selectable MTEs. You can then open the remote system, complete the logon if necessary, and add monitoring elements from the remote system to your monitor. Alternatively, you can create your own monitor [Seite 884].

If you have already set up your own rule-based monitors, then the new system will be automatically added to your monitors.

Defining an RFC Connection to a Remote R/3 System

Before you can monitor a remote R/3 System, you must define an RFC connection to the remote system as follows:

1. Choose *CCMS* → *Configuration* → *Alert monitor*. Alternatively, call Transaction RZ21.
2. Choose *Technical infrastructure* → *Create entry for remote monitoring*.
3. Choose *Goto* → *RFC connections,* or alternatively call Transaction SM59.
4. In the RFC destinations tree, expand the *R/3 Connections* tree.
5. If you do not see an appropriate RFC connection for the system that you want to monitor, choose *Create*.
6. Fill out the definition screen as follows:

   RFC destination: A freely selectable name for the RFC connection that you are defining.

   Connection type: Enter 3, for R/3 System.

   Description: Identifying text.

   In the *Logon* group box, either mark *Current user*, or enter the user specifications for a user in the destination R/3 System. The RFC connection logs on with this user to the remote system. Whether you use the current user or a specific user, this user must have authorization for the alert monitor and for system administration in the target system.
7. Save the definition. The system displays some additional fields in the *Technical settings* group box. Select *Yes* for *Load distrib.* and then save the definition again.
8. Complete the definition by filling out the remaining fields in the *Technical settings* group box:

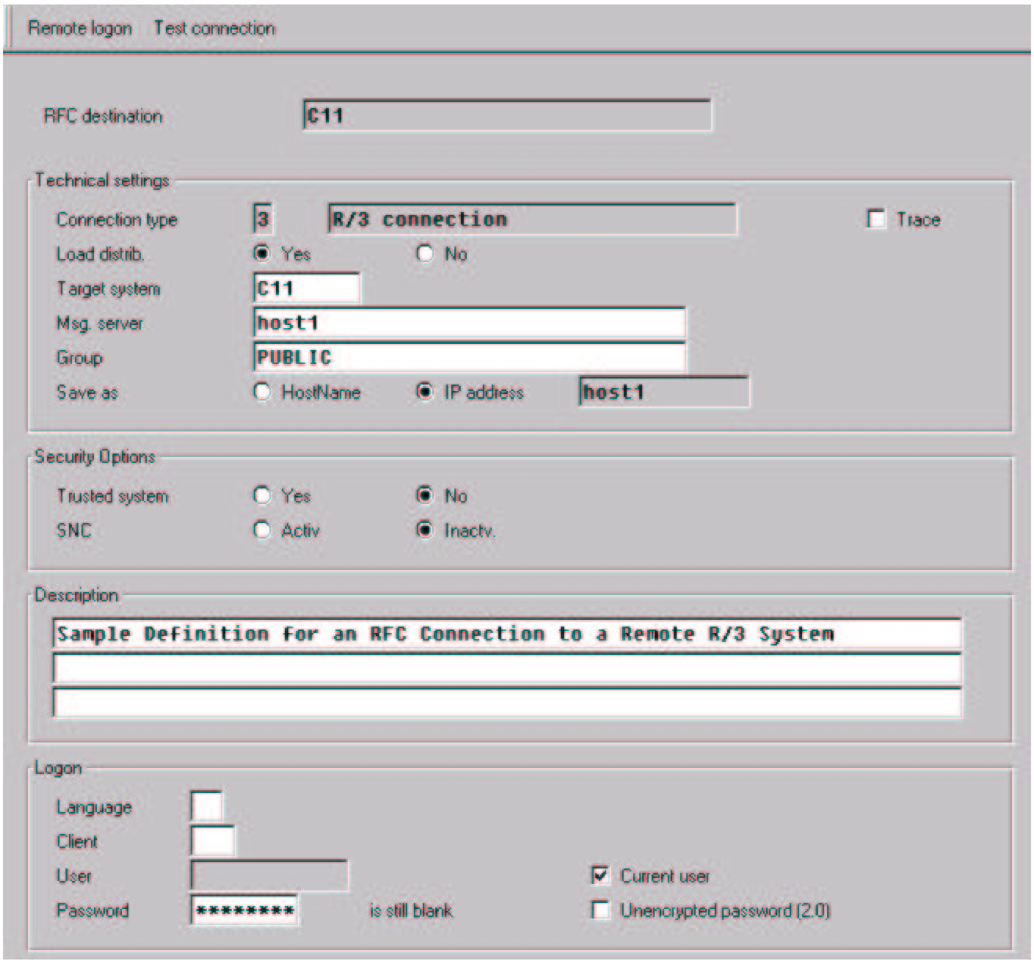
   Target system ID: Enter the R/3 System ID (C11, for example).

   Target host: Enter the host name of the message server. If you are not sure of it, then log onto the target system and call Transaction RZ03. The message server is the R/3 instance with the *M* in the *Services* column. The host name is the first part of the R/3 instance name: **host1**_C11_01.

   Group: Enter the logon group to use on the target system. If you are not sure, enter Transaction SMLG on the target system to display the available groups.

   Save as: Select *IP address*. The message server host name or IP address is automatically entered in the adjacent field.
9. Save the completed definition.
10. Test the RFC connection by choosing *Test connection* and *Remote logon*.

Here is a sample definition screen:



For more information, see: Maintaining Remote Destinations [Extern].

Result

You can monitor multiple R/3 Systems in the alert monitor on a single R/3 System.

Turning Automatic Refresh On and Off

Use

You can decide whether the display in the alert monitor should be periodically refreshed or not. With automatic refresh, the alert monitor updates the display with the information that is currently available in the monitoring system. This refresh occurs at the frequency that you specify in the display options. The refresh does not collect new information; it simply updates the display with the currently available information.

Procedure

To refresh the alert monitor display automatically, proceed as follows:

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor*.
3. Choose *Extras* → *Display options*, and switch to the *General* tab.
4. Mark *Yes, interval* and specify a refresh period in the *Refresh display* group box.

   It is recommended that you do not set a refresh interval of less than 300 seconds.

   You can also update the alert monitor display manually by choosing *Refresh*, or by performing other operations that prompt a refresh (examples of such operations are leaving the alert browser to return to the alert monitor display, or changing from the *Open alerts* to *Current system status* views).

Setting the Time Zone

Use

Before using the monitoring architecture you must check the time zone and set it if necessary. If the time zones are not set correctly then the date and time entries will be incorrectly displayed.

Normally, the time zone will be set during the R/3 installation procedure.

Monitoring the Operating System Using the Alert Monitor (DB2/400)

Use

Using the alert monitor [Seite 812], you can monitor the following objects:

- CPU
- Paging
- OS_Collector
- Lan
- Filesystems
- MainStoragePools

Procedure

1. To call the alert monitor, choose *Administration* → *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand *SAP CCMS Monitor Templates*.
3. Open the MTE (monitor tree element) *Operating System*. The monitoring objects *CPU*, *Paging*, *OS_Collector*, *Lan*, *Filesystems*, and *MainStoragePools* appear.

Monitoring Object	Task
CPU	Monitors CPU utilization
Paging	Monitors page faults
OS_Collector	Checks whether the operating system collector is running.
Lan	Monitors LAN activity.
Filesystems	Monitors the % ASP (auxiliary storage pool) used.
MainStoragePools	Pools, for example, active to wait

For more information, see the Operating System Monitor [Seite 750]

Result

If you use the alert monitor continually during production operation, you can find out quickly and easily whether there are problems with the operating system.

Monitoring Memory Management Using the Alert Monitor (DB2/400)

Use

Using the alert monitor [Seite 812], you can monitor the following objects:

- R3AS4RollMgmtResources
- R3AS4EsMgmtFreeResources

Procedure

1. To call the alert monitor, choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the *SAP CCMS Monitor Templates*.
3. Choose *Entire System* → *R3BasisSystem*.
4. Open the MTE (monitor tree element) MemoryManagement. The monitoring objects *R3AS4RollMgmtResources* and *R3AS4EsMgmtFreeResources* appear.

Monitoring Object	Task
R3AS4RollMgmtResources	Monitors roll management resources.
R3AS4EsMgmtFreeResources	Monitors extended memory resources.

Result

If you use the alert monitor continually during production operation, you can find out quickly and easily whether there are problems with memory management.

Monitoring the Database Using the Alert Monitor (DB2/400)

Use

Using the alert monitor [Seite 812], you can monitor the following objects:

– SpaceManagement

– Performance

Procedure

1. To call the alert monitor, choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand *SAP CCMS Monitor Templates*.
3. Choose *Entire System*.
4. Open the MTE (monitor tree element) *Database*. The system displays the following monitoring objects.

Monitoring Object	Task:
SpaceManagement	Monitors the space situation in your database. State on Disk (DB2/400) [Seite 511]
Performance	Monitors database performance. Database Monitor: Detail Analysis Menu (DB2/400) [Seite 499]
Health	Monitors ASP usage. In the standard R/3 configuration, the database library is located in ASP1.The journal receivers are in ASP2. This separation avoids disk failure. When the ASP2 is filling, the system starts storing journals in ASP1. This is called ASP overflow. ASP overflow is dangerous because it does not ensure possible data recovery.

R/3 Consistency	Monitors the database tables and checks for missing unique indexes (it also takes the SAP exception table into account). Monitors consistency between ABAP Dictionary and Database. There are no thereshold values to be maintained for this monitoring object. **Green**: Unique indexes exist for all database tables. **Red**: At least one unique index for a database table missing. *Actions*: Create the missing unique indexes in the ABAP Dictionary (Transaction SE11) in your R/3 System.

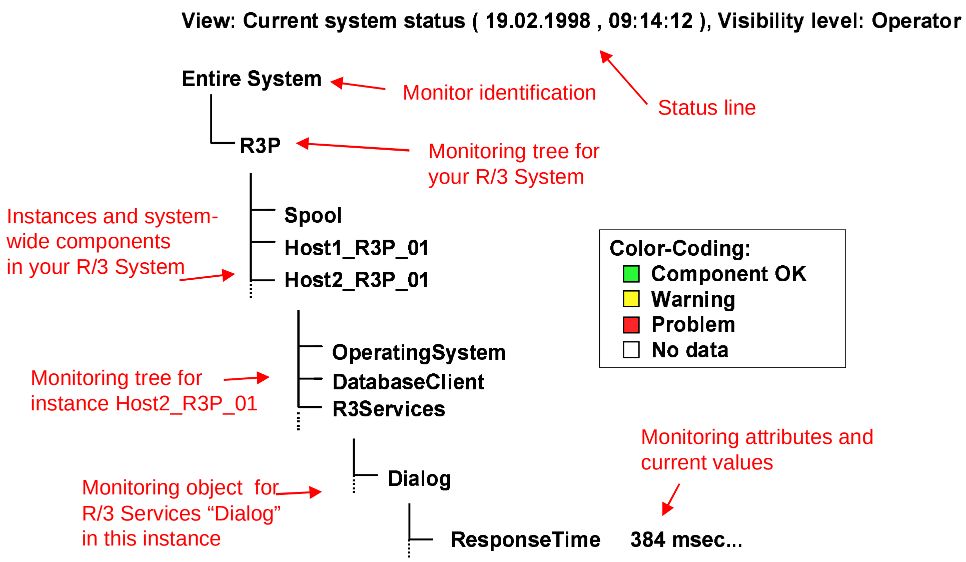
To display up-to-date and detailed information about what the alerts mean and how you should react, use the online help in the R/3 System.

Result

If you use the alert monitor continually during production operation, you can find out quickly and easily whether your database has problems. The result is a more highly tuned database and reduced downtime.

Alert Monitoring Tree: Illustration

Here is a schematic view of the monitoring tree. The monitoring tree is the hierarchy of system components that the alert monitor displays. From the monitoring tree, you can check the condition of your system. You can also respond to alerts, should problems occur in your system.



A little bit of action:

- Pressing F1 on any element in the monitoring tree displays the element's description. If the element has been properly maintained, you can display a text that explains the meaning of the element.
- Clicking on any element in the *Current status* display starts the analysis tool associated with the element. This tool shows you detailed status information on the element.
- Clicking on any element in the *Open alerts* display starts the alert browser, which shows you any alerts that have not yet been analyzed.

Completing Alerts

Use

Setting an alert to the *Complete* status removes it from the set of active alerts that are shown in the alert monitor and alert browser. Set alerts to complete as soon as you have resolved the underlying problem or have determined that the alert was transient or can be ignored.

Completing an alert saves it in the R/3 database. The alert is retained there until you remove it by reorganizing completed alerts [Seite 865].

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor*. Display
3. Display the *Open alerts* for that monitor.
4. Display the alert browser by selecting a node in your monitor and choosing *Display alerts*.
5. In the alert browser, mark the alert(s) that you want to complete.
6. Choose *Complete alerts*.

See also:

Displaying Completed Alerts [Seite 863]

Reorganizing Completed Alerts [Seite 865]

Displaying Completed Alerts

Use

You can display completed alerts in the alert browser. Completed alerts are those that have already been removed from the alert browser and stored in the database.

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transation RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor*.
3. Display the *Open alerts* for that monitor.
4. Start the alert browser by choosing *Display alerts* and then choose *Show alert history*. Completed alerts are inserted into the alert browser according to the sorting scheme that is active.

Reorganizing Completed Alerts

Use

When you complete an alert, the alert monitor saves it in the R/3 database. You can recall completed alerts with the alert history function [Seite 863].

To prevent completed alerts from taking up too much space in the database, the alert monitor watches the amount of database storage used. If thresholds are exceeded, then the alert monitor triggers alerts in the CCMS self-monitoring part of the alert monitor. The relevant object is called *AlertsInDB* and is visible in the expert analysis in Transaction RZ20.

Procedure

You can reorganize the completed alerts with the analysis method for *AlertsInDB*. This method deletes old alerts according to your specifications and reduces the amount of database space used. To do this, proceed as follows:

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the *SAP CCMS Technical Expert Monitors*, position the cursor on the *CCMS Selfmonitoring* monitor and choose *Load monitor*.
3. Position the cursor on *CCMS_Selfmonitoring\Alerts in DB* and choose *Start analysis method.*
4. Specify the date and time from which you want to delete the completed alerts.

Resetting MTEs and Alerts

Use

You should normally never need to reset an MTE. Resetting an MTE is intended as an emergency measure. Only reset an MTE if it is clear that the data and alerts that are being collected for the MTE are faulty and therefore not worth keeping.

Resetting an MTE deletes alerts and performance data. Use resets to clear MTEs only if you do not need to keep records of alerts and if you do not need to retain performance data for the performance database.

By resetting a monitoring tree element (MTE), you can delete open alerts and data that have been collected for that MTE. You can also reset multiple MTEs that you have marked, and to reset all MTEs in a particular hierarchy in the monitoring tree.

Prerequisites

To reset MTEs and alerts, you must first activate the maintenance functions.

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor*.
3. To change the monitor, choose *Extras* → *Activate maintenance functions.*
4. Mark the MTEs that are to be reset and choose *Edit* → *Nodes (MTE)* → *Reset*.

   When you acknowledge the alert monitor's warning, the MTEs are reset to their status when the system was started, or when they were added to the alert monitor. The alert monitor continues to collect data and to trigger alerts for the MTEs.

See also:

Stopping and Restarting Alert Generation [Seite 846]

Displaying Detailed Data and Adjusting the Display

Use

You can display detailed information from both the alert monitor display (from either the *Current system status* view or the *Open alert*s view) and from the alert browser.

Displaying detailed data gives you the data that has been collected for one or more monitoring tree elements (MTEs) in the alert monitor. An alternative to displaying detailed information is to start the analysis method that is associated with an MTE. This method lets you collect information on your own.

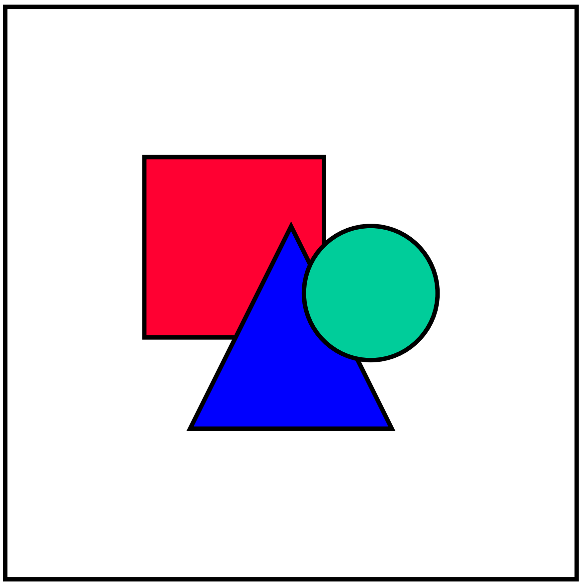
The detailed information display shows you the following:

- The most serious unprocessed alert, if any.
- The current performance data or most recent status message (performance and single message MTEs).
- Alert thresholds (for performance data MTEs)
- Summarized, smoothed data for the last 30 minutes and 24 hours (for performance data MTEs)
- The messages in the message container (for message container MTEs).

You can merge data from more than one MTE. For example, you can display both CPU utilization and dialog response time together. This is useful, for example, if you wish to graph data from related MTEs.

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor*.
3. Select one or more MTEs in the alert monitor or alerts in the alert browser.



In the alert monitor, select only monitoring attribute MTEs. These are the end nodes in the monitoring tree in the alert monitor. In the alert browser, you can display detailed information on any alert.

4. Choose *Display details*. The system displays detailed data for the MTE(s) that you have selected.
5. Choose *Current display variant*. This allows you to:
    - Add or delete fields from the display
    - Redistribute fields from a single list line to two or three list lines per entry
    - Change the order and or width of fields
    - Sort the entries in ascending order or descending order.

See also:

Working with Message Containers [Seite 825]

Graphing Data [Seite 919]

Defining Automatic Alert Notification [Seite 900]

Availability Monitoring

Purpose

With the availability monitoring function of the CCMS alert monitor [Seite 812], you can watch over the availability of remote SAP Systems and their application servers from a central SAP System. 'Availability' monitoring means determining whether a system and its application servers are running and therefore available for work.

Integration

Availability monitoring uses the standard alert-triggering and display functionality of the monitoring architecture [Seite 814]. However, for data collection (determining whether a remote system is active and available), availability monitoring uses an agent. An agent is a small free-standing program that runs outside the SAP System. The use of an agent solution for availability monitoring makes it possible to watch over the availability of a lot of systems efficiently.

Features

Availability monitoring features:

- Use of a free-standing availability agent. The agent solution makes it possible to verify the availability of several remote systems from a single central system, without running the risk that a system that is down will cause the alert monitor display to 'hang'.
- Configurability: You can choose which systems in your environment should be monitored for availability.
- No remote system monitoring entry required. Monitoring for availability is independent of other monitoring in the alert monitor. That is, you do not have to add a remote SAP System to the alert monitor to monitor it for availability.

  The alert monitor does not have to 'log on' to a remote system in order to determine its availability. The result is an extremely fast data collection procedure that enables you to monitor the availability of hundreds of systems from a single CCMS alert monitor, if you so desire. The availability monitor also observes a very brief time-out in the event that a system is not reachable.

Setting Up Availability Monitoring

Use

Before you can start using availability monitoring [Seite 871], two tables used by the monitor must be filled.

Prerequisites

- Decide which system you would like to use for availability monitoring. It should be a system that you use for system management. It should not be a production system.
- Download the availability monitoring agent (CCMS_PING) from one of the SAP SAPSERVx servers. You can download the agent from SAPSERV using FTP from the following directory: general/misc/ccms-ma/availability. The readme file contains directions for installing the agent at the SAP System to be used for monitoring.

Procedure

1. Log on to the system from which you wish to monitor availability. Use a front-end workstation from which you can access many different systems. That is, the SAPLogon or session manager program at the workstation can display many different SAP Systems in your environment.
2. In the SAP System, choose *Development* → *ABAP editor*.
3. Enter RSCSM_UPDATE_CSMNSDIC in the *Program* field and execute the program. Accept the default file location that the program suggests, (c:\winnt\sapmsg.ini or c:\windows\sapmsg.ini) unless you know that the sapmsg.ini file is located in a different directory on your front-end computer.

   The program reads the sapmsg.ini file on your front workstation and uses it to fill the CSMNSDIC table in the SAP System. This table contains the information that the availability monitor needs to contact a remote SAP System (message server host and service name).

4. Choose the systems that you wish to monitor for availability. Choose *Overview* → *Data browser* and do the following:
   - Enter **CSMNSDIC** in the *Table name* field and display the entries in the CSMNSDIC table. For each system that you wish to monitor, enter an **X** in the *AVCHECK* field of the corresponding entry.
   - Enter **CSMSYS** in the *Table name* field and display the entries in the CSMSYS table. For each system that you wish to monitor, add an entry to the table specifying the system name (C11, BJA, ...) in the *SYSGUID* and *SYSID* fields.

   Example: CSMNSDIC:

Sysguid	NSType	AVCheck	Service	DNSName
CX4	3		SapmsCX4	host1.bbb.com
BNT	3	X	SapmsBNT	host2.bbb.com

CSMSYS

Sysguid	Sysid
BNT	BNT

System BNT would be monitored for availability.

5. Install the availability agent in the system in which you are working. The readme file contains directions for installing the agent.
6. In the SAP System, choose *Development* → *ABAP editor*.
7. Enter **RSDS_SYSTEM_AVAILABILITY** in the *Program* field and execute the program. Running this program once starts availability monitoring in the monitoring architecture [Seite 814].

Result

Monitoring of availability has been set up and will continue automatically. You can display the availability data by opening the *Availability: Selected Systems* monitor in the *SAP CCMS Monitor Templates* in Transaction RZ20.

Customizing Availability Monitoring

Use

You can change the behavior of availability monitoring [Seite 871] in two ways:

- You can turn availability monitoring on or off with respect to entire systems
- You can turn availability monitoring on or off with respect to individual application servers of an SAP System. The availability monitor automatically updates table CSMSYSAS with the names of the application servers of a system. By default, monitoring of all application servers is active. However, you can also turn off monitoring for a server.

Procedures

Turning System Availability Monitoring On or Off

To turn off availability monitoring for a system, delete the CSMSYS entry for the system. Also clear the AVCHECK field in the entry for the system in the CSMNSDIC table. You will find instructions for changing these tables in Setting Up Availability Monitoring [Seite 872].

In the availability monitor in Transaction RZ20, you must also manually delete the monitoring tree element (MTE) of the system that is no longer to be monitored. Otherwise, the MTE will remain in the monitor in the inactive (gray) state.

To turn on availability monitoring for a system, add entries for the system to the CSMNSDIC and CSMSYS tables, as described in Setting Up Availability Monitoring [Seite 872]. If the system is not found in the sapmsg.ini file in your front-end computer, then you must add the CSMNSDIC table entry by hand. The CSMSYS table must always be maintained by hand. If availability monitoring has already been activated, then the newly added system will appear automatically in the monitor.

Required fields in the CSMNSDIC table are as shown in the table below.

Field name	Meaning
Sysguid	Name of the system: C11, BJA, ...
Nstype	3 (R/3 Systems)
Avcheck	X (if a system is to be monitored for availability)
Service	Message server service (sapms<SID>: sapmsC11, sapmsBJA)
Dnsname	Message server host (host1.wdf.sap.de)
Route to	Router string for reaching the target system from the location of the CCMS_PING agent, if there is an SAP Router in the path to the target system

Turning Server Availability Monitoring On and Off

To turn off availability monitoring for an application server:

1. Choose Overview → Data browser.
2. Enter *CSMSYSAS* in the table name field and display the entries in the table. For each instance that should no longer be monitored, clear the INST_CHECK field in the entry for the system and instance.

In the availability monitor in transaction RZ20, you must also manually delete the monitoring tree element (MTE) of each instance that is no longer to be monitored. Otherwise, the MTE will remain in the monitor in the inactive (gray) state.

Note that application servers that no longer belong to a system are automatically deleted from the availability monitor.

To turn on availability monitoring for an application server:

1. Choose *Overview* → *Data browser*.
2. Enter *CSMSYSAS* in the table name field and display the entries in the table. For each instance that should be monitored, enter a **C** in the INST_CHECK field in the entry for the system and instance.

   Monitoring of the instance is started automatically when the availability data supplier next runs. This data supplier runs by default every five minutes.

Rule MTEs: Rule Descriptions and Use

Definition

The rule-based monitoring tree element (MTE) technology of the alert monitor allows you to create your own monitors. Rule based monitors have the following qualities:

- The monitor automatically includes sets of MTEs
- The monitor automatically stays up-to-date with changes in the R/3 System landscape.

Rules enable dynamic selection of sets of MTEs according to selection criteria. The other way to create your own monitor (static selection, where the alert monitor is not automatically updated if the system infrastructure changes) enables you to choose single MTEs explicitly.

Use

When you create a rule MTE, you choose a predefined rule for selecting MTEs for your new monitor. The alert monitor interprets the rule and chooses the MTEs that meet the selection criteria that you have specified. The following MTE rules are available:

MTE Selection Rules

- `CCMS_DEFINE_R3_SYSTEMS`

  Creates virtual MTEs for R/3 Systems that have been included in the alert monitor.

  The selection options include ALL (all available R/3 Systems); CURRENT (R/3 System in which the alert monitor is running), and specific systems by name. You can also include in your monitor those systems that you have defined as PRODUCTIVE or TEST systems in your system landscape.

  You use this rule to set up rule-based monitoring across one or more R/3 Systems. Rule MTEs that you add below this MTE are interpreted for each system that you have selected. You add monitoring functionality by creating virtual and/or rule MTEs under this MTE.

- `CCMS_GET_MTE_BY_CLASS`

  This rule inserts monitoring functionality by MTE class. The <MTEclass> parameter lets you add monitoring functionality by MTE type: CPU, response time, background, and so on. The MTE class is displayed as a real node in the monitoring tree, that is, the MTE is monitored by the alert monitor.

  All monitoring objects and attributes are associated with MTE classes. Examples: The *ResponseTime* monitoring attribute is assigned to the MTE class *R3DialogResponseTime*; *AbortedJobs* is assigned to MTE class *R3BPServerSpecAbortedJobs*.

  Since all instances of a particular MTE share a single MTE class, the class selection rule lets you select monitoring objects across SAP instances and systems. For example, you can create a monitor for dialog response times in all SAP instances.

  You use this rule to add monitoring functionality to your monitor by monitoring object type and attribute.

Adding MTEs by class also adds any subordinate MTEs. Example: Adding the MTE class "CPU" also adds the subordinate MTEs *CPU_Utilization* and *5minLoadAverage*, even though these MTEs have different MTE classes.

You can display the MTE classes that are available using the F4 help. This does not display where an MTE class appears in the monitoring tree, so it can be hard to find the MTE class that you want. To do this, open one of the standard monitors of the MTE that you want to include in a monitor. Then choose *Properties* to display the MTE class of the MTE.

- `CCMS_GET_MTE_BY_CLASS_AS_VIRTUAL` and `CCMS_GET_MTE_BY_CLASS_UNDER_CLASS`

  Use these two rules in conjunction. They return the same results as the `CCMS_GET_MTE_BY_CLASS` rule, but the information you require is displayed in a more clearly structured form.

  When you select the `CCMS_GET_MTE_BY_CLASS_AS_VIRTUAL` rule, use the <MTEclass> parameter to include the MTE class as a virtual node in the tree. (Virtual nodes are used as "headings" under which you can group MTEs.

  You then select the `CCMS_GET_MTE_BY_CLASS_UNDER_CLASS` rule. In the <ChildMTEclass> parameter, you then specify the MTE classes that you want to monitor as real nodes in your monitor.

  You may also see two additional rules. These are `CCMS_GET_MONITORING_SEGMENT_NAMES` and `CCMS_GET_MONITORING_CONTEXT_NAMES`. These rules are intended for SAP-internal use only. Although you can experiment with them, SAP recommends against trying to use them in your own monitor definitions.

  The rule `CCMS_GET_AVAILABILITY_FOR_SYSTEM` is reserved by SAP for future use.

Notes

Interpretation of stacked rule MTEs: If you have stacked rule MTEs (that is, inserted `CCMS_GET_MTE_BY_CLASS` under `CCMS_DEFINE_R3_SYSTEM`) then the subordinate rules inherit the SYSTEM specification in the primary rule.

Example: If you specify <ALL> for `CCMS_DEFINE_R3_SYSTEM`, then subordinate "MTE_BY_CLASS" rules also select from <ALL> R/3 Systems.

The effect: Your monitor is structured by R/3 System. Under each system entry in the monitoring tree, you will find the MTE_BY_CLASS selection for that particular system.

Periodic update: The alert monitor periodically re-evaluates the rule MTEs in your monitor. This ensures that changes in the system environment, such as the start-up or shutdown of an R/3 instance, show up in your monitor.

Construction restrictions:

- You cannot stack multiple instances of a rule MTE in a hierarchy. Example: The alert monitor issues an error message if you try to create a CCMS_DEFINE_R3_SYSTEMS rule MTE under another **CCMS_DEFINE_R3_SYSTEMS** MTE.
- You cannot create rule MTEs or virtual MTEs in the *Assigned MTE* hierarchy of MTEs for static selection.

For more information, see: Creating Your Own Monitors [Seite 884].

Monitor Sets

Use

As a system administrator, you can conveniently monitor your system with monitor sets. A monitor set contains several monitors. In turn, these monitors consist of sub-trees of all available objects. You can group the monitors according to task by setting up several monitor sets in parallel.

The monitoring architecture [Seite 814] helps you to monitor several systems in parallel by enabling you to create several monitors and group them in monitor sets.

You can either create the monitor sets temporarily, in order to solve a problem and then delete them, or you can store them permanently in the system.

So that you can reuse the monitor sets, you should save any monitor set that you have created in the database. If necessary, you can reload the set again later.

SAP provides two monitor sets, the *SAP CCMS Monitor Templates* and the *SAP CCMS Technical Expert Monitors.* You cannot modify these SAP monitor sets.

See also:

Creating Your Own Monitors [Seite 884]

Monitoring Multiple R/3 Systems [Seite 848]

Customizing the Alert Monitor

Use

You can customize the alert monitor to adapt when alerts are triggered. You can also define and modify methods, their assignment and their configuration.

You can make specific settings for the following components:

- MTE classes
- Attribute groups
- MTE-specific elements

Customizing in the alert monitor is very convenient. You can access it directly from every monitoring tree element and it provides you with a basis for defining rules and thresholds. You can define comparison values (thresholds) that trigger an alert when exceeded, or fallen short of. You can determine, for example, the weighting of the alerts to be triggered. You can also set permanent thresholds for monitoring attributes.

If you make changes to properties of MTEs in the alert monitor, your changes are saved permanently in the database. Even if a collection method tries to reset the changes to their original state, your changes are preserved and therefore remain in effect in the alert monitor.

The general Customizing settings are valid for all monitor tree elements. However, you can only make type-specific settings for monitoring attributes.

Grouping several monitoring attributes saves you administration duties. To avoid having to maintain the same Customizing settings for many monitoring attributes and then save them in the database, you can assign monitoring attributes to attribute groups. You then only need to maintain the attribute groups. All monitoring attributes assigned to this group are automatically given the new or changed settings.

By editing and saving the relevant attribute group, the corresponding monitoring attributes are assigned the settings saved for this group.

Every application server is assigned monitoring attributes for R/3 Service performance.

You can assign the attributes for the general Customizing settings to a group *ServiceDefaultCust*. The same settings then apply for all these monitoring attributes, with regards to alert severity and so on. You only need to maintain the group.

See also:

Working with Monitoring Properties Variants [Seite 905]

Entering Specific Settings [Seite 889]

Customizing Method Assignments [Seite 891]

Creating Your Own Monitors

Use

SAP has predefined a wide variety of monitors for you in the *SAP CCMS Monitor Templates* and in the *SAP CCMS Technical Expert Monitors*. In addition, you can quickly and easily build additional monitors to meet special requirements.

For example, you want to monitor the relationship between your CPUs, operating system paging, and the R/3 System dialog response time. You can build an alert monitor that contains only these monitoring tree elements (MTEs). You can decide whether this monitor should cover one or multiple R/3 Systems.

The CCMS monitoring architecture provides the following features to increase your flexibility in creating monitors:

- **Static or rule-based MTE selection, or both:**

  You can explicitly pick MTEs for your monitor. This is **static MTE selection**; only the MTEs that you select are added to the monitor, and this selection is not updated.

  You can also use **rule-based MTE selection** (dynamic selection) to build your monitor. In this case, you tell the alert monitor to include all MTEs that meet a particular selection rule [Seite 876] in your monitor. If the MTEs that meet this rule change, then the system automatically updates your monitor. For example, your monitor can be updated automatically when a new R/3 System is added to the alert monitor.

  You can combine both static and rule-based MTE selection in the same monitor.

- **Virtual MTEs:**

  You can define virtual MTEs to structure your monitor. Virtual nodes are used as titles or labels in the alert monitor. They have no function other than to visually group together monitoring tree elements. For example, you can create virtual MTEs as titles for different groups of "real" MTEs in your monitor.

Procedure

1. Choose *CCMS → Control/Monitoring → Alert monitor.* Alternatively, call Transaction `RZ20`.
2. To create a monitor, you must first activate the maintenance functions by choosing *Extras → Activate maintenance functions.*
3. Choose *Monitor (set) → Create* and specify whether you want to create a new monitor set, or to create a new alert monitor in an existing monitor set.

   If you are creating a new monitor set, you should specify who can modify the monitors it contains (you, you and the system administrator, or other users) and whether or not the monitor set should be included in the public monitors, that is, whether it is visible to all users.
4. Enter the name of your new monitor or monitor set and choose *Continue.* When you confirm the name, the new monitor set is added to the list of available monitor sets.
5. The alert monitor displays the complete monitoring tree. You can choose from all of the MTEs that are currently available in the alert monitor. The top line of the tree shows that this is a *New monitor*.

6. There are now three ways to choose monitoring functionality for your new monitor. You can combine all three options. You can either:

   - **Choose the MTEs that you want from the set of all available MTEs (static selection).**
     Expand the *Selectable MTEs* monitoring tree that is already shown to you. Mark the MTEs or branches in the hierarchy that you want to include in your monitor. Only those MTEs that you select will be visible in the completed monitor. The set of MTEs that you choose will not automatically be updated if the system infrastructure changes, that is, your selection is static.

   - **Create a virtual MTE or a structure of virtual MTEs.**
     This enables you to create your own monitor structure. Position the cursor on the *New monitor* line and choose *Edit* → *Create node*. In the dialog box, choose *Virtual node* and enter a name for your new MTE.

     The set of all existing *Selectable MTEs* is available under each virtual node. You can make a static selection of monitoring functionality from these MTEs. For example, you might want to choose background processing MTEs from the R/3 Systems that are offered.

     You can create as many layers of virtual MTEs as you wish. For example, under background processing, you could add other virtual MTEs. You can also add virtual MTEs under rule MTEs (see below).

     You must start defining virtual MTEs, however, from the *New monitor* line. You cannot create virtual MTEs in the set of "real" MTEs, which are reserved for static selection.

   - **Create rule-based MTE selections.**
     Alternatively, or in addition to static selection, you can make rule-based MTE selections under your virtual MTE. The alert monitor dynamically selects and updates the MTEs in your monitor. (The procedure is outlined here. For details on the rules and strategies for building rule-based monitors, see also Rule-Based MTEs: Rule Descriptions and Use [Seite 876].)

     Position the cursor on the *New monitor* line or on any virtual node. Choose *Edit* → *Create node*. In the dialog box, choose *Rule node*. Then use the F4 help to choose a rule for dynamically selecting MTEs.

     In the rule specifications screen, specify a selection parameter for the rule. Example: <ALL> or <CURRENT> for the rule CCMS_DEFINE_R3_SYSTEMS (select from available R/3 Systems). Then choose *Continue*.

     You can then add other rule-based or virtual MTEs under the rule MTE.

7. Save your new monitor by choosing either *Save* or *Activate*. The system prompts you to enter a name for your monitor, which will then be saved. The new monitor is now part of your monitor set. You can use it just as you would any other monitor.

Changing an Existing Monitor

Use

You can modify existing monitors by changing their definitions.

So that you can change a monitor, you must first activate the maintenance functions.

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20. The system displays the *CCMS Monitor Sets.*
2. Choose *Extras* → *Activate maintenance functions.*
3. Expand the monitor set that contains the monitor you want to change and choose *Change.* The system displays the monitor definition.
4. To change the existing static selections, choose the MTE(s) you want to monitor by changing the checkbox selections. To change an existing rule MTE or virtual MTE, position the cursor on it and choose *Change nodes*.

To modify a standard *SAP CCMS Monitor Template* or an *SAP CCMS Technical Expert Monitor*, first copy it to your own monitor set. You cannot modify the monitor definitions in these sets of predefined monitors.

See also:

- Creating Your Own Monitors [Seite 884]
- Rule MTEs: Rule Descriptions and Use [Seite 876]

Changing Properties and Maintaining Methods

Use

You can display and maintain the properties (thresholds, for example) and methods [Seite 891] assigned to monitoring objects in the monitoring architecture in two ways.

You can either display or change the properties and methods assigned to an MTE that is **active** in the alert monitor.

With this option, although you only select one MTE in the alert monitor, this usually changes the properties for the attribute group and MTE class.

Alternatively, you can display or change the properties and methods assigned to nodes that are **not currently active** in the alert monitor. You can also display or change the properties assigned to any monitoring properties variant in the monitoring architecture.

Procedures

To display or change the properties and methods assigned to a specific node that is active in the monitor:

You can display or change the values that are used for the monitoring properties variant that is currently active using the same procedure.

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor.* Alternatively, call Transaction RZ20.
2. Display a monitor from the monitor sets.
3. Select a node in the alert monitor and choose *Properties*.
4. Choose *Display* ↔ *Change* and make the necessary changes.

To display or change properties and methods assigned to nodes that are not active in the alert monitor:

You can display and change the properties assigned to any monitoring properties variant in the monitoring architecture in the same way.

1. Choose *CCMS* → *Configuration* → *Alert monitor.* Alternatively, call Transaction RZ21.
2. From the *Monitoring: Properties and Methods* screen you can display an overview of the methods and properties assigned to the MTE classes and attribute groups in the monitoring architecture. Select one of the options and choose *Display overview.*
3. In the overview, select the entry you want to change and choose *Edit data.*
4. Then choose *Display* ↔ *Change* and make the necessary changes.

Method Assignments

Use

Each node in the monitoring tree of the alert monitor can have up to three different methods assigned to it.

The alert monitor comes with all of the possible method assignments and settings already made. This means that you will need to change method settings rarely (to change the interval at which a method is called, for example). And you will need to change method assignments only if you are adding a new method or want to substitute your own method for the SAP standard methods.

You can execute the following types of methods:

Collection method	Purpose: ABAP program that is responsible for collecting information on an MTE. How is it started? Starts itself or is started automatically by the alert monitor at a specified interval. SAP default: Method assignments made for all monitoring objects and attributes.
Auto-reaction method	Purpose: ABAP program that responds automatically to an alert. Example: The program may send an e-mail warning of an alert to the administrator. See Defining Automatic Alert Notification [Seite 900] How is it started? Started automatically when an alert is triggered. SAP default: No method assignment made for most nodes.
Analysis method	Purpose: ABAP program for displaying information on a node in the monitoring tree and for gathering information on the problem that triggered an alert in that node. How is it started? Started manually when you want to display information or investigate an alert. SAP default: Method assignments made for many nodes. Analysis methods are still being added to the system.

See also:

Defining Methods [Seite 893]

Starting Methods [Seite 896]

Defining, Releasing and Transporting Methods

Use

The standard methods for monitoring tree elements in the alert monitor are created automatically. You can also define your own methods in the monitoring architecture. The method definition is a logical name that assigns the method to tree elements in the alert monitor.

Before you can use a new method, you must first define it and release it in the alert monitor before you can assign it to an MTE. You can also transport the method definition to a different SAP System.

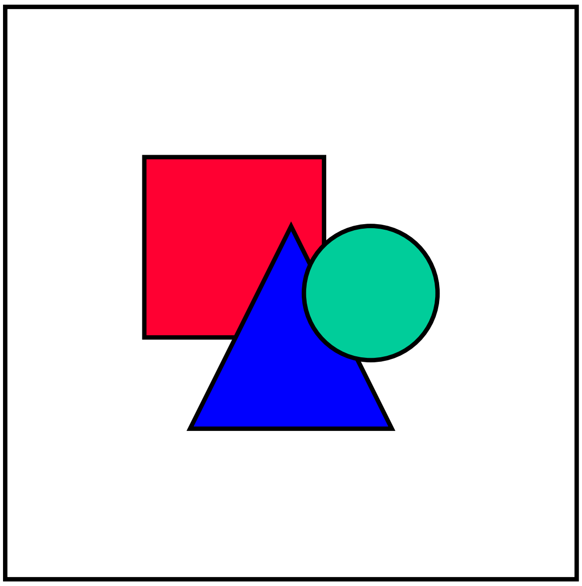
The methods that are predefined already exist in other SAP Systems of the same release. This means that you only have to transport the method definitions that you have created yourself.

You define, release and transport a new method as described below.

Procedure

1. Choose *CCMS* → *Configuration* → *Alert monitor* → *Methods* → *Create new method.* Alternatively, call Transaction RZ21.
2. Enter the name and description of the method that you are defining and select the appropriate radio buttons for your method. For example, if you want to execute Transaction ST04 as a method, select the appropriate radio button and and enter `ST04` in the *Call* field.

   If you want to execute a function module as your method, you must specify the method parameters in the *Parameters* tab.
3. Choose the *Control* tab to specify where the method should be executed. You can have the method executed automatically in the dialog process or as a background job [Seite 74]. Alternatively, you can specify that the method should be executed manually.



SAP recommends that you execute long-running methods as background jobs. When you choose this option, the SAP System assigns the method to available background work processes. You do not need to schedule the background jobs manually, but you can use Transaction SM37 to check their progress.

4. The method you have just defined must be released by the system administrator so that it can be used in the monitoring architecture. To release the method you have just defined in the alert monitor, choose the *Release* tab. Specify whether your method should be used as a collecting method, an auto-reaction method or an analysis method, and choose *Continue.*

5. When you have saved your method definition, you can transport it to a different system by choosing *Transport*.

See also:

Method Assignments [Seite 891].

Starting Methods

Use

MTEs in the alert monitor and alerts in the Alert Browser have methods associated with them.

You can call methods that investigate the status of the alert. For example, you can call an analysis method to analyze a component of the R/3 System; collection methods can be used to add monitoring functionality to the monitoring architecture, and auto-reaction methods can be called by the monitoring architecture to react automatically to an alert once it is triggered.

You can call the methods from the monitor display or in the alert browser.

Procedures

To start a method from the monitor display:

To start all three method types from the monitor display, you must first activate the maintenance function. If you do not activate this function, you can only start the analysis method.

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor.* Alternatively, call Transaction RZ20.
2. Display a monitor from the monitor sets.
3. Choose *Extras* → *Maint. functions on.*
4. Select an MTE in the monitor.
5. Choose *Edit* → *Nodes (MTE)* → *Start methods* → and start the *Analysis method*, the *Collection method* or the *Auto-reaction method*.

To start a method from the alert browser:

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Display a monitor from the monitor sets.
3. Select an MTE in the monitor display.
4. Choose *Open alerts* and then *Display alerts.*
5. Select an alert and then choose *Edit* → *Nodes (MTE)* → *Start methods* → and start either the *Analysis method* or the *Auto-reaction method*.

Assigning Methods to MTE Classes and Monitoring Properties Variants

Use

Collection methods and analysis methods are automatically assigned to MTE classes at system installation by SAP's data supplier.

You will only need to assign a method or methods to an MTE class if you write your own data supplier program. You will also need to assign a method to an MTE class if you want to set up an auto-reaction method or to change the analysis method that was specifed by SAP.

Do not change the assignment of collection methods. This interferes with the correct running of the data supplier program.

From Release 4.6A, methods that have been assigned to MTE classes can also be assigned to monitoring properties variants. The advantage of this is that you can execute different methods in different monitoring properties variants, providing you with greater flexibility in monitoring your R/3 System.

Procedure

1. Choose *CCMS* → *Configuration* → *Alert monitor.* Alternatively, call Transaction RZ21.
2. Choose *Methods assigned to MTE classes* and choose *Display overview.*
3. Select an MTE class and then choose *Edit data*
4. Then choose *Display* ↔ *Change* and then assign the method that you require to the MTE class. You can also specify the monitoring properties variant name for which the method assignment should be used.

   You do not have to assign a method to each MTE class in the monitoring architecture. Where you have not specifed a method for an MTE class, by default the system uses the method that is assigned to the higher-level MTE.

See also:

Method Assignments [Seite 891]

Creating and Changing Monitoring Properties Variants [Seite 905]

Defining Automatic Alert Notification

Use

The monitoring architecture enables you to define auto-reaction methods [Seite 891] that are automatically executed when an alert occurs.

From Release 4.6A, you can assign the *CCMS_OnAlert_Email* method to the MTE classes that are most critical in your system. You will then be informed by e-mail, fax or pager if an alert occurs for these MTE classes, even if you are not currently working with the alert monitor.

CCMS_OnAlert_Email can dial a pager number or send e-mails and faxes to:

- A Business Workplace user in client 000

  The e-mail is sent between 0 and 5 minutes after the alert occurs and is delivered immediately.

- A distribution list or an external e-mail address

  The e-mail is sent between 0 and 5 minutes after the alert occurs, but depending on the settings that are made in SAPconnect [Extern], there may be a delay before delivering mails to external e-mail addresses (that is, e-mail addresses of users who are not defined in client 000 or in the R/3 System). As a result you should set the period of the SAPconnect send process [Extern] to less than one hour.

The message text of the e-mail contains the same information that is displayed in the alert monitor, that is, what the problem is and where it has occurred, when, and the criticality of the alert (a red alert indicates a problem or an error, a yellow alert is a warning).

Prerequisites

To page or send an e-mail or fax to more than one recipient (either an external recipient or one in client 000), you must first create a distribution list [Extern] that contains the pager number, names and/or e-mail addresses of those recipients.

Procedure

1. Choose *CCMS* → *Configuration* → *Alert monitor.* Alternatively, call Transaction RZ21.
2. Flag *Method definitions* and choose *Display overview.*
3. Select the *CCMS_OnAlert_Email* method and choose *Edit data.*
4. Choose *Display* ↔ Change and choose the *Parameters* tab.
5. Fill the fields using information from the table below:

   The method dispatcher is executed every five minutes for short-running methods in the CCMS monitoring architecture that are to be executed automatically. The method dispatcher is processed in client 000 with the system user ID SAPSYS.

   Users who are defined in the R/3 System in clients other than 000 cannot be addressed "directly" in the Business Workplace as senders or recipients and are regarded as external recipients.

You should therefore define client-specific senders and recipients with address type R (remote address) as the *CCMS_OnAlert_Email* parameters.

SENDER	SAP user name of the person in whose name the e-mail is sent. If the user is not defined in client 000, you must also specify the system and client in which the user exists. Example: C11:003:PARKERJ
RECIPIENT	SAP user name of the e-mail recipient, or distribution list containing the recipients' name, external e-mail address, pager or fax number If you want to send an e-mail automatically to an external e-mail address, you can enter the address as the *Recipient* parameter value. However, if you want to send an e-mail automatically to several external e-mail addresses, you must create a distribution list in client 000 that contains these "remote addresses".
RECIPIENT-TYPE ID	Enter the corresponding ID as displayed in the table in Recipient Type [Extern]

If you want to define different recipients for different alerts, copy the complete method definition of *CCMS_OnAlert_Email* in Transaction RZ21 (select the method and choose *Copy*). Rename this copy and then assign the new method to an MTE in the monitor tree.

See also:

Note 0176492 in SAPNet

Assigning Methods to MTE Classes and Monitoring Properties Variants [Seite 898]

Monitoring Properties Variants

Definition

All monitoring tree elements (MTEs) in the alert monitor are predefined with standard general properties, method assignments, and alert thresholds. If you make changes to these settings, your changes are assigned to a monitoring properties variant.

You must activate a monitoring properties variant so that the settings it contains are used actively in the monitoring architecture.

There is a properties variant hierarchy in the monitoring architecture, with the values contained in SAP-DEFAULT [Seite 911] at the bottom level. MTE values that have been changed by customers and stored in an active properties variant are at the top of the hierarchy. From Release 4.6A there is an additional layer in this hierarchy. When you create a monitoring properties variant [Seite 905], you can specify a "parent" properties variant, whose values should be used if the customer-defined properties variant does not contain values for an MTE class. If, in turn, this properties variant does not contain values for that MTE class, then the system uses the values contained in the SAP-DEFAULT properties variant.

From Release 4.6A, you can also assign a monitoring properties variant to an operation mode [Seite 192]. The advantage of this is that you no longer have to activate the variant manually, it is activated when the corresponding operation mode is started.

See also:

Defining Operation Modes [Seite 211]

Creating and Changing Monitoring Properties Variants

Use

A monitoring properties variant [Seite 903] is a collection of settings for one or more MTEs in the alert monitor.

When you activate a properties variant, the changes that are saved in it can be actively used in the alert monitor.

You can define multiple properties variants. Each can have different settings for the same MTEs or MTE classes. Properties variants can also be transported. These qualities mean that you can use properties variants to define "monitoring policies".

If you make changes to an MTE from the alert monitor, then your changes are automatically stored in the properties variant **that is currently active**.

If you call the Customizing transaction directly (Transaction RZ21), then you can make changes to **any** properties variant that you wish.

Procedures

Creating a Variant and Storing Customizing Settings in It

1. Start the Customizing transaction directly by choosing *CCMS* → *Configuration* → *Alert monitor*. Alternatively, call Transaction RZ21.
2. To create a monitoring properties variant, choose *Properties* → *Variants* → *Create*. Enter a name and description for the variant and specify the parent properties variant [Seite 903] whose values should be used if the variant you are currently defining does not have specific values for that MTE class.

   You can also specify the name of a manufacturer of a system management program that provides an agent for logging onto the R/3 System and making use of the CCMS system management interfaces.
3. Choose *Save* and the system returns you to the initial screen of Transaction RZ21.

   You must now assign objects (that is, MTE classes, attribute groups, or individual MTEs from the monitoring tree) to your new monitoring properties variant. You should assign all objects that should share a "monitoring policy" to your variant.
4. In the *Properties* frame of Transaction RZ21, select an object type and choose *Display overview* to view the list of objects.
5. Select the objects that you want to include in the properties variant and choose *List* → *Selected entries* → *Copy to variants*.
6. The system displays a dialog box in which you should choose the properties variant to which the object settings should be copied.
7. If you now want to modify the settings for this object, choose *Change* and then *Display* ↔ *Change*. The changes are stored in the properties variant to which you have assigned the object. Your changes become effective when you activate the variant (as described below).

Making Changes to the Active Properties Variant

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor* (*set)* → *Load monitor.*
3. To change the monitor, choose *Extras* → *Activate maintenance functions* and mark the MTE whose Customizing you wish to change.
4. Choose *Properties* and then *Display* ↔ *Change.*
5. Confirm the dialog box and make the necessary changes. When you save the changes, they are automatically stored in the properties variant that is currently active in the alert monitor. The Customizing transaction shows which variant is active.

Activating a Variant

1. Start the Customizing transaction by choosing *CCMS* → *Configuration* → *Alert monitor.* Alternatively, call Transaction RZ21
2. Choose *Properties* → *Variants* → *Activate*.
3. The system displays a dialog box in which you should choose the properties variant that you wish to activate.

The settings in the properties variant become active in the alert monitor immediately. You do not have to set specific values for each individual MTE class in your variant. If you do not specify your own values for a specific MTE class, the R/3 System automatically uses the customer preset values (the properties variant "✶") or the SAP-DEFAULT values.

The changes only affect the R/3 System in which the properties variant is activated. Each of the systems that you include in an alert monitor has its own Customizing settings.

See also:

Copying Properties Variants [Seite 908]

SAP-DEFAULT [Seite 911]

Assigning Methods to MTE Classes and Monitoring Properties Variants [Seite 898]

Copying Monitoring Properties Variants

Use

You can copy the values from one monitoring properties variant to a different one. Alternatively, you can select the individual properties for specific MTE classes from one properties variant, and copy these to a different properties variant.

SAP recommends that you copy only specific values from one properties variant to another, rather than copying the whole variant.

When you copy the whole properties variant, all of the values it contains are copied to your destination properties variant.

You do not have to set specific values for each individual MTE class in your monitoring properties variant. If you do not specify your own values for a specific MTE class, the R/3 System automatically uses the customer preset values (the properties variant "*") or the SAP-DEFAULT [Seite 911] values.

Procedures

To copy specific properties from one monitoring properties variant to another:

SAP recommends that you use the following procedure.

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor.* Alternatively, call Transaction RZ21.
2. Choose either *Properties assigned to MTE classes* or *Properties assigned to attribute groups* and then choose *Display overview.*
3. Position the cursor on the *Values for variant* column displayed on the screen and choose either *Sort in ascending order* or *Sort in descending order.*
4. Select the individual MTE classes or attribute groups that you want to include in your destination properties variant and choose *List* → *Selected entries* → *Copy to variants.* Use the F4 help to display a list of the properties variants that are available.

To copy a monitoring properties variant to a destination properties variant:

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor.* Alternatively, call Transaction RZ21.
2. Choose *Properties* → *Variants* → *Variant overview*.
3. If necessary, create a new monitoring properties variant [Seite 905] by choosing *Variant* → *Create*.
4. Choose *Variant* → *Copy.* Specify the source and destination variants. You can use the F4 help to display the properties variants that are available in the R/3 System.

5. Select the types of data that you want to copy from the source properties variant to the destination properties variant.
6. Choose *Continue.* The data that you have specified is copied to the destination variant.

   This overwrites the existing data in the destination monitoring properties variant.
7. You can view the contents of the destination properties variant by returning to the initial screen of Transaction RZ21. Select either *Properties assigned to MTE classes* or *Properties assigned to attribute groups* and then choose *Display overview.*
8. Position the cursor on the *Values for variant* column displayed on the screen and choose *Sort in ascending order* or *Sort in descending order.* You can then see clearly properties assigned to the MTE classes or attribute groups in the monitoring properties variant.
9. You can reduce the size of this properties variant by selecting specific properties that you do not want to include in your properties variant and choosing *Delete entry.*
10. To change the properties of an individual MTE class or attribute group, select it and then choose *Edit data.*
11. Choose *Display* ↔ *Change* and change the settings as required.

SAP-DEFAULT

Definition

SAP-DEFAULT is a monitoring properties variant that contains SAP's default values for properties (alert thresholds, for example) of all the MTEs in the monitoring hierarchy (monitoring contexts, objects and attributes in the tree display in Transaction RZ20).

Use

You can use SAP-DEFAULT as a template for creating your own properties variants for specific MTEs in your system. You cannot delete or change the values contained in SAP-DEFAULT, although you can copy either the whole variant or specific values in it to a separate properties variant, and then make changes to this copy.

A properties variant hierarchy exists in the monitoring architecture, with the values contained in SAP-DEFAULT at the bottom level. MTE values that have been changed by customers and stored in an active properties variant are at the top of the hierarchy. From Release 4.6A there is an additional layer in this hierarchy. When you create a new monitoring properties variant [Seite 905], you can specify a "parent" properties variant, whose values should be used if the customer-defined properties variant does not contain values for an MTE class. If, in turn, this properties variant does not contain values for that MTE class, then the system uses the values contained in the SAP-DEFAULT properties variant.

The advantage of the SAP-DEFAULT properties variant is that you do not have to set specific values for each individual MTE in your properties variant(s). If you do not specify your own values for a specific MTE, then the SAP-DEFAULT values will apply.

At a Release upgrade, user-defined properties variants are unaffected by any new SAP-DEFAULT values that are imported into the R/3 System. This means that both the latest default values proposed by SAP and the user's own settings are present in the system. In this way, SAP-DEFAULT can be used as a fallback if you accidentally change the values provided by SAP in the R/3 System, as well as a reference to the values that SAP recommends you should use.

See also:

Creating a New SAP-DEFAULT Variant [Seite 913]

Monitoring Properties Variants [Seite 903]

Creating a New SAP-DEFAULT Variant

Use

SAP recommends that you create a new SAP-DEFAULT monitoring properties variant [Seite 911] when you install a new component of the R/3 System, after a Release upgrade, or if you have written a new data supplier whose values have changed and you want to upload the new values into the database.

When you create a new SAP-DEFAULT variant, the old default values are saved to the properties variant SAP-DEFAULT_OLD_X. When you restart the application server, the newest default values are uploaded into the database and saved as the new SAP-DEFAULT variant. User-defined properties variants are not affected by these changes, meaning that you can compare the values in your own properties variants with the default values suggested by SAP.

Each time you create a new SAP-DEFAULT variant, the R/3 System records the old SAP default values in a subsequent SAP-DEFAULT_OLD_X properties variant. These variants display the values suggested by SAP. You can delete these properties variants if necessary.

Procedure

1. Choose *CCMS* → *Configuration* → *Alert monitor*. Alternatively, call Transaction RZ21.
2. Choose *Properties* → *Variants* → *Variant overview* → *Variant* → *SAP-DEFAULT variant* → *Create new version*.
3. The system displays a dialog box with text for information only. Choose *Continue* and confirm that you want to create a new SAP-DEFAULT variant.

Result

Once you restart the application server, the new SAP-DEFAULT values are uploaded into the database. The old values are stored in the properties variant SAP-DEFAULT_OLD_X.

Performance Database

Definition

Release 4.6A of the CCMS (Computing Center Management System) alert monitor [Seite 812] (Transaction RZ20) delivers the performance database. You can use this to analyze the current performance data and to compare it with the values that were reported over a specific time period.

The performance database stores and displays performance data that was reported to the monitoring architecture. By default, basic data of all performance attributes is recorded in the performance database: hourly average value, hourly sum of value and hourly occurrence (reporting counts). Therefore, each attribute has 24-hour basic data for each day, which is stored in a database table.

You can aggregate the basic daily data over a specific period of time for each hourly slot (x days, weekly, monthly, quarterly and yearly) and use this data for analysis. There is an aggregation hierarchy:

- Weekly aggregate is derived from the daily data
- Monthly aggregate is derived from the weekly data
- Quarterly aggregate is derived from the monthly data
- Yearly aggregate is derived from the quarterly data.

The overall concept of the performance database is to store as much information as possible, but at the same time to retain as little data as possible. As the performance database is constantly growing, it is reorganized on a regular basis according to the aggregate hierarchy described above. Data is marked for deletion and held for a defined number of days. If the data has not been used within the defined days, it is deleted. If it has been used, the data remains in the performance database until the next time the database is reorganized.

See also:

Creating and Scheduling a Reorganization Schema [Seite 921]

Displaying Performance Data [Seite 917]

Assigning a Reorganization Schema to an Attribute Group [Seite 924]

Maintaining the Performance Database [Seite 926]

Displaying Performance Data

Use

The performance database [Seite 915] records all performance data that was reported to the monitoring architecture. You can use it to display the current values of performance data and to compare it with the values that were reported over a specific period of time.

You can aggregate data from the performance database for a single day, or on a weekly, monthly, quarterly or yearly basis.

To display aggregate data, you must follow the procedure described below. No aggregate data exists in the performance database until you request it. Once the data has been requested, it is stored in the performance database as defined in the reorganization schema, until it is requested again or deleted.

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor.* Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor.*
3. Select the performance MTE(s) that you require (if you are unsure about the icons used in the alert monitor, choose *Legend*) and choose *Display details.*
4. Select the checkboxes of the data rows whose performance data you want to display and choose *Performance history.*
5. The system displays the *Time Range for Selecting Performance Data Aggregate* dialog box. Here you can determine the period over which to display the data from the performance database. You can display the data on a daily, weekly, monthly, quarterly or yearly basis. Use the F4 pushbutton to display a calendar to help you make your selections.

See also:

Creating and Scheduling a Reorganization Schema [Seite 921]

Graphing Performance Data [Seite 919]

Graphing Performance Data

Use

You can display performance data that you have called in the Display details [Seite 869] screen. You can use this data to compare at a glance, for example,the individual response times for three application servers, and to analyze any trends that occur within your R/3 System.

Each value in the graphical display is expressed as a value between 0 and 1 so that you can compare different values with each other. For example, you can easily compare CPU performance with the number of users who have logged onto an application server, even though these are expressed as different units of value.

Release 4.6A offers new functions for displaying the curve. You can use the *Fit* function to examine the correlation between different curves. When you choose this function, the maximum value of the curve is assigned to value 1 in the graphical display, and the minimum value is assigned to value 0. As a result you can clearly see the correlation between the two curves.

There are two further display options: you can display only a section of the performance data curve, and you can change the scale of the display so that any differences in the values that were reported become more visible.

The new graphical display can also indicate "holes" in the measurement data, for example, if a system has not been active for a certain period of time, there is a gap in the display to indicate that no values have been reported over that period.

You can also choose how you want to display the performance data graphically. Depending on your preference, you can display the data in several different forms (such as a bar graph, pie chart or table).

Procedure

1. From the R/3 initial screen, choose *Administration* → *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the monitor set that contains the monitor you require and choose *Monitor (set)* → *Load monitor.*
3. Select the performance MTE(s) that you require (display the *Legend* if you are unsure of the icons used in the alert monitor) and choose *Display details*
4. Select the checkboxes of the data rows that you wish to include in the graph. You can combine data from multiple rows that belong to different MTEs.
5. Choose *Display performance values graphically*. The alert monitor displays the data that you have selected.
6. If you want to save the graph by printing it, making a screen capture of the graphics window, mailing it, and so on, choose *Settings* → *Legend.*

See also:

Performance Database [Seite 915]

Creating and Scheduling a Reorganization Schema

Use

You can create your own reorganization schema to define the period of time over which performance data should be retained in the performance database. After this period, the data is flagged for deletion and deleted when the reorganization is next run. You can also determine whether or not to retain data after it is flagged for deletion, in case you want to access this data at a later date.

SAP's default reorganization schema is SAP_DEFAULTSCHEMA. The performance database is reorganized once a week, when the basic data that is reported to the monitoring architecture is aggregated and summarized into weekly data. The basic data is then deleted from the database, so that only the compressed data remains.

To create an reorganization schema, you must execute two programs, **RSALPREORG** and **RSALPFPOLICY** to populate the tables that contain the schema itself. The reorganization schema is run as a background job, which the system administrator also needs to schedule.

The system automatically schedules the background job. You can check its progress by calling Transaction SM37 or by choosing *System* → *Own jobs*.

Procedures

To create a reorganization schema

1. Choose *Development* → *ABAP Editor.* Alternatively, call Transaction SE38.
2. Enter program **RSALPREORG** in the input field and choose *Execute.* Enter the system name, MTE class name and the schema ID and choose *Save.*

   Program **RSALPREORG** populates the table that stores the system ID, MTE class and the schema ID. This enables you to assign a different reorganization schema to different MTE classes in different systems.
3. Next, execute program **RSALPFPOLICY** as described above. This program determines the actual policy itself. Enter the name of your reorganization schema in the *Schema ID* field.
4. Specify the days, months, quarters or years over which you want to keep the performance data. For example, if you want to keep the performance data for 15 days and you have not accessed that data after 15 days, then the system flags the data to be deleted. The data is deleted when the reorganization schema is next scheduled to run (see below).

   You may still want to access this data even after it has been flagged for deletion. In this case, enter the number of additional days, weeks, quarters or years that the data should be retained. For example, if you specify that the performance data should be kept for 15 days but deleted after an additional 5 days, the data is deleted after 20 days.

5. Save your data.

To schedule the reorganization schema as a background job

You now have to schedule when the performance database reorganization will take place. The reorganization is scheduled as a background job. You can either define a time and date for the job, or change the date and time that has already been defined.

There are two steps to this background job: the first is to define the frequency of the data collection, the second is to define the date and time of the reorganization.

1. Choose *CCMS* → *Configuration* → *Alert monitor.* Alternatively, call Transaction RZ21.
2. Choose *Technical infrastructure* → *Performance database* → *Define background job.* The system displays when the performance data is collected and stored in the performance database. To change this background job, choose *Change start date/time.* You can change the frequency of the collection here, although SAP recommends that the performance data is collected every six hours.
3. When you have made the necessary changes, choose *Save.* Then choose *Next step* to display when the performance database is to be reorganized according to the schema that you have previously defined. To change this background job, choose *Change start date/time.* Make and save the necessary changes.

See also:

Background Processing [Seite 74]

Assigning a Reorganization Schema to an Attribute Group

Use

You can define your own reorganization schema and assign it to an attribute group [Seite 817]. The performance data held in the performance database for this attribute group is then reorganized according to the schema you have defined.

Procedure

1. Choose *CCMS* → *Configuration* → *Alert monitor.* Alternatively, call Transaction RZ21.
2. Select *Properties assigned to attribute groups* and choose *Display overview.*
3. Select the attribute group that you want to assign to the reorganization schema you have just created and choose *Edit data.*
4. Choose *Display*↔ *Change.* In the field *Reorganization schema for the performance database,* enter the name of the reorganization schema. Use the F4 help to display a list of the possible entries here.
5. Choose *Save.*

Result

The system reorganizes the performance data for the specified attribute according to the specified reorganization schema.

See also:

Using the Performance Database [Seite 915]

Creating and Scheduling a Reorganization Schema [Seite 921]

Maintaining the Performance Database

Use

The data displayed in the CCMS monitors originates in the performance database `MONI`, with only a few exceptions, such as the data of the operating system and data from database system sources.

In addition to collecting data, the programs `RSSTAT60` and `RSSTAT80` also reorganize `MONI` database and clean up files used for buffering statistical data in local host systems.

Procedure

Specify that the ABAP program `RSSTAT80` should run in the background, which will cause data from application servers to be transferred to database `MONI`. The `RSSTAT80` program collects both data stored by the monitor in a file of the host system and extended table access statistics held in shared memory.

Deleting Local Files

Use

Local host system files should be deleted after the performance data they contain has been transferred to the database. This will help save disk space.

Procedure

Running the **RSSTAT80** program or the **RSSTAT83** program will delete such files if they are 100 MB or larger, as defined by the relevant parameter.

To change this parameter, call the Workload Monitor (Transaction ST03) and choose *Goto → Parameters → Performance database*. Change the parameter: *Delete seq. Statfile after cumulation if size...*.

In the Workload Monitor, you can also manually delete the files. To do this, choose *Workload → Reorganization → Delete seq. stat file*.

Self-Monitoring in the Alert Monitor

Definition

The standard *SAP CCMS Technical Expert Monitors* contain a predefined monitor, *CCMS Selfmonitoring,* which reports on the status of the monitoring architecture and the alert monitor itself.

Check the self-monitoring system whenever you check the rest of the R/3 System. Among other monitoring functions, the self-monitoring system shows whether the alert monitor has been able to:

- Start the data collection methods for which it is responsible (monitoring tree elements for SAPMSSY8, the ABAP report that periodically starts collection methods)
- Allocate and access the shared memory that it needs for collecting and managing data and alerts
- Establish RFC communication links to remote R/3 Systems and components.

Individual alert monitor methods (data collection methods and automatic reaction methods) can also register problems and status in the self-monitoring tree. The self-monitoring tree also registers whether a reorganization of the list of completed alerts [Seite 865] is recommended, in order to reduce the storage needed in the database for these alerts.

See also:

Increasing the Shared Memory Area [Seite 931]

Increasing the Shared Memory Area

Use

When you monitor the change and transport system (CTS) with the alert monitor [Seite 812], this often creates problems with the memory space in the shared memory area. You may need to increase the size of the shared memory area for an application server or servers to provide enough space for all of the log attributes in the tree.

The CTS is a single system that has communication links with several other systems. Each system is displayed in the alert monitor as a separate sub-tree. Since a complex CTS environment can consist of 20 to 30 systems, the number of log attributes can quickly and easily exceed the amount that is permitted.

You should not increase the size of the shared memory area in productive systems. If possible, only increase the size of the shared memory area for the central monitoring system. Finally, ensure that you increase the size of the shared memory area as little as possible as described in the section below.

Procedure

1. Choose *CCMS →Control/Monitoring →Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the *SAP CCMS Technical Expert Monitors* and open the *CCMS* Self-monitoring [Seite 929] *(old version)* monitor.
3. Check the *Space* sub-tree for each application server. Position the cursor on one of the attributes of the *Space* sub-tree and then choose *Start analysis method.*

   Each of the three options (*Messages, FreeAlertSlots* and *AlertsFrequency*) available in the *Space* sub-tree is assigned to the same analysis method.
4. The system displays a dialog text showing the utilization of slots in the monitoring segment. If there are problems with the log attributes, the *Message Log MTEs* displays a low percentage of available slots.
5. Check the application server's instance profile [Seite 266] and increase the value of parameter alert/MONI_SEGM_SIZE.

   The default value for the shared memory area is 10MB. If necessary you can increase the size by 2MB to 12MB, provided that there is sufficient main memory and swap space in the application server.

Result

When the instance is next restarted, the shared memory area for the monitoring architecture and the alert monitor is created accordingly.

See also:

CCMS Alert Monitor for the Transport Management System [Extern]

System Groups in the Alert Monitor

Purpose

A system group contains a set of systems that are to be presented together in the CCMS monitoring architecture [Seite 814]. You can cite a system group in the definition of a monitor; the monitor then includes the systems that belong to the group.

You wish to divide the monitoring of your SAP Systems among the members of your staff. You can accomplish this by defining a system group for each member of your staff. You can then assign the systems that each person should watch to their system group. In the alert monitor [Seite 812], you then define a monitor for each system group. Each of the monitor definitions specifies the system group name in the CCMS_DEFINE_R3_SYSTEMS rule [Seite 876]. The monitor definitions resolve the system group name into the set of systems defined in each group. The result: Each of your staff members can see the systems for which he or she is responsible in a monitor.

The CCMS monitoring architecture [Seite 814] may not be aware of centralized system groups in systems with subordinate repositories. You must register each system in the system group with the monitoring architecture in your subordinate system, otherwise the sub-tree for the system will be invisible when a monitor is displayed.

For example, you can define a central system group that includes the systems C10 and C12. C10 and C12 have already been registered with the monitoring architecture and can be monitored in the central system using Transaction RZ20.

This may not be the case with subordinate systems to which this group is distributed. In such subordinate systems, the monitoring architecture does not include data for unknown systems.

Integration

System group definitions are held in the CCMS system repository [Seite 946]. The system repository is a new component as of Release 4.6C that models one or more SAP Systems.

System groups can be maintained centrally if you register subordinate repositories with a central repository. Any group maintenance changes that you make in the system that houses the central repository are automatically communicated to each of the subordinate repositories. The group definitions are then automatically active in the systems that have the subordinate repositories.

Localized maintenance of system groups is always available. If the name of a system group is unique, it will never be overwritten by a system group definition from the central repository. System groups are always specific to a single system until and unless you subordinate system repositories to a central CCMS system repository.

See also:

Creating Your Own Monitors [Seite 884]

System Groups in the Alert Monitor

Maintaining Centralized System Groups [Seite 936]

Maintaining Local System Groups [Seite 935]

Monitoring Multiple R/3 Systems [Seite 848]

Creating a Central Repository [Seite 953]

Maintaining Local System Groups

Use

You can define system groups that are specific to a particular SAP System. These system groups can co-exist with centrally maintained [Seite 936] system groups. Local system groups let you define groups as required in single systems, to supplement any system groups you may have maintained centrally.

Provided that a local system group has a unique name, it will not be overwritten by group definitions in the central system repository.

It is a good idea to use a naming convention that identifies system groups by the system in which they are maintained. Example: BIE_LOCAL_SYSTEMS for a local system group. This makes it clear which groups are centrally maintained and distributed and which groups are local or system-specific.

Procedure

1. Log on to a system as a user with system administration authorizations (authorization object S_RZL_ADM).
2. Choose *CCMS* → *Configuration* → *Alert monitor*. Alternatively call Transaction RZ21.
3. Choose *Technical infrastructure* → *Maintain system groups* and confirm the dialog box that warns you that any changes you make will be overwritten by centrally defined groups.
4. Expand the tree to display the groups that are already defined.
5. If no groups are defined, create a group by positioning the cursor on the top tree node. Click the right mouse button and choose *Create system group*. Later, when at least one group exists, you can add a system group no matter where the cursor is positioned in the groups tree.
6. Add a system by positioning the cursor on the system group to which the system is to be added, and clicking the right mouse button. Using the possible entries pushbutton, you can choose from systems that are known to the monitoring architecture [Seite 814] and alert monitor [Seite 812] in the system in which you are working.

   You can also add a system by entering its name directly. In this case, be sure to make the system known to the monitoring architecture, otherwise the alert monitor will not display data on the system.
7. You can delete systems and groups by positioning the cursor on group or system in question, opening the context menu, and choosing the appropriate function.

See also:

Monitoring Multiple R/3 Systems [Seite 848]

Maintaining Centralized System Groups [Seite 936]

Maintaining Centralized System Groups

Use

You can define a system group [Seite 933] once in a central SAP System and distribute the system group to other SAP Systems automatically if you are using a central CCMS system repository [Seite 953].

System groups that you maintain in the central repository are automatically distributed to the other systems each time that a subordinate repository is reconciled with the central repository.

Prerequisites

You must set up a central CCMS system repository. Maintain central system groups only in the SAP System that houses the central system repository.

It is a good idea to use a naming convention to identify system groups by the system in which they are maintained. For example: CEN_CORE_SYSTEMS for a system group in the central system repository. This makes it clear which groups are centrally maintained and distributed, and which groups are local and system-specific.

Procedure

1. Log on to the central system as a user with system administration authorizations (authorization object S_RZL_ADM).
2. Choose *CCMS* → *Configuration* → *Alert monitor*. Alternatively, call Transaction RZ21.
3. Choose *Technical infrastructure* → *Maintain system groups* and expand the tree to display the groups that are already defined.
4. If no system groups are defined, create a system group by positioning the cursor on the top tree node. Click the right mouse button and choose *Create system group*. Later, when at least one group exists, you can add a system group anywhere in the tree.
5. Add a system by positioning the cursor on the group to which the system is to be added, and clicking the right mouse button. Using the possible entries help, you can choose from systems that are known to the monitoring architecture [Seite 814] and alert monitor [Seite 812] in the system in which you are working. You can also add a system by entering its name directly. In this case, be sure to make the system known to the monitoring architecture, otherwise, no data about the system will be displayed in the alert monitor.
6. You can delete systems and groups by positioning the cursor on group or system in question, opening the context menu, and choosing the appropriate function.

Changes that you make are saved as you make them. There is no need to save your data before leaving the maintenance tool.

Changes that you make to central system groups are replicated to other systems when the system repositories on those systems communicate with the central system repository. This repository reconciliation occurs automatically if you have scheduled the background job CCMS_REPOSITORY_RECONCILIATION for periodic execution in systems that house ’subordinate’ system repositories. Otherwise, you can start the repository reconciliation and group update manually from a system with a subordinate repository.

See also:

Monitoring Multiple R/3 Systems [Seite 848]

Maintaining Local System Groups [Seite 935]

The CCMS System Repository [Seite 946]

Transaction- and Client-Specific Monitoring

Use

You can use the CCMS alert monitor [Seite 812] to monitor the response times of a specific client or SAP transaction, or transaction by client.

This monitoring can be used to implement a service level agreement, such as where service providers are under contract to provide their customers with SAP services within a specified response time. For example you could base the service level agreement on all response times from the client that the service level agreement customer is using, or on response times of critical transactions, either system-wide or from a particular client.

Release 4.6C of the CCMS alert monitor includes a pre-configured monitor for monitoring the response times of a specific client or SAP transaction, or transaction by client. This is the *Transaction-Specific Dialog Monitor* in the *SAP CCMS Monitors for Optional Components*.

You can either use this monitor or, as SAP recommends, create your own monitor using the procedure below.

Prerequisites

You must first specify which client and/or which transaction should be monitored.

Procedure

1. Call Transaction SE16 and enter the table name `ALTRAMONI`.
2. On the selection screen that is displayed, choose *Execute.*
3. Choose *Create* and enter the transaction that you want to monitor and/or the client.

   To monitor all transactions on a specific client, enter * as the transaction name and then specify the client.

   To monitor a specific transaction on all clients, enter the transaction code and enter * as the client name.
4. Enter a meaningful monitoring object name of your choice. This monitoring object name will be displayed in the monitoring tree that you define later in Transaction RZ20.
5. Enter an MTE class name.

   SAP recommends that you enter the same name for the MTE class and the monitoring object. This makes it easy for you to select by client or transaction in the monitoring definition.

   Use the same MTE class name for clients or transactions that you wish to display together in one monitor. You only need to select one MTE when you define your monitor to display all of this information, rather than selecting individual MTE classes.

6. Enter the *System ID* as `<CURRENT>`. Save these entries.
7. Call Transaction RZ20 and choose *Extras* → *Activate maintenance functions*.
8. In a monitor set of your choice, either create a new monitor definition [Seite 884] or edit an existing monitor [Seite 887].
9. Choose *Create nodes* and then *Rule node.* Select rule `GET_MTE_BY_CLASS` and choose *Continue.*
10. Specify the SAP System(s) that you want to monitor and enter MTE class `ContextR3DialogFocus`. This creates a monitor with:
    - Reference nodes that tell you the table entries that you made in Table ALTRAMONI
    - Monitoring objects and performance attributes that report on the clients and transactions that you selected for monitoring.

    Alternatively, you can enter the MTE class name that you specified in Table ALTRAMONI. When the monitor is generated, the system does not display the nodes that refer to the ALTRAMONI entries. This is not important if you specified a meaningful monitoring object name in ALTRAMONI.
11. Choose *Continue* and then *Generate monitor.*
12. To display the information-only nodes that refer to Table ALTRAMONI, choose *Extras* → *Display options* and select the *Expert analysis* option.

Monitoring Release 3.x Systems

Use

The CCMS monitoring architecture [Seite 814] and the alert monitor [Seite 812] became available with Release 4.0B. With the remote monitoring capability of the monitoring architecture, you can therefore monitor any other SAP System from Release 4.0B or above from a central system.

With the special solution described here, you can also monitor any SAP System Release 3.0 or 3.1 from the Release 4.x monitoring architecture (as of Release 3.0D). The 3.x monitoring solution cannot offer all of the information available from a Release 4.x system, but critical service level data such as availability and dialog response time can be displayed.

You do not have to make any changes to a Release 3.x System to be able to use this solution.

Procedure

1. Download the Release 3.x monitoring agent from the SAPSERVx server. You can download the agent from SAPSERV using FTP from the following directory: general/misc/ccms-ma/3x-monitoring.
2. Log on to the operating system at the host computer that is running an application server from a Release 3.x system that you wish to monitor. Start and register the 3.x monitoring agent according to the instructions in the README file included with the agent download.

   With the registration procedure, you tell the agent to which Release 4.x system it should report its data. The monitoring data on the Release 3.x system will appear in the alert monitor in that Release 4.x system.

   Note the following:

   - Only a single agent is required to monitor all of an SAP 3.x System.
   - You may need to start multiple agents on a single host if 1) there are application servers from more than one Release 3.x system running on the host and 2) you wish to monitor the Release 3.x systems from that host.
   - Each agent can report to only a single Release 4.x system. Different agents can report to different Release 4.x systems.
3. Start the agent according to the instructions in the README file.

Result

Monitoring data on Release 3.x systems is now available in the monitoring architecture (alert monitor) in the Release 4.x system. You can display the 3.x data by creating a monitor and selecting the monitoring context that is created by the monitoring agent (for example, 3x_uw1038_BIN_53).

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 [Seite 1381]

For instructions on how to handle the R/3 Alert Monitor see Detailed Tutorial [Seite 829]

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 [Seite 1384].

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. [Seite 1380]

To find out the threshold values which trigger alerts in the system, see Viewing and Customizing Threshold Values [Seite 1384].

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

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). [Seite 1379]

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 [Seite 750] (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 [Seite 1377] (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 [Seite 1379] (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 [Seite 1385] (transaction ST04).

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.

The CCMS System Repository

Purpose

The CCMS system repository is a database for storing data on the R/3 Systems and other components in a mySAP.com IT landscape. It is a new component with Release 4.6C and will serve, as it is further developed, as a key component of the CCMS central system management infrastructure.

The CCMS system repository serves the following purposes:

- It provides information storage for functions that operate across R/3 System and component boundaries. For example, some kinds of URL generation for mySAP.com use data from the repository to map generic URLs into URLs specific to your IT landscape.
- It supports central maintenance of certain types of R/3 objects. For example, you can maintain system groups for the CCMS alert monitor [Seite 812] and monitoring architecture [Seite 814] in a single central system. The system repository distributes the system groups to other systems in your IT landscape. The prerequisite for this distribution is that the other systems have registered their repositories with the central system.

Features

The system repository is automatically filled with data. Whenever you restart the central server of your system, the CCMS monitoring architecture schedules a background job that fills (first-time) or updates the data in the system repository. The central server in an R/3 System is the server that provides the enqueue (locking) service. The background job simply runs a CCMS data supplier that fills the repository.

The standard CCMS data supplier that fills the system repository collects only data on the local R/3 System, the one in which the repository is located. The standard data that is collected includes information on the servers in the system, RFC destinations, clients, components, and installed support packages.

Other data suppliers, such as those that provide extra, non-standard data for URL generation, run automatically to deposit the data that they need in the repository. You do not have to start any data supplier to fill the repository with data.

You can also fill or update the repository on demand. For more information, see Filling or Updating the Repository [Seite 947].

See also:

Checking for Repository Processing Errors [Seite 948]

Creating a Central Respository [Seite 953]

Creating the CSMREG User [Seite 949]

Displaying Information from the Repository [Seite 952]

Filling or Updating the Repository

Use

The CCMS system repository [Seite 946] is automatically filled with data, and the data is refreshed at irregular intervals. You can also update the system repository manually.

Procedure

1. Choose *CCMS* → *Configuration* → *Alert monitor*. Alternatively. call Transaction RZ21.
2. Choose *Technical Infrastructure* → *System repository* → *Update local repository*. This function schedules a background job that fills or updates the system repository, as required. You will first be asked how the job should be scheduled. The job then runs under your authorizations.

    If a suitable job already has been scheduled and has not run, this job will be used. No extra job will be scheduled to update the repository.

    To fill the system repository the first time, the background job runs less than 30 minutes long in all but extremely large or very busy systems. Updating a repository that has already been filled usually takes only a few minutes in the background processing system.

Checking for Repository Processing Errors

Use

If errors in the following repository operations occur, the CCMS system repository [Seite 946] reports them in the monitoring architecture (Transaction RZ20):

- filling or updating the CCMS system repository
- reconciling one system repository with a central repository.

Procedure

1. Choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
2. Expand the *SAP CCMS Technical Expert Monitors* and open the *CCMS Selfmonitoring (old version)* monitor.
3. Check the *MoniInfra_* branches in this monitor for *CSM* log attributes. If an error occurred during a repository operation, this MTE will be highlighted in red.

   If the MTE is highlighted in red, this usually means that the repository was unable to complete data collection fully on the local system, or that it unable to complete reconciling data with the central repository.

4. Since such problems may often be transient in nature, the correct response is to repeat the operation that failed. If the operation continues to fail, you can purge and re-build the affected repository. To purge a repository, call Transaction SE16 and delete all entries from tables CSMBK and CSMBK_OBJ. If class definition errors have occurred, you can also purge tables CSMBK_CL and CSMBK_TK. These tables are automatically refilled when the data suppliers of the system repository run.

   Before you purge a system repository, you should determine whether it is a central repository, or if it has been registered with a central repository. If so, you must also purge the central repository and all subsidiary repositories that have been registered with it. You must then repeat the registration process for each repository.

   You can refill a repository when you update the local repository [Seite 951]. You can repeat a reconciliation by recreating a central repository [Seite 953].

Creating the CSMREG User

Use

To register a CCMS system repository [Seite 946] with a central repository, the system repository uses an RFC destination with a pre-defined user, CSMREG. This user must be created in the system that houses the central repository, or the registering repository will not be able to use the standard RFC destination. The user is not created automatically; it must be created by you in the client in which the logon for repository updates should take place.

Procedure

1. Choose *System administration* → *User maintenance* → *Users*. Alternatively call Transaction SU01.
2. Create user CSMREG. Remember the password that you create for CSMREG, as you will need to enter it when you register other repositories with the repository in this central system.
3. Create authorizations and a profile for CSMREG and assign the profile to CSMREG.

   The profile should contain the authorizations shown in the table below.

Authorization object	Field	Value
S_CCM_RECV	ACTVT	P0-P9
	TABLE	*
S_RFC	RFC_FUGR	FUGR
	RFC_NAME	SCSMBK_RECONCILE, SCSMBK_DATA_OUT
	ACTVT	16

Updating the Central Repository

Use

If a system has already been registered with the central repository, then you can update the central repository on demand. It is sensible to do this if:

- You have just completed configuration changes on a system, have updated the local repository, and now wish to make the changes immediately visible in the central repository.
- You have not set up the CCMS_REPOSITORY_RECONCILIATION job to run automatically at set periods. In this case, the central repository is updated only when you explicitly start an update with this function.

Procedure

1. Choose *CCMS* → *Configuration* → *Alert monitor*. Alternatively call Transaction RZ21.
2. Choose *Techn. Infrastructure* → *System repository* → *Update (central)*.

   The transaction checks to see if there is currently a suitable background job in the state scheduled, released, or running. If there is not, the transaction schedules a job to update the central repository, after first asking you for job start conditions. The job is called CCMS_REPOSITORY_RECONCILIATION. The job reports changes made to the local repository only since its last report to the central repository. Job run times are therefore usually less than 30 seconds.

   When the job has finished, the central and local repositories are updated. The transaction does not wait for this event.

See also:

Updating the Local Repository [Seite 951]

Updating the Local Repository

Use

The system repository [Seite 946] in each system is automatically updated each time that the central server is restarted or that the monitoring segment of the central server is reset (the CCMS monitoring architecture [Seite 814], Transaction RZ21).

If you wish, you can explicitly update the local repository. This is sensible if you have just completed configuration changes on the system (new clients, RFC destinations, servers, and the like) and wish to have the changes immediately reflected in the repository. It is also sensible before registering the repository with a central repository if the local system has not been restarted recently.

You can also use this procedure to fill the repository the first time, if the automatic auto-discovery has not filled the repository.

Procedure

1. Choose *CCMS* → *Configuration* → *Alert monitor*. Alternatively call Transaction RZ21.
2. Choose *Techn. Infrastructure* → *System repository* → *Update (local)*.

    The transaction checks to see if there is currently a suitable background job in the state scheduled, released, or running. If there is not, the transaction schedules a job to update the local repository, after first asking you for job start conditions. The job is called CCMS_LOAD_REPOSITORY_LOCDATA. Depending upon the size of your system, such a job runs between 10 and 20 minutes, and except in very large or very busy systems, never more than 30 minutes. The job needs to store data in the database, so avoid this action during periods of peak user activity.

    The repository is updated when the job has finished. The transaction does not wait for this event.

See also:

Updating the Central Repository [Seite 950]

Displaying Information from the Repository

Use

You can view the data in a local or central system repository [Seite 946] using the repository's browser.

For example, from a central repository, you could view the customizing settings for clients in any of the systems whose repositories are registered. Or you could see which systems have application servers that run on a particular host.

Procedure

1. Choose *CCMS* → *Configuration* → *Alert monitor*. Alternatively, call Transaction RZ21.
2. Choose *Techn. Infrastructure* → *System repository* → *Display*.
3. The system starts the repository browser.

   The left side of the screen displays the object hierarchy in the repository. Objects are such components as systems, support packages, RFC destinations, hosts, application servers, and the like. The repository models your systems as objects that represent components and as associations that link these objects. On the left side of the screen, you can follow associations up or down the structural hierarchies of your systems.

   To view the properties of an object, double click on the object or drag it over to the table on the right side of the screen. The browser then displays all of the information available on the object, such as the status data on a support package or the settings for a client.

   In Release 4.6C, the legends in the browser are the names of the classes and properties that are used in modeling the SAP System.

Creating a Central Repository

Use

A central CCMS system repository [Seite 946] contains data from more than one system. Setting up a central repository has the advantage that a single repository can display data from multiple R/3 Systems and other components.

To create a central repository, you must register other CCMS system repositories explicitly with the central repository. A central repository is not created automatically.

Registering a system with a central repository unifies the two repositories. This means that:

- Data belonging to the registering system is added to the central repository. For example, information on the clients and RFC destinations of the registering system are added to the central repository.
- Where there is a conflict (the registering system and the central repository each has its own version of some data), the central repository's version of the data is usually used. The central repository's data replaces that in the repository of the registering system. No objects are added to the repository in the registering system. Only object substitutions (central repository representations in place of those in the registering system) take place.

Technically, unifying the repositories means that they share a common namespace for the objects and facts that are represented in the repositories.

With Release 4.6C, you should create a central repository if either of the following applies:

- You wish to use the repository directly through the CCMS repository browser to obtain information on the R/3 Systems in your IT landscape. For example, from a central system, you can view the configuration characteristics of clients, or application servers, or system version data of any system that has registered its repository with the central system.
- You wish to maintain system groups centrally and distribute them automatically to registered repositories in other SAP Systems.
- An SAP component that you are using requires a central repository. Such a requirement will be documented in the installation or customization instructions of the component.

  Some components that use the system repository take care of their own information requirements, without requiring you to unify system repositories. Such components deposit the information that they need directly from the systems in which the data originates, into the system repository on your workplace server. You must unify repositories only if the documentation of a component requires that you do so. Unifying repositories will not affect the operation of components that have stored data on their own in a 'central' repository.

Prerequisites

- You must decide which system is to be the central system with respect to the system repository. If you are using mySAP.com, this should be the workplace server. Choose carefully, because changing the central system requires deleting the contents of the central and registered subordinate repositories and refilling them.
- Decide which systems are to register their repositories with the central system.
- On the central system, create user CSMREG [Seite 949]. This user is used by the registering systems for RFC logons to pass data to the central repository.

Procedure

1. Log on to a system that is to be registered with the central repository. You will need administrator authorizations (S_RZL_ADM, S_RFC, S_BTCH_JOB with Release authorization).
2. Choose *CCMS* → *Configuration* → *Alert monitor*. Alternatively, call Transaction RZ21.
3. Choose *Technical infrastructure* → *System repository* → *Determine repository role*.
4. On the screen that follows, choose *Managed system* and enter the name of the central system.

   Optionally, enter any management information that you wish to record on this system. You can separately save this information, as well as the system-role setting.
5. Choose *Goto* → *RFC Destination*.
6. On the screen that follows, fill out the data required for creating an RFC destination for the subordinate repository to use to reach the central system, and for destinations in the central system back to this system. These destinations are created automatically for you under the standard names CCMS_CSM_CEN_DEST_<central system name> in the registering system, and CCMS_CSM_MNT_DEST_<registering system> and CCMS_CSM_QRY_DEST_<registering system> in the central system.

   The CEN_DEST destination in the registering system has a fixed user (CSMREG) and also requires a password. This destination represents a security risk if CSMREG's authorizations are not limited to those described in Creating the CSMREG User [Seite 949]. Normally, the registering system should be set up to update the central repository automatically at periodic intervals. In this case, the CEN_DEST destination must be active and usable. For maximum security, however, you can disable the destination by deleting the password. This action also disables any automatic periodic update of the central system repository. You must then instead update the central system repository [Seite 950] manually.

   The MNT_DEST destination on the central system is a standard destination for access by dialog users to the registered system; the QRY_DEST is intended for automatic access by the central repository to the subordinate repository for purposes of querying data. Neither destination is saved with a user ID and password, and so neither represents a security risk. The QRY_DEST is not used in Release 4.6C.

   In Release 4.6C, you must create the MNT_DEST and QRY_DEST destinations yourself in the central system.
7. Choose *Enter* to return to the main registration screen.
8. Choose *Enter* to carry out the registration.

Result

The registration procedure creates the local RFC destination that is needed, and schedules a background job called CCMS_REPOSITORY_RECONCILIATION to run under your authorizations. This job reconciles the data in the repositories of the subordinate and central systems and unifies the repositories. The registration is complete when the job has finished running. The transaction does not wait for this event. In all but extremely large systems, the reconciliation job will run to finish in under 5 minutes.

After the reconciliation job has finished, data on the subordinate system is available to you in the central system, for example through the repository browser.

In Release 4.6C, the reconciliation job is scheduled to run only once. If you wish, you can manually reschedule it with automatic periodic restart in the background management transaction (Transaction SM37). Depending on the frequency with which you make reconfigurations on the subordinate system (new clients, new application servers, new RFC destinations), you may wish to schedule a daily or a weekly repetition of the job. The job need not run more frequently than you update the contents of the local system repository.

Note that the system repository does not let you change the roles of the central repository, nor of registered subordinate repositories. Should you wish to change the central repository or have a registered system report to another central repository, you must purge central system repository and each of the repositories that has been registered with the central repository. See Checking for Repository Processing Errors [Seite 948] for more information on purging and re-filling the repository.

The Control Panel

Definition

The control panel displays an overview of activity in your R/3 System.

You can check whether instances were started correctly and whether they are running in the correct operating mode, that is, whether automatic operation mode switching is possible.

From the control panel you can:

- Start and stop instances [Seite 69]
- Maintain instance definitions [Seite 238] and operation modes [Seite 211]
- Log onto a host system or an instance

  You must select the server or servers on which you want to execute an operation.

Starting the Control Panel

Use

The control panel displays a list of the host systems and the instances in your R/3 System. The list provides information on:

- Application server name(s)
- Application server service and status
- The active operating mode

Procedure

To start the control panel from the R/3 main menu choose *CCMS* → *Control/Monitoring* → *Control panel*. Alternatively, call Transaction RZ03.

You can sort the information that is displayed by server or status by choosing *Edit* → *Sort* → *By server name* or *By status.*

See also:

Creating the Instance Definition for all Servers [Seite 234]

Developer (Internal) Traces

Procedure

Developer, or internal, traces are records containing technical information that will be useful when investigating errors. To be able to use this information effectively, precise knowledge of the host system in which your R/3 System is running and of the R/3 System is required.

This type of trace is especially useful for analyzing host problems and errors that affect the operation of your R/3 System. The `disp+work` process in the developer trace shows, for example, why it is not possible to generate work processes.

To display information on a trace, from the R/3 main menu, choose *Tools* → *CCMS* → *Control/Monitoring* → *Control Panel* → *Utilities* → *Trace files* → *Dispatcher trace* **or** → *Work processes...*

See also:

Checking the Operation Mode with the Control Panel [Seite 225]

Activating and Deactivating Developer Traces

At the command line, enter the command to activate the trace or set the level of detail for the trace. This can be done in all R/3 programs. You normally need to add these options to the start command for SAP processes in the start profile of an instance of your R/3 System.

See also: *The System Administration Guide*.

The following options are available:

- For no trace, use TRACE=0.
- To write error messages to the trace file, use TRACE=1.
- For a full trace, use TRACE=2. The trace entries actually being written may vary according to the R/3 program being traced.
- To also trace data blocks, use TRACE=3.

To set trace options for a complete instance, use `rdisp/TRACE=<value>`. The values are the same as those listed above.

The trace files have the following names:

Component	File name
Dispatcher	`dev_disp`
Work process	
Task handler	`dev_w` is the interval from 0 to 1 minus the number of the work process.
Dynp (screen processor)	`dev_dy`
Roll	`dev_ro`

Paging	`dev_pg`
DB interface	`dev_db`
ABAP/4 processor	`dev_ab`
Enqueue (lock)	`dev_eq`
Logging	`dev_lg`
Spool	`dev_w`
Message server	`dev_ms`
SAPTEMU (presentation)	`dev_st<logon name>`
APPC server (CPI-C Gateway)	`dev_appc`

In the trace files, lines with information about errors begin with `***ERROR=>` and contain the name of the calling function, the action that ended with an error, the error number (if a system call is involved), and the line of the C module that failed.

Lines written for the system log entries begin with `***LOG<message no.>`.

Accessing Log Files

Use

You can display the logs for an instance at any time using either the CCMS Monitor or the Control Panel on your current host system even if the instance concerned has been shut down. For more information, see Checking the Operation Mode with the Control Panel [Seite 225]

With the exception of the system log, all logs are stored in a system's home directory.

Procedure

To display server status and alerts:

1. From the R/3 main menu, choose *Tools* → *CCMS, Control/Monitoring* → *Control Panel*
2. In the Control Panel, select an instance.
3. Choose *Utilities* → *Trace files* → *Home directory*

You can also log on directly to the operating system of an instance that has been shut down and use an editor to analyze the log files.

Although the local system log is stored as an operating system file, you will need to analyze the log from the R/3 System.

To determine how much of a log file to display:

1. From the R/3 main menu choose *Tools* → *CCMS, Control/Monitoring* → *Control Panel.*
2. Select an instance.
3. Choose *Trace files* → *Settings*.

Workload Monitor

Purpose

The workload monitor (Transaction ST03N) is an expert tool that is intended for use by EarlyWatch and GoingLive teams. Release 4.6C of the workload monitor has been reworked as part of the EnjoySAP initiative, enabling you to monitor and evaluate workload statistics [Seite 969] more easily and intuitively than before.

You use the workload monitor to analyze statistical data from the R/3 kernel. When analyzing the performance of a system, you should normally start by analyzing the workload statistics. For example, you can display the totals for all instances and compare the performance of individual instances over specific periods of time.

You can use the workload monitor to display the:

- Number of configured instances for each SAP System
- Data for all application instances, not just the one you have logged on to
- Transactions used and the users that call them
- Number of users working on the different instances
- Performance history for recent periods for all instances
- Response time distribution and resource consumption for any application server
- Application server workload for today or for a recent period.

Integration

The new workload monitor is still being enhanced and has not yet fully replaced the old Transaction ST03.

Until all of the functions have been replaced, you will still need to use Transaction ST03 to display hitlists [Seite 983], aggregation in transactions and to access detailed information on a transaction by double-clicking on an individual profile (and other profiles such as memory, RFC, user, table and client).

Features

The workload monitor has a new interface that is divided into two parts. Use the tree structures on the left of the screen to choose the user mode [Seite 967] and profiles that you require. The system then displays the results on the right.

The new workload monitor has the following features:

- Standardized ABAP ALV grid controls, enabling you to search for and filter the information you require
- User modes [Seite 967]
- Cross-client information (information on application servers other than the one you have logged on to)
- Save user-specific views [Seite 982]
- Graphics of the statistics for each task type

User Modes

Definition

The workload monitor [Seite 965] (Transaction ST03N) has been reworked for Release 4.6C. The new design was conceived with different users in mind, so that the system displays different default functions depending on the user mode you choose.

Use

There are three types of user mode. Each shows different types of information.

- Administrator mode

  This is the default user mode. SAP recommends that you use this user mode. It provides quick access to today's workload statistics [Seite 969] and provides an overview of the workload distribution. You can also display the functions that relate to the collector.

- Service engineer mode.

  This mode provides you with the workload statistics for today and the previous week, as well as an overview of the workload history and distribution, and a detailed workload analysis.

  By default the system displays the total statistics for all application servers.

- Expert mode

  This mode provides users with all of the functions that are available in Transaction ST03N. You can display all of the workload data that is available (daily, weekly and monthly records).

Integration

The user mode that you choose determines the functions that are displayed on the Transaction ST03N initial screen.

See also:

Displaying Application Server Workload [Seite 974]

Displaying Spool Statistics [Seite 980]

Displaying Transaction Workload [Seite 979]

Displaying Users per Instance [Seite 978]

Displaying Workload over Specific Time Periods [Seite 976]

Analysis Views and Profiles

Definition

The *Analysis views* enable you to analyze the workload statistics [Seite 969] according to profiles. A profile is the application server workload displayed as a specific view.

The user mode [Seite 967] that you choose determines the profiles and data that are available. You can analyze the workload for:

- A specific server [Seite 974]
- Transactions [Seite 979] (Standard or Early Watch)
- A specific time period [Seite 976]
- Spool statistics [Seite 980]
- Application statistics [Seite 984]

Profiles are only displayed if the relevant data is available. For example, if no spool statistics records are available for an application server, the system does not display the *Spool statistics* profile in the *Analysis views* tree.

Workload Statistics

Definition

System statistics are written to the R/3 kernel and can be displayed using the workload monitor [Seite 965] (Transaction ST03N). They provide system administrators with varied and detailed information on CPU time, numbers of database changes, roll out times and so on.

For Release 4.6C, the workload monitor [Seite 965] has been enhanced to include GUI times, frontend network times and RFC time.

Use

You can use these statistics to compare the response times of different application servers over a specific period of time, for example, or to compare the number of users who are working on various application servers.

Integration

You can display the workload statistics for all task types (such as *dialog, background, RFC, ALE* and *update*). Alternatively you can display the statistics for a specific task type.

The table below defines the most important workload statistics:

Average CPU time	Average CPU time used in the work process. During a dialog step, the CPU of the application server is used for processing (loading, generating, database request processing, ABAP processing and so on). The CPU time is determined by the operating system. At the end of a transaction step, the R/3 work process queries the CPU time from the operating system. The CPU time is therefore not an additive component of the response time, unlike the wait, roll-in, load and database time.

Average response time	The average response time of a dialog step is the average time measured from the time a dialog sends a request to the dispatcher work process through the processing of the dialog up to the time the dialog is completed and the data is passed to the presentation layer. The response time between the SAP GUI and the dispatcher is **not** included in this value. The response time does NOT include the time taken to transfer data from the R/3 frontend to the application server. For networks with low performance, this can create a highly subjective response time. The transfer time is contained in the GUI network time. Response time is usually split into wait time plus dispatch time. The SAP response time is composed of the following components: Response time = wait time + dispatched time Dispatched time = Generation time during runtime + Load time for programs, screens and CUA interfaces + Roll times to roll-in work data + ABAP processing time + Database time + Enqueue time for logical SAP lock procedures + CPIC/RFC time + Roll wait time (apart from task types RFC/CPIC/ALE). The CPU time is not an additive component of the response time, but the sum of the CPU time that is used by the individual components. The CPU time therefore provides additional, independent information on the response time.
Average wait time	The time an unprocessed dialog step waits in the dispatcher queue for a free work process. Under normal conditions, the dispatcher work process should pass a dialog step to the application process immediately after receiving the request from the dialog step. Under these conditions, the average wait time would be a few milliseconds. A heavy load on the application server or on the entire system causes queues at the dispatcher queue.
Average load time	The time needed to load and generate objects such as ABAP source code and screen information from the database.
Database calls	The number of parsed requests sent to the database

Database requests	The number of logical ABAP requests for data in the database. These requests are passed through the R/3 database interface and parsed into individual **database calls**. The proportion of database calls to database requests is important. If access to information in a table is buffered in the SAP buffers, database calls to the database server are not required. Therefore, the ratio of **calls/requests** gives an overall indication of the efficiency of table buffering. A good ratio would be 1:10.
GUI time	The GUI time is measured in the work process and is the response time between the dispatcher and the GUI.
Roll ins	Number of rolled-in user contexts.
Roll outs	Number of rolled-out user contexts.
Roll in time	Processing time for roll ins.
Roll out time	Processing time for roll outs.
Roll wait time	Queue time in the roll area. When synchronous RFCs are called, the work process executes a roll out and may have to wait for the end of the RFC in the roll area, even if the dialog step is not yet completed. In the roll area, RFC server programs can also wait for other RFCs sent to them
Average time per logical DB call	Average response time for all commands sent to the database system (in milliseconds). The time depends on the CPU capacity of the database server, network, buffering, and on the input/output capabilities of the database server. Access times for buffered tables are many magnitudes faster and are not considered in the measurement.

Response Times: Rules of Thumb

Definition

The following times give you an overview of optimal response times in your SAP System. Use the workload monitor [Seite 965] to display the current workload statistics [Seite 969] in your system.

Performance Data	Time
Average response time	approx. 1 second (dialog), <1 second (update)
Average CPU time	approx. 40% of average response time
Average wait time	<1% of average response time
Average load time	<10% of average response time
Average DB request time	approx. 40% of average response time

The operating system can affect these values by about 10%.

The proportion of CPU time to the complete time elapsed should not fall below approximately 5%. A low percentage might mean that there is an I/O bottleneck, or that a database overload has caused excessive wait times.

The average response time for dialog (online) transactions should not be more than 1 second, as this can unduly tax users' patience.

Database Requests	Time
Direct reads	<10 ms
Sequential reads	<40 ms
Changes	>25 ms

High Values for	Reason
DB req. (Change/Comm.)	Database or index problems?
Load time	Buffer problems?
Wait time	Not enough work processes? Locked tasks? Long-running transactions?

The times for direct reads and changes should not exceed 10 milliseconds in a healthy database system.

The sequential reads time should not exceed 30-40 milliseconds.

Displaying Application Server Workload

Use

You can use the workload monitor [Seite 965] (Transaction ST03N) to analyze the workload statistics [Seite 969] for each task type (for example for background processing, dialog processing, update processing, ALE, and RFC) for the application servers in your SAP System.

You can now switch easily between statistics for **all** of the application servers configured in your system, by double-clicking the instance in the tree.

You can also switch easily between the statistics for different time periods.

The user mode that you select determines the period over which you can display the workload for a specific application server:

- Administrator mode [Seite 967] displays the workload for the current day only
- Expert mode [Seite 967] displays weekly and monthly cumulated statistics for the workload as well as the daily, weekly, and monthly statistics for each instance
- Service engineer mode [Seite 967] displays information on the workload for the current day and the previous week.

If you have already saved a user-specific view [Seite 982], the system automatically displays the statistics that you have chosen when you call Transaction ST03N.

Procedure

1. Start the workload monitor by calling Transaction ST03N.
2. Select the user mode [Seite 967] that you want to use (expert, administrator or system engineer).
3. Expand the selection tree to display the application server and the time period that you want to analyze. Double-click the application server (or time period) that you want to analyze.
4. Ensure that you have selected the *Workload overview* in the *Analysis views* below the selection tree.

   Total statistics are not calculated until the end of each day. If you want to display the total workload on an application server so far today, the TOTAL data has to be calculated first. You can do this either in the dialog mode or as a background process. The value that the system calculates will be overwritten at the end of the day with the complete workload statistics for that day.
5. Choose the task type and tab page that contains the parameters you require.

To display all of the statistics from each tab page in a single table, choose *All data.* If you wish you can choose *Show/hide tree* to save space on your screen. Choose the same pushbutton to redisplay the tree.

See also:

Displaying Workload over Specific Time Periods [Seite 976]

Displaying Application Statistics [Seite 984]

Displaying Workload over Specific Time Periods

Use

You can use the workload monitor [Seite 965] to display and compare the time profiles [Seite 968]. A time profile displays the workload for a specific task type over a specific period of time.

You can display the workload [Seite 969] of the instances that are configured in your SAP System, not just the instance that you have logged on to. You can also switch easily between the information for these instances.

You can display and analyze the time profile for one individual task type, or for all task types. You can display the workload for the following parameters:

- Times
- Database
- Proportion of response time
- GUI times
- All data.

You cannot display all of the parameter options for the task types. For example, you cannot display the GUI time for background processes, since this is not applicable. If you try to display this, the system returns you to the *Times* tab and the *GUI time* tab disappears from the table.

Procedure

1. Start the workload monitor by calling Transaction ST03N.
2. Choose the user mode that you want to use (expert, administrator or service engineer).
3. Expand the selection tree to display the application server that you want to analyze and the time period.

   The user mode that you select determines the time period for which you can display workload statistics. For example, if you use the administrator user mode, you can only display the statistics for the current day. For the expert and system engineer modes, you can switch between the time profiles of the task types you have selected for other application servers and/or time periods.
4. In the *Analysis views*, choose *Time profile*.
5. Choose the task type and corresponding parameters that you want to analyze.

See also:

Workload Statistics [Seite 969]

[Displaying Application Server Workload [Seite 974]]

[Displaying Transaction Workload [Seite 979]]

[Displaying Users per Instance [Seite 978]]

[Displaying Spool Statistics [Seite 980]]

Displaying Users per Instance

Use

You can use the workload monitor [Seite 965] to display the number of users working on an application server. You can display the number of dialog steps that each user has executed, and check that application server response times are acceptable.

Procedure

1. Start the workload monitor by calling Transaction ST03N.
2. Choose the user mode [Seite 967] that you require (administrator, expert or service engineer).
3. Choose *Load history and distribution* (or *Load distribution* in the administrator user mode) and then *Users per instance.*
4. To display the users who are logged on to a particular application server, position the cursor on an instance and choose *Choose instance.* The system displays a list of users, the dialog steps they have executed and response times.

    To display all of the application servers that a user has logged on to, choose *Instances per user.* To return to the previous screen, choose *Users per instance*.

See also:

Displaying Application Server Workload [Seite 974]

Comparing Workload Between Servers [Seite 981]

Displaying Transaction Workload

Use

You can use the workload monitor [Seite 965] to display workload statistics [Seite 969] for individual transactions. You can display, for example, the number of dialog steps for each transaction on an application server.

If you choose the system engineer or expert user mode [Seite 967], you can display either the standard transaction profile or the EarlyWatch profile.

The EarlyWatch profile is intended for monitoring systems by the EarlyWatch service. It is almost the same as the standard profile, but has additional columns that are sorted differently. Note that you cannot select task types in EarlyWatch profile.

Procedure

1. Start the workload monitor by calling Transaction ST03N.
2. Choose the user mode that you require (expert, administrator or service engineer)
3. Select the application server and time period that you want to analyze. The user mode that you choose determines the time period for which you can display statistics.
4. From the Analysis views [Seite 968], choose *Transaction profile* and then choose either the standard or the EarlyWatch profile.

See also:

Workload Statistics [Seite 969]

Displaying Application Server Workload [Seite 974]

Displaying Spool Statistics

Use

You can use the workload monitor [Seite 965] (Transaction ST03N) to display spool statistics for the application servers in your SAP System.

The spool statistics are useful for spool administrators who want to check what has been output in each application server, the number of dialog steps and so on.

The *Spool statistics* profile [Seite 968] may not be available for every application server that is configured in your SAP System.

Procedure

1. Start the workload monitor by calling Transaction ST03N.
2. Choose the user mode [Seite 967] that you require (expert, administrator or service engineer).
3. Expand the selection tree to display the application server and the time period that you want to analyze and double-click it. The user mode that you choose determines the time period for which you can display statistics.
4. From the Analysis views [Seite 968], choose *Spool statistics*.

   If no spool statistics are available for a specific application server, the system does not display the icon for the *Spool statistics*.

Comparing Workload Between Servers

Use

You can use the workload monitor [Seite 965] to compare at a glance the workload statistics [Seite 969] for the application servers configured in your SAP System.

You should choose the *Workload overview* profile from the *Analysis views* if you want to examine the workload of each application server in great detail. Use the procedure described below if you want to compare the workload for all of the application servers.

Procedure

1. Start the workload monitor by calling Transaction ST03N.
2. Expand the selection tree and choose *Load (history and) distribution* → *Compare instances*.
3. The system displays the *Workload: Compare all instances for one selected period* screen. From here you can choose one or more of the following:
   - *Choose* to display workload statistics for a specific instance according to task type
   - <<< or >>> to display the rest of the table information
   - *Instance -* or *Instance +* to display information for other instances
   - *Other task type* to display the statistics for a specific task type

     When you choose *Choose* or *Instance +* or *Instance -,* the system automatically displays two tables. The first table displays three instances, enabling you to compare the information. The application server that you have chosen is always displayed in the middle line.

     The second table then displays the broken-down statistics for this instance.

See also:

Displaying Application Server Workload [Seite 974]

Displaying Workload over Specific Time Periods [Seite 976]

Business Transaction Analysis [Seite 985]

Saving User-Specific Views

Use

With Release 4.6C of the workload monitor [Seite 965] you can save user-specific default views. For example, if you are mainly interested in analyzing the response times for a time profile [Seite 968] in the service engineer user mode [Seite 967],you can save this as your default view. When you next call Transaction ST03N, the system displays this view automatically.

You can save a specific profile, the tab page that you have selected, the task type and the ALV layout to a user-specific view.

When you save a user-specific view, the expert and administrator user modes include the application server that you have logged on to in this view.

If you have chosen the service engineer user mode, the system will display the TOTAL server.

Procedure

1. Call Transaction ST03N and select the user mode that you want to use.
2. Expand the selection tree and double-click the sever and/or the time period that you want to analyze.
3. Select the Analysis view [Seite 968] that you want to use and specify the task types and parameters that you want to display as your default view.
4. Choose *Save view*.

Result

When you next call Transaction ST03N, the system displays automatically the view that you have just saved.

See also:

User Modes [Seite 967]

Displaying Application Server Workload [Seite 974]

Displaying Top 40 Response Times

Use

From the workload monitor [Seite 965], you can display further detailed workload statistics [Seite 969] for an application server configured in your system, over a specified time period. These statistics, or hitlists, help you pinpoint the transactions that use the most resources, by displaying a list of the 40 dialog steps with the longest response time or the 40 dialog steps with the highest number of database calls.

For Release 4.6C you should display the hitlists using the old workload monitor in Transaction ST03.

You can use the hitlists to answer the following questions:

- Does the same program always have high response times
- Is this program running at the same time in several sessions (locking work processes)?
- How many KB of information are transferred?
- Could the program be started at night in the background?

Procedure

1. Start the workload monitor by choosing *CCMS* → *Control/Monitoring* → *Performance Menu* → *Workload* → *Analysis.*
2. Choose *Choose for analysis* and select the instance that you want to analyze.
3. In the next dialog box, select the time period that you want to analyze.
4. Choose *Goto* → *Hitlists* → *Top 40 resp. time,* or *Top 40 DB requests,* or choose *Top Time* on the application toolbar.

   You can switch between the response time list and the DB requests list by choosing either *Top time* or *Top requests*.
5. To display more details, double-click a line in the table. Choose *Task* to display a list of memory usage.

See also:

Workload Monitor [Seite 965]

Response Times: Rules of Thumb [Seite 972]

Workload Statistics [Seite 969]

Displaying Application Statistics

Use

Application statistics enable you to analyze resource consumption in more detail than using the workload statistics [Seite 969].

Using special calls within the ABAP code, the system collects statistics for individual parts of an application. Whereas the workload statistics analyze a complete dialog step, application statistics analyze the resources that an individual function consumes within a dialog step (such as price determination). In some cases, the application statistics can replace an SQL trace.

The statistics collected by the application statistics are generally the same as the standard workload statistics (response time and so on).

Prerequisites

To use the application statistics, special function calls must be implemented in the application programs. Application statistics cannot be collected without these calls. In addition, the following profile parameters must be set as indicated: `stat/as_level=1` and `stat/as_collect=2`. The application statistics collector (`RSSTAT87`) must be scheduled using table `TCOLL`. You must also activate the individual components of the application statistics by calling Transaction ST03N and choosing *Collector & Perf. Database → Parameter & Reorg. → Statistics records → Application statistics.*

Procedure

1. Call the workload monitor (Transaction ST03N).
2. Specify the application servers and the time period that you want to analyze.
3. From the *Analysis views,* choose *Application statistics.*

If no application statistics are available for the given instance / time period, the system does not display the icon for the application statistics.

See also:

Workload Monitor [Seite 965]

Business Transaction Analysis

Use

You can use the workload monitor [Seite 965] to display the business transaction analysis. This analysis calculates the use of system resources by individual transactions, and provides the **most detailed analysis** of a transaction and dialog steps. The selection criteria includes: user, transaction, program, task type, start date, and start time.

The time frame that is analyzed may be larger than the interval that was set by the read time, since the system always attempts to analyze complete transactions, although a complete analysis of long-running transactions may not be possible. As the business transaction analysis is time-intensive, set the interval that you want to monitor as short as possible (around ten minutes).

Procedure

1. Start the workload monitor by calling Transaction ST03N.
2. Select the user mode [Seite 967] that you want to use.

   You can only display the business transaction analysis using the expert or service engineer user modes.
3. Choose Detail analysis → Business transaction analysis.
4. You can choose one of the following display modes on the initial screen:
   - *Show all statistic records, sorted by start time*
     Displays a chronological list of dialog steps and resource consumption (corresponds to individual record statistics, transaction statistics)
   - *Show all records, grouped by business transaction*
     Displays a list of dialog steps and their resource consumption. The individual dialog steps for an individual transaction are included in the list.
   - *Show business transaction totals*
     Displays the total for each transactions. This is the variant that is most frequently used.

     If you set certain restrictions in the first two modes listed above, such as only for certain users, transactions and so on, the system only reads the corresponding records. In the third mode, the system reads all records, but only displays those that match the selection criteria. ensure you take this into account if you choose *Disp. mode* to switch to a different display mode.
   - *Include statistics from memory*
     The system first buffers the statistical data before reading it into the statistics file. This option forces the buffer to write the data to the file before the analysis starts. This ensures that all transactions can be analyzed.

- *Include application statistics*
  The system also displays application statistics [Seite 984].
- *Server selection*
  This option restricts the analysis to specific application servers in the system.
- *Additional options*
  These can be used to set the wait time for RFCs and the maximum runtime of the analysis, as well as the time interval that you set for the analysis.

See also:

Displaying Transaction Workload [Seite 979]

Displaying Workload over Specific Time Periods [Seite 976]

Comparing Workload Between Servers [Seite 981]

Working with the Workload Collector

Use

The workload collector collects the statistics [Seite 969] that are displayed in the workload monitor [Seite 965] (Transaction ST03N). The workload collector is started by, but runs independently of, the performance monitor collector [Seite 988].

You can use the workload monitor to display a log for the workload collector, if you have problems with the workload monitor. The log may give indications as to the reason for any problems with collecting the statistics.

In addition, if the statistics file for a specific instance becomes too large, you can use the workload monitor to delete it.

With Release 4.6C, server statistical data is collected and cumulated once a day from the table MONI [Seite 989]. Data for the weekly, monthly, and total statistics are then cumulated using this daily data.

Procedure

1. Start the workload monitor by calling Transaction ST03N.
2. Select the user mode [Seite 967] that you want to use (expert, administrator or system engineer).
3. Choose *Collector* → *Workload collector* and double-click *Log*.
4. To display more detailed information, including the collector status, double-click a log entry. The system displays a list of all the programs that were executed at this time point.
5. If you want to delete the statistics file for a specific instance, choose *Collector* → *Workload collector* → *Delete statistics file* and double-click the instance you require.

Working with the Performance Monitor Collector

Use

The performance monitor collector starts several reports that are executed once an hour. These reports collect statistics on your SAP System, including report RSSTAT83, the workload collector [Seite 987], which provides the statistics for the workload monitor [Seite 965] (Transaction ST03N).

You can display a log of the performance monitor collector to ensure that the reports were started successfully. You can also check when these reports were started.

Procedure

1. Start the workload monitor by calling Transaction ST03N.
2. Select the user mode [Seite 967] that you want to use (expert, administrator or service engineer).
2. Choose *Collector* → *Perf. monitor collector* and double-click *Log*.
3. The system displays a log of the background jobs or collectors that have run, according to time and day.
4. To display more detailed information, position the cursor on a line and choose *Display protocol,* or double-click the line. The system displays a log of all the programs that were executed at this time point.
5. To display an overview of when the different reports are executed, choose *Collector* → *Perf. monitor collector* and double-click *Execution time.*

Reorganizing the Table MONI

Use

The ABAP program RSSTAT60 reorganizes the MONI database table. This table is specially keyed and cannot be accessed using conventional methods.

You can specify how much data you wish to retain in MONI, thus limiting the maximum size of the table to optimize performance of the CCMS monitors.

With Release 4.6C, server statistical data is collected and cumulated by default once a day from table MONI. Data for the weekly, monthly, and total statistics is then cumulated using this daily data.

This means an increase in performance compared with other releases, where the weekly and monthly statistics were collected every hour. In addition, the total server statistics are no longer cumulated at 4am. Instead, they are cumulated when all of the data for the previous day is available. As a result, the worklod monitor statistics are more consistent.

Procedure

1. To schedule program RSSTAT60, call the workload monitor [Seite 965] (Transaction ST03N) and choose the *Expert* user mode [Seite 967].
2. Choose *Collector & Perf. DB* → *Parameter & reorg.* → *Collector & reorg.*.
3. Choose *TCOLL configuration* to display table TCOLL. Here you can enter and schedule program RSSTAT60. The programs entered in TCOLL are automatically executed by the data collector program RSCOL00. (This is also possible in Transaction ST03N by choosing *Perf. Mon. collector* →Execution time.)
4. To set the parameters that affect RSSTAT60's reorganization of the table MONI, call Transaction ST03N and choose *Collector & Perf. DB* → *Parameter & reorg.* → *Collector & reorg.*. Choose *Modify parameters*.

   You can display the values for *Standard statistics* and *Time comparison data*. The parameter values and recommended settings for these two areas are as follows:

   - Standard Statistics

   The values here define the retention periods for the daily, weekly, and monthly generation of statistics.

   SAP recommends setting a value of at least "2" for all three types of statistics, thus causing the statistics to be retained for two days, two weeks, and two months respectively. However, longer retention periods should be used for small and mid-sized SAP Systems, as well as for testing new SAP Releases.

   During a reorganization run, RSSTAT60 deletes all date whose retention time has expired.

When statistics are deleted through the new retention times used in a reorganization of the table MONI, these lost statistics cannot be recovered except through the standard techniques for database recovery.

- Time Comparison Data

The values here define retention periods for time comparison data, which is taken from the daily, weekly, and monthly generation of statistics, and which is retained in a compressed form requiring little space in MONI.

SAP recommends setting a retention period of 20 days, 20 weeks, and 14 months respectively.

Result

The smaller MONI is, the better the performance of the various CCMS monitors is.

If you use the above recommendations for the parameters of the standard statistics and the time comparison data, MONI requires between 10 MB and 50 MB of space on the hard disk, depending on the size of your system and the type of statistics being collected.

See also:

Working with the Workload Collector [Seite 987]

Working with the Performance Monitor Collector [Seite 988]

Configuring the Data Collector

Purpose

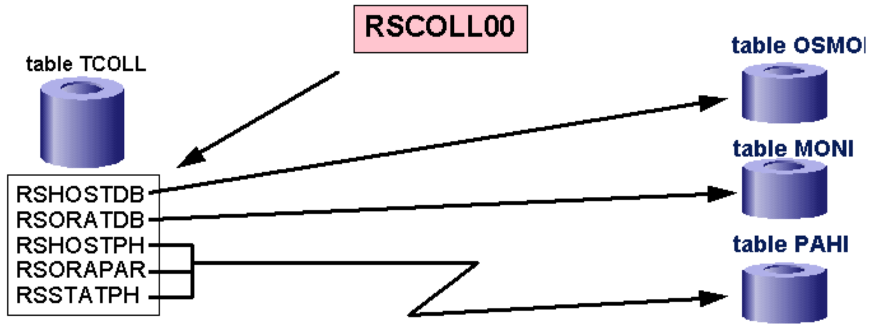
The data collector, the ABAP program `RSCOLL00`, is a centrally located program that uses RFC to start and monitor a number of other programs on the various application servers for the purpose of gathering statistics regarding the various R/3 instances, the database and the operating system. (For a central R/3 instance RFC is not required.)

The data collector and the programs it triggers enable these statistics to be:

- Collected together
- Stored centrally
- Compressed
- Analyzed
- Reorganized

Thus, the data collector triggers programs that, for example, gather statistics for the Operating System Monitor using data read locally from the operating system kernel. Other programs analyze the disk space still available to the database system or log all changes to the operating system, database, and R/3 System parameters.

The data collector is scheduled as a periodic background job and automatically reads which programs to trigger and when to trigger them from table `TCOLL`. Thus, to specify which programs to run, when they are to run, and on which R/3 instance, simply maintain the table `TCOLL` (see also R/3 Note 12103 and R/3 Note 127642).



Procedure

Scheduling the Data Collector in Background Processing

Schedule data collector program `RSCOLL00` to run every hour as an R/3 background processing job using Transaction SM37 and using the job name `SAP_COLLECTOR_FOR_PERFMONITOR` or `COLLECTOR_FOR_PERFORMANCEMONITOR`

Displaying the Data Collector Logs

You can display the data collector logs using the `RSCOLL20` program. As with the SAP background processing logs, data collector logs are reorganized by the data collector itself.

You can also display the data collector logs by calling the Workload Monitor (Transaction ST03), and then choosing *Environment* → *Data collector* → *Display protocol*.

See also:

Scheduling Data Collector Reports [Seite 994]

Ineligible Queue (OS/400) [Seite 763]

Scheduling Data Collector Reports

Use

In the following sections are short descriptions of the ABAP programs that can be started by the data collector program `RSCOLL00`. The descriptions will help you decide which of these programs to run in your system by scheduling them in table `TCOLL`.

RSHOSTDB [Seite 997]

RSHOSTPH [Seite 995]

RSORSNP [Seite 999]

RSDBPREV [Seite 1001]

RSORATDB [Seite 1003]

RSORAPAR [Seite 1005]

RSORA811 [Seite 1007]

RSSTATPH [Seite 1009]

RSSTAT60 [Seite 1011]

RSSTAT80 [Seite 1013]

RSEFA350 [Seite 1013]

RSHOSTDC [Seite 1013]

RSORAWDB [Seite 1013]

RRSSTAT83 [Seite 1013]

RSSTAT82 [Seite 1013]

RSSTAT81 [Seite 1013]

RSSTAT90 [Seite 1013]

RSSTAT98 [Seite 1013]

RSSTAT89 [Seite 1013]

RSTUNE80 [Seite 1013]

RSRFCDLT [Seite 1013]

Procedure

To schedule the programs in table `TCOLL`, call the workload monitor (Transaction ST03) and choose *Environment* → *Data collector* → *Collector frequency*.

See also R/3 Note 12103 and R/3 Note 127642 in SAPNet.

RSHOSTPH

Definition

`RSHOSTPH` checks whether system parameters of the host operating system have been changed since the last time it ran and logs any changes that have been made to database table `PAHI`.

As the parameter changes are stored only once a day, it is sufficient to schedule this program in table `TCOLL` with `system =*` to run once a day in each R/3 instance.

If there are several instances of the same R/3 System on one server, the data collector starts the program only once on this server.

RSHOSTDB

Definition

`RSHOSTDB` updates the performance database with operating system performance data.

Reads data for the Operating System Monitor from the main memory of the server and writes it to database tables `OSMON` and `MONI`.

Use

Since the data is kept for no more than 24 hours in main storage, this program should be scheduled in table `TCOLL` with *system* `=*` to run at least once a day in each R/3 instance.

If there are several instances of the same R/3 System on one server, the data collector starts the program only once on this server.

RSORASNP

Definition

RSORASNP collects snapshot-like database statistics (e.g. number of user calls, long table scans etc.). For this, it reads the current database statistics from the Oracle table V$SYSSTAT and stores them in table DBSNP. Obsolete entries are simultaneously deleted from this table. The number of entries deleted depends on the setting for parameter STATS$TABLES_DAYS_TO_KEEP in table TSEXC; the default value is 120 days.

For Oracle this report is called indirectly via report RSDBPREV [Seite 1001] which means you do not need to schedule it directly in table TCOLL.

See also:

The Database Collector in Background Processing [Seite 418]

RSDBPREV

Definition

RSDBPREV calls a database-specific report which generates and stores current statistics (snapshots) of the database system.

For an Oracle database, this report is RSORASNP [Seite 999].

See also:

The Database Collector in Background Processing [Seite 418]

RSDBPREV

RSORATDB

Definition

RSORATDB analyzes the growth of certain tables, indexes and tablespaces and writes the results to database table MONI.

This program carries out forecasts on the basis of values from the past, in order to be able to issue warnings in good time of any storage bottlenecks that may occur.

The results are stored with a timestamp (date, time). This means that you should execute the program on the database system at least once a day (entry in table TCOLL, system=C). If there is no dialog instance active on the database server, the data collector switches to the first available application server.

Considerable CPU resources are required on the database server to run RSORATDB, depending on the database size. This means that you should not schedule this program too frequently. Therefore the data determined will not always be up-to-date but will reflect the status at the time of the collector run.

See also:

The Database Collector in Background Processing [Seite 418]

RSORAPAR

Definition

Logs database parameter changes

RSORAPAR checks whether database system parameters have been changed since the last time this program was executed and logs the changes to database table **PAHI**.

As these parameter changes are saved per day only, it is sufficient to run this program once a day on the database server (in table **TCOLL**, system = C). If there is no dialog instance active on the database server, the data collector switches to the first available application server.

See also:

The Database Collector in Background Processing [Seite 418]

RSORA811

Definition

Deletes old backup logs generated by the SAP utility programs BRBACKUP and BRARCHIVE.

RSORA811 deletes obsolete log entries in the database tables SDBAH and SDBAD. For more information on the SAP utility programs BRBACKUP/BRARCHIVE, refer to the documentation SAP/Oracle DBA in CCMS [Seite 1105].

See also:

The Database Collector in Background Processing [Seite 418]

RSORA811

RSSTATPH

Definition

Logs SAP parameters changes

Like programs **RSHOSTPH** and **RSORAPAR**, program **RSSTATPH** logs changes to SAP parameters. You should run this program daily in each SAP instance (in table **TCOLL**, **system = ***).

RSSTAT60

Definition

Reorganizes the performance database table `MONI`.

Data that is older than the time specified in the relevant parameters is deleted

Due to the number of database accesses involved, schedule `RSSTAT60` in table `TCOLL` so that it runs on the database server (in `TCOLL`, *system* `= C`). A dialog instance must be active on the database server.

RSSTAT80

Definition

This report is no longer recommended. Instead, use `RSSTAT83`, which requires less time as it is not started sequentially on all servers. See also R/3 Note 127642 in SAPNet.

RSSTAT88

Definition

As RSSTAT89, but more processes are started in parallel. See Note 127642.

RSSTAT87

Definition

Schedules a job that starts RSSTAT88 approximately half an hour after SAP_COLLECTOR_FOR_PERFMONITOR.

RSSTAT83

Definition

Transfers local statistics.

This program is executed on every R/3 instance one after the other. It reads the local statistics for the various R/3 instances and writes the statistical data to the performance database (table `TCOLL` , `system = *`). The program notes which data has already been read to prevent data from being read or transferred multiple times.

Schedule this program to run hourly using table `TCOLL`. To create TOTAL statistics, use report `RSSTAT82`. For more information, see SAPNet Note 127642.

`RSSTAT83` writes its own log. To display this log, go to transaction `ST03` and choose *Environment* → *Data collector* → *Stat. rec. log*.

Parameters for RSSTAT80/83 and RSSTAT87/88/89

Definition

The following describes the parameters that you can set in the workload monitor (Transaction ST03) for programs RSSTAT80/83 and RSSTAT87/88/89.

Use

In the R/3 System, the job `SAP_COLLECTOR_FOR_PERFMONITOR` or `COLLECTOR_FOR_PERFORMANCEMONITOR` is executed hourly and in turn activates the report `RSCOLL00`. This report in turn activates further reports according to the entries in the table `TCOLL` (for example, report `RSSTAT80` or `RSSTAT83`). These reports read the statistics file of every R/3 instance, format and compress the data, and then store it in table `MONI`.

The report `RSSTAT80` is activated hourly by table `TCOLL` on all instances. Report `RSSTAT83` runs in parallel on all servers in the system, which reduces the runtime of report `RSCOLL00` considerably. As `RSSTAT83` runs in parallel on servers, it cannot calculate TOTAL server statistics. To calculate these statistics, if needed, you must run `RSSTAT82`. For more information, see R/3 Note 127642 in SAPNet.

Report `RSSTAT89` reads the application statistic records, compresses the information and then stores it in the database. In place of this report, you can run report `RSSTAT88` (runs in parallel on multiple servers, see R/3 Note 127462) in SAPNet. `RSSTAT88` is started by report `RSSTAT87` approximately a half an hour after `SAP_COLLECTOR_FOR_PERFMONITOR`.

Structure

The parameters (and the fields in which you set the parameters) are grouped in the workload monitor under the following headings on the *Workload Analysis: Parameters Relevant to MONI Reorg.* screen.(To call this screen, call Transaction ST03 and choose *Goto* → *Parameters* → *Performance database.*)

Residence times of statistical data (for all servers)

These parameters specify how long statistics are retained for each time period (days, weeks, months) (see Reorganizing MONI Database [Seite 989]).

Statistical data to be cumulated & Controls for the collector

You can influence the runtime behavior of the statistics collector using several parameters. If there are collector performance problems, you can reduce the runtime by excluding some of the statistics details.

- `Cumulate statistics for period 'Week'.`
  Generate weekly statistics.
- `Cumulate statistics for period 'Month'.`
  Generate monthly statistics.
- `Cumulate hitlists (Top time & Top Requests).`
  Display 40 dialog steps within selected period with the highest response times and database requests.

- `Cumulate memory profile.`
  Transaction memory use.
- `Cumulate accounting & client profile.`
  Generate statistics according to user or client.
- `Cumulate RFC profiles.`
  RFC statistics.
- `Cumulate terminal in/out messages (Req. Bytes profile).`
  Data exchange between application server and clients.
- `Cumulate CUA proc. profile.`
  Statistics for function codes (only generated by report `RSSTAT80`).
- `Cumulate server statistics to a systemwide total statistics (only valid for the RSSTAT80/RSSTAT89).`
  Creates systemwide TOTAL statistics across all servers (for `RSSTAT80`).
- `Max. number of parallel RFCs (only valid for RSSTAT83/88).`
  Maximum number of RFCs executed in parallel ("999" means a RFC is started simultaneously for all servers).

Options for reorganizing statistical data (for all servers)

- `Delete seq. statfile after cumulation if size > (default: 100Mb).`
  This parameter specifies from what file size the system should delete the statistics file. The statistics file is required for individual statistics and is therefore not deleted until the file size has passed a specified maximum file size. Of course, the file is only deleted if it was completely processed by `RSSTAT80` or `RSSTAT83`..
- `Max. no. of records cumulated per call (default: 20.000).`
  This is the maximum number of entries in the statistics file that can be processed by `RSSTAT80` or `RSSTAT83` in one session. This parameter is used to restrict the runtime of the collector.

Options for reorg of application statistic data (valid for all servers)

- `Delete appl. statfile after cumulation if size > (default: 30Mb).`
  This parameter specifies from what size the system should delete the application statistics file. Of course, the file is only deleted if it was completely processed by `RSSTAT88` or `RSSTAT89`.
- `Max. number of records cumulated per call (default: 20.000).`
  This is the maximum number of entries in the application statistics file that can be processed by `RSSTAT80` or `RSSTAT83` in one session. This parameter is used to restrict the runtime of the collector.

Data Archiving Monitor

Use

You can use the data archiving monitor to monitor and analyze information about writing of archive files (write jobs) and deletion of data from the database (delete jobs) during an archiving session.

Every archiving and delete job executed in the background in production mode sends information to the alert monitor. Archiving and delete programs started in dialog mode for test purposes only send information to the alert monitor, if the program is running on the server where the monitoring context *DataArchiving* was created.

Integration

The data archiving monitor is integrated in the alert monitor (transaction `RZ20`) and can be called by choosing *SAP CCMS Monitor Templates → Data Archiving*. You can monitor data archiving in the database monitor under *SAP CCMS Monitor Templates → Database → DataArchiving*.

Features

The data archiving monitor provides the following functions:

- A complete overview of all executed archiving objects
- Details about archiving and delete sessions (such as start time, runtime, size of the archive files, and number of archived data objects)
- A progress indicator of archive files processed
- Recognition of need for action (for example, open delete jobs) or errors (for example, I/O error when alerts triggered)
- Analysis of open alerts by displaying information about the executed jobs as well as their logs

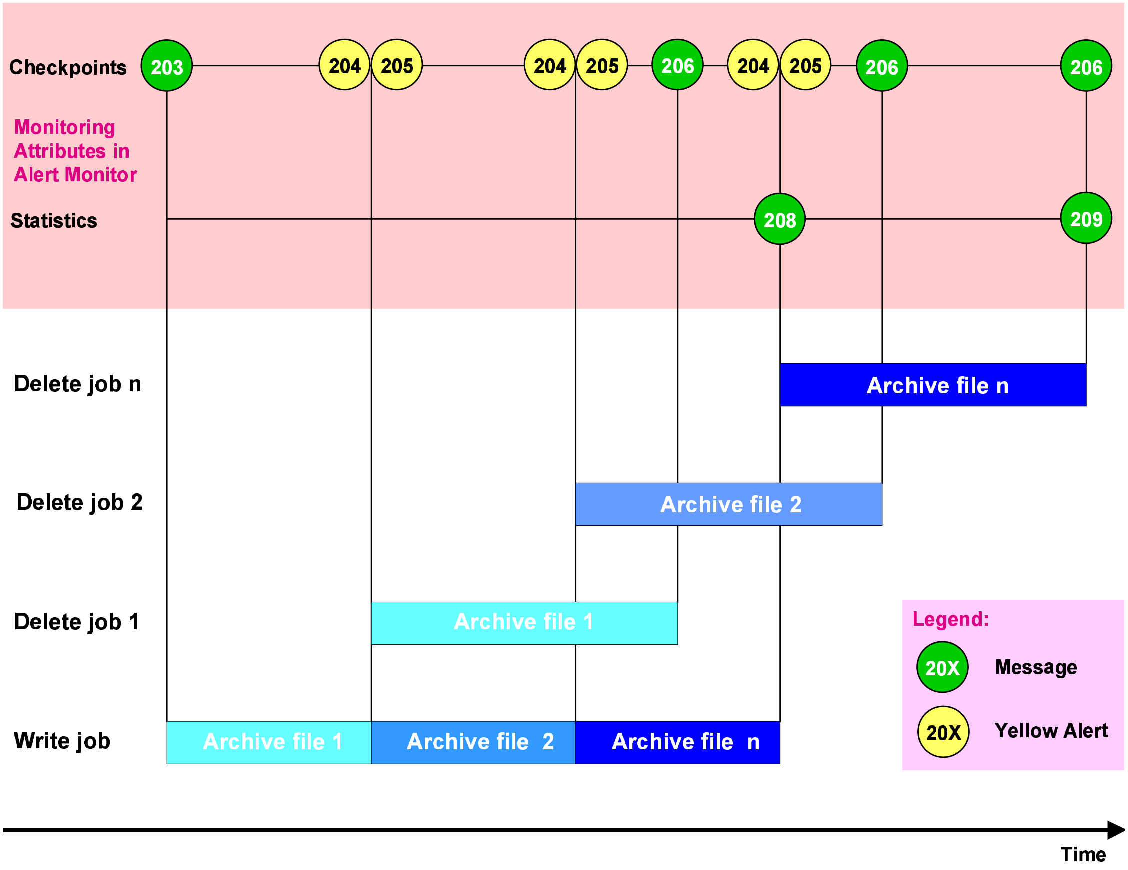
Activities

Every archiving and delete program based on the Archive Development Kit (ADK) generates a monitoring node (corresponding to the name of the archiving object) and the monitoring attributes *Checkpoints* and *Statistics* when first run.

The monitoring attribute *Checkpoints* collects important status messages and error messages (alerts). Once alerts are completed and there is no session that has errors or is incomplete for this archiving object (according to archive management), the monitoring attribute *Checkpoints* is automatically deleted along with all messages and alerts.

The monitoring attribute *Statistics* collects statistics about the last successfully completed archiving and delete sessions. The monitoring attribute *Statistics*, in contrast to the monitoring attribute *Checkpoints*, is not deleted.

The following graphic illustrates the timeline for a typical archiving session, including the messages and alerts for the monitoring attributes *Checkpoints* and *Statistics* displayed in the alert monitor. The messages are described in *Messages and Alerts*.


Checkpoints
203
204
205
204
205
206
204
205
206
206
Monitoring Attributes in Alert Monitor
Statistics
208
209
Delete job n
Archive file n
Delete job 2
Archive file 2
Delete job 1
Archive file 1
Write job
Archive file 1
Archive file 2
Archive file n
Legend:
20X
Message
20X
Yellow Alert
Time

Messages and Alerts

Messages

Messages are used to inform you of important status information, such as that an archiving session started; they do not require you do something. Data archiving includes the following messages (message class *BA*):

- `203` (Checkpoints)
  Session ...: Archiving started <Job: ... ID number: ...>
- `206` (Checkpoints)
  Session ...: File ...: Delete from DB finished <Job: ... ID number: ...>
- `221` (Checkpoints)
  Session ...: File ...: Move completed <Job: ... ID number: ...>
- `208` (Statistics)
  Sess. ... : WriteJob closed after appr. ... <total MB: ... DataObjects: ...>
- `209` (Statistics)
  Session ...: Delete jobs finished after an average of ... <no. of jobs: ...>

Yellow Alerts

Yellow alerts are automatically taken care of when the corresponding delete program is run. However, the "yellow alert" remains. Use the alert browser in the alert monitor to determine if there are any open alerts left. Data archiving includes the following yellow alerts:

- `204` (Checkpoints)
  Session ...: File ...: Write finished <Job: ... ID number: ...>
- `205` (Checkpoints)
  Session ...: File ...: Delete from DB started <Job: ... ID number: ...>

A yellow alert means a delete job still needs to be run. Therefore, you must either wait for the delete job to be automatically executed or manually start the delete job.

Red Alerts

A red alert means action is required. Analyze the error by starting the analysis method in the alert browser. The affected jobs are displayed. Goto the job log and fix the error. Reschedule the job using transaction **`SARA`**. Data archiving includes the following red alerts:

- Cannot open file to read (for delete sessions)
- Logical file name incompletely maintained or incorrect (for archiving and delete jobs)
- Cannot recreate file
- Cannot read a file header or the data objects from a file from the file system
- Cannot read a file header through ArchiveLink
- Cannot write to (new) archive file

- File is not an archive file
- Archiving object does not exist (archiving and delete jobs)
- Insufficient authorization (archiving and delete jobs)
- Cannot archive due to scheduled local currency changeover to euro (only for archiving jobs)
- Cannot find any files whose data is to be deleted from database
- Error generating a new management entry for a new archiving session
- Error creating a file

Additional Technical Information

When the system is started, the startup method *CCMS_DA_TREE* is called. This method builds the data archiving tree for all archiving objects. The monitoring context *DataArchiving* in archiving class *CCMS_DA_mcmtc* is used for this.

For displaying jobs that have triggered alerts, the standard system includes the analysis method *CCMS_DATA_ARCHIVING_ANALYZE*.

Archiving object developers are provided with the following function modules for cross-object data archiving control:

- *ARCHIVE_ENABLE_MONITORING*
- *ARCHIVE_DISABLE_MONITORING*
- *ARCHIVE_GET_CUSTOMIZING_MONIT*

You can deactivate the default active data archiving monitors in cross-archiving object Customizing in transaction *SARA*.

R/3 Accounting Interface

Definition

The R/3 accounting interface is an application programming interface (API) to statistics data on system resource consumption. The statistics data is collected by the CCMS (Computing Center Management System) workload monitor. Application developers can use the R/3 accounting interface to access this data. The collected data can be mapped to accounts and passed on to accounting software for accounting purposes.

Data is collected and summarized by the job COLLECTOR_FOR_PERFORMANCEMONITOR.

The R/3 accounting interface has two summary modes:

- User summary: Summarizes the data for each R/3 user
- Account summary: Uses the user summary data to generate data for all users assigned to an account

Use

System resources include the number of requests that are made, the CPU time that is used on the R/3 application servers, and other common time values that are collected by the CCMS workload monitor.

Requests can be:

- Dialog steps
- Background job steps
- Update requests
- Spool requests

Statistical data is available for each user for daily, weekly or monthly periods, and can be summarized for accounts.

You can select statistics data for one instance, or for the entire R/3 System, if you use TOTAL. The statistics data can be read for a specific client or for all existing clients.

From Release 4.5, you can individually switch off the creation of total weekly and monthly statistics or accounting statistics. For more information, see Note 127642 in SAPNet and Parameters for RSSTAT80/83 and RSSTAT87/88/89 [Seite 1018].

Function Modules for R/3 Accounting

The R/3 accounting interface consists of the following function modules. Both have an RFC interface, which means that they can be accessed from outside the R/3 System.

- `SAPWL_ACCNT_GET_DIRECTORY:` Creates a table of contents describing the available accounting data from the CCMS workload monitor
- `SAPWL_ACCNT_GET_SYSTEM_USAGE:` Reads, prepares and summarizes the available accounting data

See also:

[Installing and Setting up the R/3 Accounting Interface [Seite 1031]]
[Creating an Account [Seite 1032]]
[Reading the Accounting Data [Seite 1035]]
[Creating the Table of Contents [Seite 1033]]

Database Resources

Definition

The R/3 accounting interface provides information on the following database system resources:

Name	Explanation	Unit	Repository Definition
Application database access time	Time required to read and process application data	ms	SAPWLACCTS-A_DB_TIME
System database access time	Time required for R/3 internal loading of programs and user interfaces	ms	SAPWLACCTS-S_DB_TIME
Application data volume	Data that was read or processed by the application	KB	SAPWLACCTS-A_DATA_VOL
System data volume	Data that was read to load programs and user interfaces	KB	SAPWLACCTS-S_DATA_VOL

See also:

Operating System Resource [Seite 1029]
R/3 System Resources [Seite 1030]

Operating System Resource

Definition

The R/3 accounting interface provides information on the following operating system resource:

Name	Explanation	Unit	Repository Definition
CPU time	CPU time required for all requests	ms	SAPWLACCTS- CPU_TIME

See also:

Database Resources [Seite 1027]

R/3 System Resources [Seite 1030]

R/3 System Resources

Definition

The R/3 accounting interface provides information on the following R/3 system resources:

Resource	Explanation	Unit	Repository Definition
Dialog steps	Number of requests for all task types	-	SAPWLACCTS-STEPS
On-line dialog steps	Number of requests for task type *dialog*	-	SAPWLACCTS-DSTEPS
Update dialog steps	Number of processed update records	-	SAPWLACCTS-USTEPS
Background dialog steps	Number of processed background steps	-	SAPWLACCTS-BSTEPS
Response time	Total response time of all requests	ms	SAPWLACCTS-RESP_TIME
Wait time	Total wait time in the dispatcher queue (of all requests)	ms	SAPWLACCTS-QUEUE_TIME

See also:

Operating System Resource [Seite 1029]

Database Resources [Seite 1027]

Installing and Setting up the R/3 Accounting Interface

Procedure

The R/3 accounting interface is installed as part of the standard SAP System. No special actions are required during installation.

To Activate the R/3 Accounting Interface:

The workload statistics on the application servers must be active. SAP delivers the SAP System pre-configured with active statistics. This is indicated by the instance profile parameter "`stat/level = 1`" for the application systems in question.

1. To execute the report RSCOLL00, you must schedule job COLLECTOR_FOR_PERFORMANCEMONITOR to run hourly. Choose *CCMS* → *Jobs* → *Definiton*. Alternatively, call Transaction SM36.
2. The job can also be called SAP_COLLECTOR_FOR_PERFMONITOR (see Note 16083 in SAPNet). You must maintain table TCOLL as described in Note 12103 in SAPNet. To maintain tables, call Transaction SM31.

You must have maintained user master records. Users must be assigned to accounts. If a user is not assigned to an account, that user is managed under the account "No account".

Changing the Statistics Retention Time

You can display statistics data as daily, weekly or monthly statistics. You can decide how long the statistics are retained in the database. You can also define the data retention time in the CCMS workload monitor. To change the statistics retention time, proceed as follows:

1. Choose *CCMS* → *Control/Monitoring* → *Performance menu* → *Workload* → *Analysis*
2. Then choose *Goto* → *Parameters* → *Performance database*

You can change the retention times in the group box *Residence times of statistical data (for all servers).*

See also:

Creating an Account [Seite 1032]

R/3 Accounting Interface [Seite 1025]

Creating an Account

Use

The costs incurred by one or more users are invoiced to an account. An account can be set up for an individual user or a work group. An example of an account is a cost center.

Procedure

1. Choose *Administration* → *System administration* → *User maintenance* → *Users*.
2. Specify a user name and then choose *Create* or *Change*.
3. Choose the *Logon data* tab and specify the account in the field *Accounting number*.

   For example, the account can be a cost center. The account is stored in table USR02, field ACCNT.

   Normally the current user master record is used to assign users to accounts. You can set other assignments using a function module parameter.

   Users who do not have an account defined in their own or the specified user master record are assigned to the account “No account”. Users not assigned to the specified user master record are assigned to the user account “Unknown user”.

   Predefined R/3 System users, such as sapsys, sap* or sapr3, who are not assigned to the specified user master record, are assigned to an account with the same name. For example, user sapsys is assigned to the “account” sapsys.

See also:

Example 1: Using the User Master Record [Seite 1037]

Example 2: Not Using the User Master Record [Seite 1040]

R/3 Accounting Interface [Seite 1025]

Creating the Table of Contents

Use

You can create the table of contents using function module SAPWL_ACCNT_GET_DIRECTORY (see the table below). This function module returns an internal table containing descriptions of the data that is available.

Due to the host name enhancement in Release 4.6A, the input parameters for accounting modules have changed. Instance IDs are now used for internal administration of data (you do not need to know the IDs).

Parameter	**Input/Output**	**Repository Type**	**Description**	**Optional**
PERIODTYPE	SAPWLACCTP -PERIODTYPE	Input	Pre-selection for the time period (day, week, month) *Default*: All available time periods	Yes
INSTID	SAPWLSERV-INSTSHORT	Input	ID of the instance checked. This field should not usually be filled If an instance name is transferred to the parameter INSTANCE, the parameter INSTID is deactivated (except where the parameter equals "*")	Yes
INSTANCE	SAPWLSERV- NAME	Input	Full name of the instance checked (server_<SID>_Instance number) You can check either all instances or only certain instances. *Default*: All available instances (that is, the parameter is transferred empty)	Yes
DIRECTORY	SAPWLACCTD	Output	Internal table with data descriptions matching the pre-selection The instance ID is transferred to INSTID	No

SAPWLACCTD contains details about the time period, the data that is available, and the application server.

You can set a PERIODTYPE of D, W, or M (for day, week, month).

If you specify PERIODTYPE = W (for week) and INSTID = * (all instances), the system displays the following output:

STARTDATE	PERIODTYPE	INSTID
19991206	W	ID for instance A
19991206	W	ID for instance B
19991213	W	ID for instance A
19991213	W	ID for instance B

This output shows that there is data for the weeks beginning December 6, 1999 and December 13, 1999 for both instance A and B.

Reading Accounting Data

Use

You read accounting data using function module SAPWL_ACCNT_GET_SYSTEM_USAGE. This function module returns the accounting data for an account or a user name over a specified period of time. In addition, it can be used to collect data for one or more instances or clients in your R/3 System.

Due to the host name enhancement in Release 4.6A, the input parameters for the accounting function modules have changed. Instance IDs are now used for internal administration of data (you do not need to know the IDs).

Parameter	Input/Output	Repository Type	Description	Optional
PERIODTYPE	Input	SAPWLACCTP -PERIODTYPE	Time period (day, week or month)	No
INSTID	Input	SAPWLACCTP -HOSTID	ID of the instance, as in the module SAPWL_ACCNT_GET_DIRECTORY. *Default*: All available instances are transferred (either empty or "*") If an instance name is transferred to the parameter INSTANCE, the parameter INSTID is deactivated (except where the parameter equals "*")	Yes
INSTANCE	Input	SAPWLACCTD-INSTANCE	Full name of the instance checked (server_<SID>_Instance number) *Default*: All available instances are transferred (either empty or "*")	Yes
STARTDATE	Input	SAPWLACCTP -STARTDATE	Start of time period (date of day, date of the Monday in the week, date of the first day of the month)	No
CUMULATON _MODE	Input	SAPWLACCTP -CUMUL _MODE	Data is summarized by user name or account. *Default*: User names	Yes

CLIENT	Input	SAPWLACCTP -CLIENT	Client for whom the data is to be read. *Default*: All clients	Yes
USER _ACCOUNT _RELATION	Input	USRACCNTV	Table with a user-defined account, different from the current user master record. If the table is empty, the user master record is used (table USR02).	Yes
STATISTIC	Output	SAPWLACCTS	Table with summarized data	No

See also:

R/3 Accounting Interface [Seite 1025]

Example 1: Using the User Master Record

Use

This is an example of ABAP code that is used to read accounting data for a specific account.

The statistics of the current day are read from all the available application servers, using the user account assignment from the user master record.

Example

```
REPORT ZACCDEMO.
* read the accounting data using the user-account relation
* defined in the user master records
* table with cumulated statistics
DATA BEGIN OF ACCOUNT_STATISTIC OCCURS 10.
         INCLUDE STRUCTURE SAPWLACCTS.
  DATA END   OF ACCOUNT_STATISTIC.
* read the accounting data
CALL FUNCTION 'SAPWL_ACCNT_GET_SYSTEM_USAGE'
      EXPORTING
           PERIODTYPE                = 'D' "daily statistics
  "        HOSTID                    = "of all available servers
           STARTDATE                 = SY-DATUM "from today
  "        CUMULATION_MODE           = 'A' "cumulate data for accts
  "        CLIENT                    =
      TABLES
           STATISTIC                 = ACCOUNT_STATISTIC
  "        USER_ACCOUNT_RELATION     = "use user master records
      EXCEPTIONS
           UNKNOWN_PERIODTYPE        = 01
           UNKNOWN_CUMULATION_MODE   = 02
           WRONG_CLIENT_DESCRIPTION  = 03
           WRONG_STARTDATE           = 04
           NO_DATA_FOUND             = 05.
```

See also:

Example 1: Using the User Master Record

Example 2: Not Using the User Master Record [Seite 1040]

Example 2: Not Using the User Master Record

Use

This is an example of ABAP code for reading the accounting data for a specified account.

This example differs from Example 1 [Seite 1037] in that the statistics of the current day are read using the specified user account assignment. Data is cumulated from all available application servers.

Example

```
REPORT ZACCDEMO.
* read the accounting data for a defined account
* table with cumulated statistics
DATA BEGIN OF ACCOUNT_STATISTIC OCCURS 10.
         INCLUDE STRUCTURE SAPWLACCTS.
  DATA END   OF ACCOUNT_STATISTIC.
* table with defined accounts
DATA BEGIN OF USER_ACCOUNT_RELATION OCCURS 10.
         INCLUDE STRUCTURE USRACCNTV.
  DATA END   OF USER_ACCOUNT_RELATION.
* definition of three account relationships
"first relation
  USER_ACCOUNT_RELATION-MANDT = '001'.        "set the client
  USER_ACCOUNT_RELATION-BNAME = 'USER_01'.    "set the user name
  USER_ACCOUNT_RELATION-ACCNT = 'ACCOUNT_01'. "set the account
  "append new relationship USER_01 in client 001 <-> ACCOUNT_01
  APPEND USER_ACCOUNT_RELATION.
"second relationship
  USER_ACCOUNT_RELATION-BNAME = 'USER_02'.
  "append new relationship USER_02 in client 001 <-> ACCOUNT_01
  APPEND USER_ACCOUNT_RELATION.
"third relationship
  USER_ACCOUNT_RELATION-BNAME = 'USER_03'.
  USER_ACCOUNT_RELATION-ACCNT = 'ACCOUNT_02'.
  "append new relationship USER_03 in client 001 <-> ACCOUNT_02
  APPEND USER_ACCOUNT_RELATION.
* read the accounting data for the defined account
```

```
CALL FUNCTION 'SAPWL_ACCNT_GET_SYSTEM_USAGE'
      EXPORTING
           PERIODTYPE                 = 'D' "daily statistics
 "         HOSTID                     = "of all available servers
           STARTDATE                  = SY-DATUM "from today
 "         CUMULATION_MODE            = 'A' "cumulate data for accts
 "         CLIENT                     =
      TABLES
           STATISTIC                  = ACCOUNT_STATISTIC
           USER_ACCOUNT_RELATION      = USER_ACCOUNT_RELATION
      EXCEPTIONS
           UNKNOWN_PERIODTYPE         = 01
           UNKNOWN_CUMULATION_MODE    = 02
           WRONG_CLIENT_DESCRIPTION = 03
           WRONG_STARTDATE            = 04
           NO_DATA_FOUND              = 05.
```

Global Work Process Overview: Contents

System-wide Work Process Overview

Use

You can quickly investigate the potential cause of a system performance problem by checking the work process load. Using the systemwide work process overview, you can:

- Monitor work process load on all active instances **systemwide**
- Identify locks in the database (lockwaits)

Procedure

1. Call Transaction SM66 or choose CCMS → Control/Monitoring → Work process overview. The system displays the *Global Work Process Overview* screen, which includes all current work processes.

   To display the work processes for individual instances, *Goto* → *Local work processes*.

   You may need to scroll in order to see all the information displayed. You can also sort the information displayed.

See also:

Selecting Work Processes [Seite 1044]

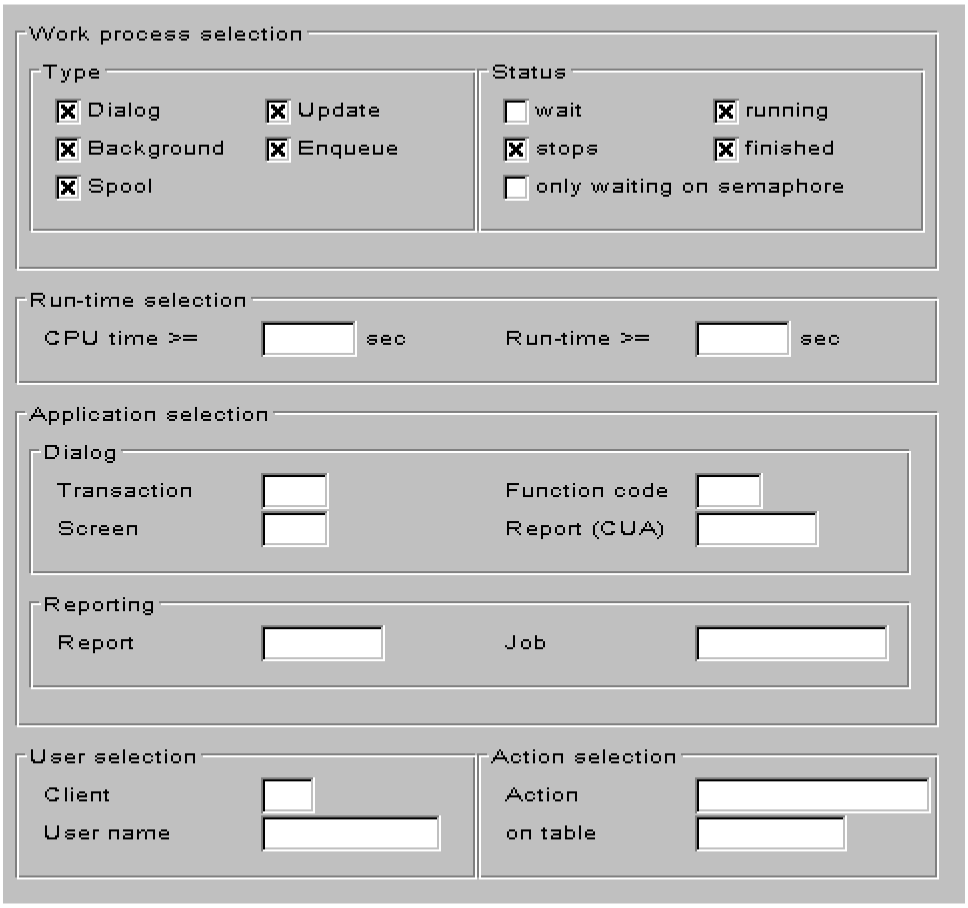
Detailed Work Process Information [Seite 1046]

Selecting Work Processes

To select the work processes to be displayed:

Select *Select process*.

The following dialog box is displayed:



Select the work processes and statuses you want to know more about. You can also get information on specific programs and users.

Type

Choose the work process type.

Status

You can select the work process statuses you are interested in.

Runtime selection

Use this option to select long-running work processes.

Application selection

Use this option to select requests for specific R/3 transactions.

Reporting

Use this option to select specific ABAP/4 programs.

User selection

You can investigate potential specific causes of a problem. If you suspect that a particular user is blocking work processes, enter the name of the program or user, then select *Continue* to filter the information.

See also:

Work Process Load Monitor: Overview [Seite 1043]

Detailed Work Process Information [Seite 1046]

Detailed Work Process Information

To display more detailed information about a work process:

Position the cursor on the instance, then choose *Choose*.

You can terminate the program currently running and debug it.

For background processes, additional information is available for the job currently running. You can only obtain this information, however, if you are logged onto the instance where the job is running, or if you have deselected *Display only abbreviated information, avoid RFC* under *Settings.* In any case, the job must still be running.

SAP Buffers

Purpose

Each SAP instance (application server) has its own buffers. These buffers are also known as client caches because they are implemented on the client, that is, the application server. SAP buffers occupy memory areas that are local to the work process, and in individual shared memory segments that can be accessed by all work processes. These memory areas are executed for the application server.

Some of the shared memory segments in an SAP System are grouped into one shared memory segment known as a pool. This is done to meet the operating system limits on the number of shared memory allocations per process. In most operating systems, you can allocate as many shared memory segments as required. The limits depend on the kernel configuration. The AIX operating system, for example, allows 10 shared memory segments per process.

SAP buffers store frequently-used data, and make this data available to the local application server instance. This helps to reduce the number of database accesses, the load on the database server (it does not need to be accessed repeatedly to obtain the same information), and network traffic. As a result, system performance is considerably improved.

The data that is buffered includes ABAP programs and screens, ABAP Dictionary data, and company-specific data. Typically these remain unchanged during system operation.

You can change, or tune [Seite 1071], the sizes of buffers to optimize performance for a particular hardware configuration. There are several ways to tune buffers. As there are many constraints to consider when change the buffer size, several difficulties may arise.

You can use table buffering to fine-tune applications, that is, some or all of the contents of infrequently changed tables can be held in local buffers.

SAP Buffers

Program Buffer	This buffer occupies a whole shared memory segment.
Generic Buffer **Screen Buffer**	These buffers are held in a shared memory pool. All work processes can access this pool.
Roll Area	Local work process buffers. Only one work process can access these buffers at a time.

See also:

Buffer Structure [Seite 1048]

Buffer Types [Seite 1052]

Working with the Buffers Monitor [Seite 1066]

Tuning Buffers [Seite 1071]

Buffer Components

Definition

An SAP buffer consists of the following parts:

Mode table	The mode table resides in shared memory and tells you which pool contains which shared memory areas. The mode table is part of the common information on the shared memory areas that are accessed by the work processes. For example, SAP Key 1 with Mode = 0, instructs the OS kernel to extract this buffer from the default pool and to allocate a unique shared memory segment. SAP Key 10 with Mode = pool size instructs the OS kernel to store the buffer specifically in pool 10. SAP Key 11 with Mode = -10 means that the buffer is located in pool 10.
SAP Global Management Table	A shared memory area that is allocated by the dispatcher during system startup. When semaphore protection is on, the SAP Global Management Table is addressed exclusively by **SAP Shared Memory Management**. This is a central agent that is found in each work process and that sets up a shared memory area for the local application server or instance. The SAP Shared Memory Management issues a call to the operating system (OS) when it creates a shared memory area. As a result, the **SAP key** is assigned to an **OS key**. The OS returns a unique identifier (**handle**) for the shared memory area, with which the SAP Shared Memory Management addresses the shared memory area that was created by the OS. All work processes in the SAP System can access the SAP Global Management Table. The handle can be accessed by all work processes.
Address Table	Every work process contains this table. Assigns virtual addresses to the physical addresses of the shared memory areas.
Shared Memory Objects	These include the buffers, for example.
Header	Contains information on the shared memory area (also called memory segment). If a write error occurs outside the segment area, then the uniformity of the header is destroyed. The control function of the SAP Management of Shared Memory checks the consistency of the headers.
ID	Identifies the memory area. The ID is assigned when a SAP Shared Memory Management user requests the memory area.

Storage Class	The memory class. Examples of memory classes: permanent (local), shared, roll, paging and short.
Subdivision	A mark for the requested area that can be referred to later when you release the memory area.
Size include header	Buffer size including the header.
Alignment	Alignment of memory areas in accordance with hardware constraints.

See also:

Buffer Synchronization [Seite 1050]

Buffer Types [Seite 1052]

Buffer Synchronization

The fact that each application server has its own buffers could result in data inconsistency across the various application servers (instances). To prevent data inconsistency, the SAP System uses **periodical buffer synchronization**, which is sometimes called **buffer refresh**.

Every modifying action on buffered data, which could also be buffered by other application servers, produces synchronization telegrams that are written to a central DB table (DDLOG). Every application server periodically reads the telegrams written since the last synchronization, and checks its buffers for data to be refreshed.

Buffer synchronization can be controlled by changing the following parameters in the instance profile [Seite 266]:

- `rdisp/bufrefmode = sendon | sendoff, exeauto | exeoff`
- `rdisp/bufreftime =` (in seconds, time between two synchronization)

During the period between two refreshes, an application server may read data from its buffers while they are being modified by another application server. For this reason, no important volatile customer data should be buffered in the SAP buffers.

Examples of buffered data:

- Table TSTC (SAP transaction codes)
- Table T100 (error messages)
- ABAP executables
- Screens

Buffer synchronization is required only for distributed SAP Systems when more than one application server (instance) is used. If your SAP System utilizes only one application server (instance), buffer synchronization is not needed. When the application server is restarted, all buffers are erased and dynamically reconstructed.

Before you use `tp` (SAP transport program) to import objects into a central instance (that is, only one instance in the whole SAP System), you should set the following parameter:

- `rdisp/bufrefmode = sendoff, exeauto`

If you set the paramter to '`exeoff`', the central instance does not read the DDLOG table. This means that any changes to repository objects in the database (that is written to using `tp`) are not updated in the SAP repository buffers. This may mean that the system displays syntax error messages for the ABAP programs that are affected.

The ABAP processor can detect whether a version of the ABAP program imported via `tp` is new, and reloads the program buffer. The SAP repository buffers still contain the old repository objects.

See also:

R/3 Repository Buffers (Nametab Buffers) [Seite 1053]

Buffer Types [Seite 1052]

Buffer Types

Definition

There are seven main groups of buffers [Seite 1047] found in the shared memory. For more information on each buffer type, see**:**

Repository Buffers [Seite 1053]

Table Buffers [Seite 1055]

Program Buffer [Seite 1057]

SAPgui Buffers [Seite 1058]

Roll and Paging Buffers [Seite 1059]

SAP Calendar Buffer [Seite 1060]

SAP Cursor Cache [Seite 1062]

Repository Buffers (Nametab Buffers)

Definition

The name table (nametab) contains the table and field definitions that are activated in the SAP System. An entry is made in the Repository buffer when a mass activator or a user (using the ABAP Dictionary, Transaction SE11) requests to activate a table. The corresponding name table is then generated from the information that is managed in the Repository.

The Repository buffer is mainly known as the nametab buffer (NTAB), but it is also known as the ABAP Dictionary buffer.

The description of a table in the Repository is distributed among several tables (for field definition, data element definition and domain definition). This information is summarized in the name table. The name table is saved in the following database tables:

- DDNTT (table definitions)
- DDNTF (field descriptions)

The Repository buffer consists of four buffers in shared memory, one for each of the following:

Table definitions	TTAB buffer	Table DDNTT
Field descriptions	FTAB buffer	Table DDNTF
Initial record layouts	IREC buffer	Contains the record layout initialized depending on the field type
Short Nametab	SNTAB buffer	A short summary of TTAB and FTAB buffers

The *Short nametab* and *Initial record layouts* are not saved in the database. Instead, they are derived from the contents of tables DDNTT and DDNTF.

When access to a table is requested, the database access agent embedded in each work process first reads the *Short nametab* buffer for information about the table. If the information is insufficient (for example, the `SELECT` statement uses a non-primary key) it accesses the *Table definitions* buffer and then the *Field descriptions* buffer. By reading the Repository buffers, the database access agent knows whether the table is buffered or not. Using this information, it accesses the table buffers (partial buffer or generic buffer) or the database.

The IREC buffer is read:

- When a `REFRESH` command is executed in an ABAP program
- At an `INSERT` command, when a record is created in the buffers before the data is inserted and the fields are initialized with the values found in IREC buffer

You can set the buffers mentioned above by editing the parameters in the instance profile [Seite 266].

See also:

Buffer Types [Seite 1052]

Typical Parameter Settings for SAP Buffers [Seite 1080]

Table Buffers

Definition

There are two kinds of table buffers:

- **Partial table buffers**
- **Generic table buffers**

Use

The table below displays these table buffers and their functions.

Buffer	Also known as	Function
Partial table buffer	*TABLP* *partial buffer* *single record table buffer* *single key buffer*	Stores single table entries, that is, one record with its field values.
Generic table buffer	*TABL* *generic buffer* *resident-table buffer* *100% buffer* *generic key buffer*	Stores a range of table entries, that is, a range of records with their field values. The generic table buffer can also store all the entries (records) in a table. This is known as resident (or full) buffering.

Whether a table is partially buffered, generically buffered, or fully buffered depends on its attribute settings. You can change the buffer attributes of a table using Transaction SE13. For more information, see Typical Parameter Settings for SAP Buffers [Seite 1080].

See also:

Displaying Table Buffers [Seite 1056]

Buffer Types [Seite 1052]

Displaying Table Buffers

Procedure

1. From the SAP initial screen, choose *Administration* → *System administration* → *Monitor* → *Performance* → *Setup/Buffers* → *Buffers* → *Goto* → *Current local data* → *Table buffers* → *Single record* **or** Generic key.

You should only buffer tables that you change infrequently. You specify whether or not a table should be buffered in Transaction SE13.

Changes to buffered information must be updated in the buffers of other application servers that are sharing that information. These updates can adversely affect performance. The more servers that need updating, the more expensive the update process.

If all operations are performed on a central server, you can deactivate the buffer update messaging service.

There are two profile parameters [Seite 247] that control the behavior of synchronization for the application server:

Parameter name	Recommended value
`rdisp/bufrefmode`	sendoff, exeoff (central system)
`rdisp/bufrefmode`	sendon, exeauto (client/server)
`rdisp/bufreftime`	60

Displaying Parameter Settings for the Current Instance

You define current parameters in the SAP instance profile [Seite 266]. Restart the instance to activate any changes. To display the parameter settings:

1. From the SAP initial screen, choose *Administration* → *System administration* → *Monitor* → *Performance* → *Setup/Buffers* → *Buffers* → *Goto* → Profile *parameters* → *Current*.

See also:

Fine-Tuning Buffers [Seite 1071]

Program Buffers

Definition

The following table displays the program buffer and its functions.

Buffer	Also known as	Function
Program buffer	*SAP executable buffer* *ABAP buffer* *PXA (Program Execution Area)*	Stores the compiled executable versions of ABAP programs (**loads**). The contents of this buffer are stored in tables D010L (ABAP loads), D010T (texts) and D010Y (symbol table).

The program buffer has a hash structure and supports LRU (**Least Recently Used**) displacement.

You can reconfigure the program buffer by adjusting its instance profile [Seite 266] parameters.

See also:

Buffer Types [Seite 1052]

Typical Parameter Settings for SAP Buffers [Seite 1080]

SAPgui Buffers

Definition

There are two kinds of SAPgui buffers:

- Presentation buffers
- Menu buffers

The following table shows the SAPgui buffers and their functions:

Buffer	Also known as	Function
Presentation buffer	*Screen buffer* *Dynpro buffer*	Stores the generated screens (DYNPRO loads). The presentation buffer is adjusted by changing its instance profile [Seite 266] parameters.
Menu buffer	*CUA buffer*	Stores objects from the SAPgui. For example, menus, pushbutton definitions. These objects are from tables D345T (CUA texts) and D342L (CUA loads). The buffer has directory structure and supports LRU displacement. The menu buffer is adjusted by amending its instance profile parameters.

See also:

Buffer Types [Seite 1052]

Typical Parameter Settings for SAP Buffers [Seite 1080]

Roll and Paging Buffers, Extended Memory

Definition

The roll and paging buffers are the preferred working area of the roll and paging areas for an instance (application server). The remaining area is located on disk as roll and paging files. The user context is stored in the extended memory and the roll area (when the job is “rolled out” of a work process). The paging area stores special data for the ABAP processor, while the extended memory stores a large portion of the internal tables of a program.

You set the roll and paging buffers, as well as the extended memory using the parameters in the instance profile [Seite 266].

See also:

Buffer Types [Seite 1052]

Typical Parameter Settings for SAP Buffers [Seite 1080]

SAP Calendar Buffer

Definition

The SAP calendar buffer stores all defined factory and public holiday calendars.

Calendars are stored in the database tables TFACS and THOCS.

The buffer has a directory structure. This means that if the shared memory is configured too small, only the required data is loaded; there is no LRU displacement of the contents of the buffer.

You can change the calendar buffer by editing the parameter in the instance profile [Seite 266].

See also:

Buffer Types [Seite 1052]

Typical Parameter Settings for SAP Buffers [Seite 1080]

Import / Export Buffers

Definition

Import / export buffers are used to store data that must be available to several work processes. They are used, for example, for the Available-To-Promise logic (ATP logic) in Logistics.

Integration

The system fills or reads the buffers using the ABAP command: `EXPORT TO/IMPORT FROM SHARED BUFFER`.

SAP Cursor Cache

Definition

The SAP cursor cache helps to improve system performance by reducing the number of parsing of SQL statements; it is database-dependent. The SAP cursor cache is only slightly different for Oracle, Informix and SAP DB. It is totally different for AS/400 and MS SQL Server.

There are two types of cursor caches:

- Statement ID cache
- Statement cache

Changing the SAP cursor cache parameter value in the default profile will affect other areas as well. You are therefore advised **not** to tune it without the recommendation of a qualified SAP expert.

Statement IDs and the Statement Analyzer

The source of each SQL statement in the SAP System (ABAP, DYNP, the C modules of the database interface) assigns an ID to its Open SQL / Native SQL etc. statement. The statement ID includes:

- Module name (report name)
- Statement number (line number)
- Timestamp (time of ABAP generation)

The statement ID provides an easy way to recognize statements. There may be different statement IDs for one statement (for example, different ABAP programs doing the same `SELECT`). The Statement Analyzer eliminates such duplicities. When it receives an SQL statement (in control block form), this database interface module checks if the statement is simple (for example, `SELECT * FROM T100 WHERE... =... AND... =...`), or complex (for example, `SELECT * FROM T100 WHERE... <... AND... >...`). If the statement ID is simple, the Statement Analyzer assigns a ‘normalized’ statement ID.

The analyzer is called by the RSQL or Open SQL interface. If it is able to assign a normalized ID, the original ID (if existing) is replaced.

See also:

SAP Cursor Cache for Oracle, Informix, SAP DB and DB2 for UDB [Seite 1063]

SAP Cursor Cache for AS/400 and MS SQL Server [Seite 1065]

Buffer Types [Seite 1052]

Typical Parameter Settings for SAP Buffers [Seite 1080]

SAP Cursor Cache for Oracle, Informix, Adabas and DB2 for AIX

Purpose

The procedure for Oracle is in principle similar to Informix and Adabas. There are two SAP cursor caches:

- **Statement ID cache**
- **Statement cache**

All the statement IDs are stored in the statement ID cache. Each statement ID in the statement ID cache is mapped to an entry in the statement cache. The total number of entries in the statement ID cache is five times that of the statement cache. Both the statement ID cache and statement cache adopt the LRU (Least Recently Used) displacement.

The statement cache is an array of SQL statement structures or strings.

```
SELECT * FROM T100 WHERE MSGNR = :A001
```

In this example, table 'T100' and field 'MSGNR' are defined. The host variable 'A001' uses Oracle syntax.

Input variables such as '?' are used for Informix and DB2 for AIX. The DBSL (Database SQL Library) assigns the actual values to the variables. The output values are given descriptors and are allocated to the output buffers by the DBSL.

The number of entries in the statement cache is determined by the instance parameter (this is not the case with DB2 for AIX), where `dbs/stmt_cache_size = 250` (default value). You should **not** change this value.

DB2 for AIX: The number of statement cache entries is defined at 290.

The memory space of each entry in the statement cache is allocated dynamically. The maximum size of each entry is 64KB.

Each statement cache entry is mapped to the corresponding cursor, which points to a context area in the shadow process. This context area contains the representation of the parsed SQL statement.

The statement ID cache, statement cache, and cursor cache are in each SAP work process. The context areas are in the shadow process. Each SAP work process owns a shadow process.

The SAP cursor cache is an additional layer before using the database cursors.

Process

Executing a Normal SQL Statement:

A normal SQL statement is a statement with statement ID (may or may not be normalized by the Statement Analyzer):

SAP Cursor Cache for Oracle, Informix, Adabas and DB2 for AIX

DBSL uses the statement ID and searches the statement ID cache for a match.

1. If the ID is found, the corresponding cursor is executed. The process is now finished.
2. If the ID cannot be found, it is inserted into the statement ID cache.
3. Using string comparison, the DBSL search the statement cache for an identical SQL statement structure.
4. If it is found, map the statement ID in the cache to this SQL statement structure, which is already mapped to an existing cursor. Execute the cursor. You have now completed the procedure.
5. If it is not found, parse the SQL statement and execute it. The corresponding SQL statement structure will also be created and inserted into the statement cache. It is mapped to the corresponding cursor. The statement ID in the statement ID cache is mapped to this SQL statement structure.

Executing a Dynamic SQL Statement

With dynamic SQL statements, no statement ID is created.

1. DBSL search the statement cache (using string comparison) to find an identical SQL statement structure or string.
2. If it is found, the corresponding cursor is executed. The process is now finished.
3. If an identical structure cannot be found, the SQL statement is parsed and executed. The SQL statement structure is created and inserted into the statement cache. This SQL statement structure is then mapped to the cursor that was created.

DB2 for AIX: A DB2 agent process is assigned to every work process. The agent processes have access to the DB2 Cursor Cache, in which the access information for each ‘prepared’ statement is stored.

SAP Cursor Cache for MS SQL Server

Purpose

For MS SQL Server, the process is different from the process for Oracle.

For MS SQL Server, DBSL transforms the statement ID to a stored procedure name. The stored procedure with this name is generated in the database. The set of stored procedures is shared by all SAP work processes.

If the statement does not have a statement ID (for example, a dynamic statement), the DBSL uses a temporary statement cache. Each entry in the temporary statement cache holds the statement string and its associated temporary stored procedure's name. Currently, each SAP work process owns a set of temporary stored procedures.

Process

Executing Normal SQL Statements

A normal SQL statement is a statement with statement ID (that may or may not be normalized by the Statement Analyzer):

1. DBSL creates the stored procedure name from the statement ID.
2. Executes the stored procedure.
3. If it works, the process is finished.
4. If it does not work, DBSL creates the stored procedure in the database and then executes it.

Executing Dynamic SQL Statements

A dynamic SQL statement is a statement without statement ID (such as a dynamic statement that is specified completely at runtime):

4. DBSL searches the temporary stored procedure cache to find the SQL statement and the name of the associated temporary procedure.
5. If it is found, the temporary stored procedure will be executed. The process is now finished.
6. If it is not found, the temporary stored procedure is generated in the database and executed. The statement string and the new temporary stored procedure's name are inserted into the temporary statement cache. This entry then maps to the temporary stored procedure.

See also:

SAP Cursor Cache for Oracle, Informix, Adabas [Seite 1063]

The Buffer Monitor

Purpose

With the CCMS buffer monitor you can analyze the state of the SAP buffers and evaluate their quality. You can use this information to determine the areas in which you should change buffer sizes to improve performance.

The buffer monitor gives you the following information for a selected server:

- Quality of the most important buffers
- Sizes of the most important buffers
- Call statistics (database activity)
- Memory usage
- Semaphore usage
- Table calls
- Usage and configuration of the roll file
- Usage and configuration of the paging file
- Usage of extended and heap memory
- Buffer resets
- Number of objects in the buffers

See also:

Buffer Monitor and Tune Summary [Seite 1067]

Checking Memory Usage [Seite 1084]

Displaying Table Buffers [Seite 1056]

Buffer Quality: Overview [Seite 1076]

Buffer Monitor and Tune Summary

To call the buffer monitor:

From the initial R/3 screen, choose *Tools* → *Administration* → *Monitor* → *Performance* → *Setup/Buffers* → *Buffers* (or call Transaction `ST02`)

The buffer monitor displays information about buffer and memory usage and load for the instance where the user is logged on. Statistics are compiled from the time the server is started.

The *Tune Summary* screen is divided into four parts:

- Buffer
- SAP memory
- Cursor cache
- Call statistics

Buffers

The first column of the tune summary shows the names of the buffers:

- The four R/3 Repository buffers
- The program, CUA, screen, and calendar buffers
- The table buffers

See also:

Repository Buffers (Nametab Buffers) [Seite 1053]

Hit Ratios

The hit ratios are displayed in percent.

The following diagram illustrates what is meant by hit ratios. A hit is when a object (such as a table, screen or program) in the buffer(1) is accessed. If the object has to be read from the database(2), the buffer access fails.

Buffer hit ratio = Buffer object reads / logical requests

Buffer Quality

Buffer Quality = saved database calls / (database calls + saved database calls)

The database interface can translate one logical request into several database calls.

Buffer Quality: Overview [Seite 1076]

Allocated Size

The allocated size is measured in KB. It is different from the **available buffer size** because part of the space is used for buffer management.

Freespace

Freespace is important for analyzing the buffer size. The space remaining in the buffer is displayed in KB and as a percentage of the available buffer size.

Number of Directories

Even if there is freespace in the buffer, objects may not always be loaded into the buffer because there are no more free directories. The buffer monitor displays the number of directories available for the buffer, and the number and percentage free. The buffer directories point to the location of the objects stored in the buffer.

Swapping

When a buffer has insufficient freespace or free directories, it has to **swap** objects out of the buffer in order to load a new object. The column *Swap* shows how many objects have been swapped out since system startup.

Database Accesses

When an object cannot be read from the buffer, the database has to be accessed. The number of **database accesses** is displayed in the last column on this screen.

If a critical situation occurs in a buffer, the data for that buffer (freespace/object swaps) is displayed in red.

SAP memory

The following information is displayed:

- The amount of space currently used in percent and in KB
- The maximum value (max. use) since system startup
- The amount of space used in shared memory and on the disk.
-

See also:

Roll and Paging Buffers [Seite 1059]

For more information:

For details about an individual memory area, select a line.

The *Detailed Analysis* windows (*Extended Memory* and *Heap Memory*) include functions for information about:

- **Quotas**

Memory allocation (sequence, size)

- **Ext. mem. blocks**

  Extended memory user

- **Current parameters**

  The instance profile settings for memory management

- **History**

  Memory usage over the course of several days.

SAP cursor cache

The R/3 System has a cursor cache that stores cursors for **SELECT** statements to avoid time consuming **PREPARE** processing.

This cache has a fixed size and cannot be changed.

SAP Cursor Cache [Seite 1062]

Call statistics

The *Tune Summary* screen displays the access statistics for all data either residing in the SAP pool buffers or the database.

The following table describes the information of the tune summary.

Table: Tune Summary

Screen	Information
First column	Contains the different kinds of statements that can be used to access a table. (**SELECT SINGLE**, **SELECT**, **INSERT**, **UPDATE**, **DELETE**).
Last line	Shows the total - or in the case of the hit ratios, the average - data of the call statistics.
Next column	Shows the hit rates for the **SELECT** statements for buffered tables. A hit rate for the other statements (**UPDATE**, **INSERT** and **DELETE**) is not displayed because these statements always have to be passed to the database.
ABAP Processor	Displays the number of logical requests to the buffered tables, and how many of them failed.

The subsequent columns display:

- The average database call time
- The number of affected rows.

Buffer Monitor and Tune Summary

> Table access failures are not the same as buffer access failures. The SQL statements SELECT SINGLE, INSERT, DELETE, and UPDATE can fail if the specified data record does not exist. A buffer access can "fail", if the table has not yet been loaded into the buffer.

Some logical requests cannot be satisfied by buffer access and require that the database be accessed.

Buffer history

Analyze the buffer history to find the correct buffer sizes.

To display the buffer history:

From the initial R/3 screen, choose *Tools* → *Administration* → *Monitor* → *Performance* → *Setup/Buffers* → *Buffers* → *Goto* → *Performance database* → *This server history*

The data displayed is similar to the columns in the *Tune Summary*.

The sizes of the buffers may be appropriate for the current system situation, but may have been too small for recent days. If so, you should resize the buffers even if the sizes are appropriate for the current situation.

See also:

Checking Memory Usage [Seite 1084]

Displaying Table Buffers [Seite 1056]

Tuning SAP Buffers

Definiton

It is beneficial to keep as much data buffered as possible. This helps avoid repetitive database accesses, for example from the network and Interprocess Communication (IPC), and maximize system performance.

Reduced performance is caused by

- Buffers set too small.

  The required data cannot be stored in the buffers. Instead, objects have to be swapped [Seite 1083] out of the buffers. This causes expensive database accesses.

- Buffers set too large

  Memory is wasted. Paging may occur if too much memory is taken from the operating system and allocated to the SAP buffers and database.

You should check regularly whether the buffer size is suited to your system requirements. Since buffers are crucial for the performance of the SAP System, **all buffers** (except the SAP cursor cache) should be adjusted [Seite 1085] to their optimal value.

The optimum size for each buffer depends largely on the specific configuration of the server, that is, the applications, the number of users working in each module, and so on. Therefore, it is difficult to specify values suitable for all configurations. The most important criterion for the correct buffer size is the buffer quality [Seite 1078].

You should not adjust buffers in cases where poor buffer quality [Seite 1077] is due to special circumstances, for example, object swaps in the program buffer in a system with a high level of development activity.

Buffer Tuning Checklist

If there is insufficient memory to set the appropriate buffer sizes, you should consider adding physical memory. However, since this could take some time, you may have to decide which buffers are most important and should be adjusted first using the existing memory.

As a guideline, the most important buffers are those that:

- Are responsible for good dialog performance
- Are used most frequently
- Require relatively few memory resources

These criteria give you the following priority list:

1. Repository buffers (nametab buffers) [Seite 1053]
2. Table buffers [Seite 1055]
3. Program buffers [Seite 1057]
4. Roll and page file buffers [Seite 1059]
5. SAPgui buffers [Seite 1058]

See also:

High Buffer Quality [Seite 1078]

How Much Operating System Paging is Acceptable? [Seite 1088]

Directory Space and Data Space [Seite 1089]

Adjusting Pool Sizes [Seite 1086]

Special Aspects of Tuning [Seite 1073]

Special Aspects of Tuning

Definition

Only transparent tables and pooled tables can be buffered. You should buffer tables that

- Only have read-only accesses
- Have **not** been modified.

Other tables should only be buffered if the write accesses occur very infrequently and the tables do not contain customer data. In the case of tables that are modified frequently, the additional processing required could cancel out any performance gains achieved by buffering.

Buffering types

Full (residential) buffering

Either the whole table or none of the table is stored in the buffer. This type of buffering is recommended (as a rule of thumb) for the following tables:

- Tables up to 30KB in size, and accessed frequently, but all accesses are read accesses.
- Larger tables where large numbers of records are frequently accessed. However, if the application program is able to formulate an extremely selective `WHERE` condition for these multiple accesses using a database index, it may be advisable to dispense with buffering. In this case, pooled tables should be converted to transparent tables.
- Tables where frequent attempts are made to access data not contained in the table, resulting in a “No record found“ message. With full buffering all records of a table are contained in the buffer, which means a faster response to indicate whether or not the table contains a record for a specific key can be displayed.
- Tables which are small and are subjected to large number of read accesses, but are rarely written to.

Generic buffering

When you access a record from the table, other records whose generic key fields correspond to this record are also loaded into the buffer. This type of buffering is recommended (as a rule of thumb) for the following tables:

- Client-dependent, fully buffered tables are automatically buffered generically (even if full buffering was selected in the table’s settings). The client field is the generic key.
- Language-dependent tables.

Single-record buffering (partial buffering)

Only records in a table, which is being accessed, are loaded into the buffer. This type of buffering is recommended (as a rule of thumb) for the following tables:

- Large tables where few records are accessed. The amount of records accessed should be between 100KB and 200KB.

The partial buffer also contains negative information. That is, if a record is accessed which does not exist in the database table, an empty record is stored in the buffer and a flagbyte is set to indicate the record does not exist.

See also:

SAP Buffers [Seite 1047]

Tuning SAP Buffers [Seite 1071]

Working with Call Statistics

Use

You can use the buffer monitor [Seite 1066] to help you decide which buffers to tune.

Procedure

1. To display the call statistics, from the initial screen, choose *Administration* → *System administration* → *Monitor* → *Performance* → *Setup/Buffers* → *Buffers* → *Detail analysis menu* → *Call statistics.* On the next screen, make a selection and choose Enter.
2. Sort the display according to the data records called. Position the cursor on an entry in the column *Calls* (under *DB activity*) and choose *Sort*.

Result

The *Changes* column indicates which tables are being changed most frequently. You can generally consider buffering tables with many read accesses but with fewer than **100** changes per day.

The system displays the number of changes made since the system was started. By checking the dates at the top of the screen, you can determine the number of days that have elapsed since system startup.

Therefore: **Changes per day = Total changes ÷ Total number of days in the selected time period**

You should also consider the table type. Only buffer tables of type *TRANSP* or *Pool* are important here. Ignore those that are of type *System* as these tables should not be changed.

See also:

Reasons for Poor Buffer Quality [Seite 1077]

Buffer Types [Seite 1052]

Working with the Buffer Monitor [Seite 1066]

Buffer Quality

Definition

You only can decide whether the buffer quality is poor after system startup, when the buffer loading phase is complete.

Once an instance is started, the SAP buffers begin to load. After significant initial activity, the buffer quality is soon greater than 90%. (The single key buffer may take a little longer, depending on system activity.)

To find out the buffers with poor quality, first check the system startup time. When the system is started, all buffers (except the program buffer which has a pre-load) are initially empty. Therefore, all objects that are accessed for the first time have to be read from the database and then loaded into the buffers.

If objects are not yet in the buffer, the hit ratio for the buffer will be low. The hit ratio increases from the time objects are loaded into the buffers. The rate of the increase depends on the workload in the system and is different for each buffer.

The single key buffer generally has the slowest increase in quality. For example, you do not have to worry about a single key buffer with a quality of around 80%, if the other buffer qualities are around 90%.

If buffer quality is poor, you should investigate what is happening to a buffer. However, poor buffer quality is not always due to a real problem. For example, transports into a system can reduce buffer quality. This is different for object swapping [Seite 1083]

See also:

High Buffer Quality [Seite 1078]

Reasons for Poor Buffer Quality [Seite 1077]

Reasons for Poor Buffer Quality

Definition

Poor buffer quality may be caused by any of the following:

Transports

Every time you transport new developed source into the system that is running the `tp`, you invalidate all entries in the program buffer and clear all other buffers (command `$SYNC`). Unlike the other buffers, the program buffer does not start empty, but is filled with invalid programs. It is very important to execute as **few transports** as possible and to collect data for collective transports.

Buffer resets

Setting up a buffer requires a large number of database and network accesses, and places a considerable load on the system. Therefore, only reset the buffers if inconsistencies occurred between the buffer and the database. This might happen, for example, if you update a buffered table with native SQL, that is, the database table is updated directly by bypassing the buffer. To reset the table buffers, enter `$TAB` in the command field. Use the command `$SYNC` to reset all the SAP buffers on the application server. These commands only affect the buffers of the application server on which the commands are entered. The buffers of the other application servers in the network are not affected.

Using the commands `$TAB` and `$SYNC` places an extremely large load on the system. In large systems, it could take up to one hour (depending on the access profile) for the buffer load to return to its original state. System performance is greatly impeded during this time.

Offline backups, SAP System reboots

Both of these actions require the SAP System to be shut down, and then started up again. All buffers of all servers are reset. Again, this takes a considerable amount of time.

Program development

Software development should not be done in a productive system, as development systems require a very large program buffer.

See also:

High Buffer Quality [Seite 1078]

High Buffer Quality

Definition

The following describes how the buffer quality of the SAP buffers should be in a SAP System with good performance.

Repository buffers (nametab buffers)

The qualities of the Repository buffers [Seite 1053] can reach 99.9% in a system that has been running for a few days. If the buffer quality is less than 95%, you should investigate the situation further.

Check the:

- Available freespace
- Available directories
- Object swaps
- Buffer history.

Program buffer

The program buffer contains the generated ABAP programs (load).

Critical effects on the program buffer

- **Transports**

  Every time a new source is transported into the system using the `R3trans` program, all entries in the program buffer are invalidated, and all other buffers are cleared (equivalent to the `$SYNC` command). Unlike the other buffers, the program buffer is not empty after transport, but is filled with invalid programs.

  Therefore, it is better to execute as few transports as possible and to collect data to do several transports at once. For more information, see Program Buffer [Seite 1057].

Presentation and calendar buffers

The quality should also be above 95%. Also check available freespace and the freespace history. For more information, see SAPgui Buffers [Seite 1058] and SAP Calendar Buffer [Seite 1060].

Table buffers

As mentioned above, the quality of the single key buffer increases very slowly from system startup. Therefore, bad quality (< 90%) should only be of concern if there is no freespace left in the buffer.

The quality of the generic key buffer should be greater than 95% and can be up to 99%. For more information, see Table Buffers [Seite 1055]

See also:

Roll and Paging Buffers [Seite 1059]

SAP Cursor Cache [Seite 1062]

Typical Parameter Settings for SAP Buffers

Repository Buffers (nametab buffers)

Table definition buffer (TTAB)	For the TTAB buffer, the size of the management part and the user data cannot be adjusted separately. Both are managed by the parameter **rsdb/ntab/entrycount**, which specifies the number of directory entries in the buffer. Each entry has a size of about 0.17 KB. This parameter also specifies the number of directory entries for the other three Repository buffers [Seite 1053].
Field description buffer (FTAB)	The number of directory entries is twice the number specified by the parameter **rsdb/ntab/entrycount**. Each entry requires about 0.039 KB in the management part of the buffer. The size of the data portion in KB is determined by **rsdb/ntab/ftabsize**.
Short nametab buffer (SNTAB)	The number of directory entries is twice the number specified by the parameter **rsdb/ntab/entrycount**. Each entry requires about 0.039 KB in the management part of the buffer. The size of the data portion in KB is determined by **rsdb/ntab/sntabsize**.
Initial record buffer (IRBD)	The number of directory entries is twice the number specified with the parameter **rsdb/ntab/entrycount**. Each entry requires about 0.039 KB in the management part of the buffer. The size of the data portion in KB is determined by **rsdb/ntab/irbdsize**.

The following table summarizes the buffer sizes that can be allocated, along with the corresponding profile parameters. It also contains the number of the pool containing the buffer:

Table: Buffer Capacity

Buffer	Management Header (Size in KB)	User Data (Size in KB)	Key	Pool
TTAB	0.17 KB × rsdb/ntab/entrycount		42	40
FTAB	0.039 KB × 2 × rsdb/ntab/entrycount	rsdb/ntab/ftabsize	43	40
SNTAB	0.039 KB × 2 × rsdb/ntab/entrycount	rsdb/ntab/sntabsize	45	40
IRBD	0.039 KB × 2 × rsdb/ntab/entrycount	rsdb/ntab/irbdsize	44	40

Thus, the size of the shared memory pool is controlled by the parameter **ipc/shm_psize_40**. For more information, see Repository Buffers [Seite 1053]

Table Buffers

Generic key (resident) buffer (TABL)

The number of directory entries (one for each resident table or generic area) is specified by `zcsa/db_max_buftab`. The size of the data space in **bytes** is defined by `zcsa/table_buffer_area`. The placement of the buffer can be controlled with `ipc/shm_psize_19` and is usually set to `10` (pool 10). The parameter `zcsa/exchange_mode` should not be changed. Keep its default value **off.**

Single key (partial) buffer (TABLP)

The number of directory entries (one per table) is specified by `rtbb/max_tables`. The size of the data space in **KB** is defined by `rtbb/buffer_length`. The placement of the buffer can be controlled with `ipc/shm_psize_33` and is usually set to **0,** which means that it is not in a pool. The parameter `rtbb/frame_length` specifies the size of one frame in **KB** and should always be kept at its default value of **4.** For more information, see Table Buffers [Seite 1055]

Program Buffers

The size of the program buffer can only be specified by one parameter, `abap/buffersize` which defines the size in **KB**. The number of directory entries is calculated from this parameter. The placement of the buffer can be controlled with `ipc/shm_psize_06` and is usually set to **0**, which means that it is not in a pool. For more information, see Program Buffers [Seite 1057].

SAPgui Buffers

Screen buffer (PRES)

The size of the directory, that is, the maximum number of screens is specified by `zcsa/bufdir_entries`. The total size of the buffer in **KB** is defined by `zcsa/presentation_buffer_area`, which includes the space for the directory.

Control the placement of the SAPgui buffers using parameter `ipc/shm_psize_14`. This parameter is usually set to `-10`, which means that it is in pool 10.

CUA buffer

The parameter `rsdb/cua/buffersize` defines the total size of the buffer in **KB.** The number of directory entries is calculated as: `total size / 2KB`. The placement of the buffer can be controlled with `ipc/shm_psize_47` and is usually set to `-40`**,** which means that it is in pool 40. For more information, see SAPgui Buffers [Seite 1058].

Roll and Paging Buffers

The parameters `rdisp/ROLL_SHM` and `rdisp/PG_SHM` are used to allocate the roll and paging buffer in **8KB** blocks.

This buffer is normally placed outside a pool. To place the buffer inside a pool, set the parameters `ipc/shm_psize_08` and `ipc/shm_psize_09`. For more information, see R/3 Roll and Paging Buffers [Seite 1059].

SAP Calendar Buffer

The size of the calendar buffer in **bytes** is specified by the profile parameter `zcsa/calendar_area`. For more information, see SAP Calendar Buffer [Seite 1060].

See also:

Note 103747 in SAPNet.

Special Aspects of Tuning [Seite 1073]

Call Statistics [Seite 1075]

Reasons for Poor Buffer Quality [Seite 1077]

Buffer Types [Seite 1052]

Object Swapping

Use

Swapping occurs when the buffer is full, and the SAP System has to load additional objects into the buffer. Objects in the buffer that were used the least recently are removed. In this context, the term “swap” means the objects removed from the buffer are lost and cannot be replaced until a new database access is performed (replacing what was lost).

There are two possible reasons for swapping

- There is no space left in the buffer data area

  The buffer is too small. You should increase the buffer size [Seite 1085].

- There are no directory entries left

  Although there is enough space left in the buffer, no further objects can be loaded because the number of directory entries is limited.

For the Repository buffers (nametab buffers) [Seite 1053], the number of directory entries is determined by the number of entry counts, all other buffers have there own parameter for entries.

The parameter **`abap/programs`** is no longer used. You do not have to change it if you want to increase the program buffer. For all other buffers you have to change the number of directory entries accordingly to the change in buffer size.

See also:

Tuning Buffers [Seite 1071]

Buffer Quality [Seite 1076]

Profiles [Seite 240]

Checking Memory Usage

Use

Before you change the size of a buffer, you should determine how much memory the buffer is currently allocated. This is important because the memory that is allocated to buffers should not significantly exceed the size of the physical memory, otherwise operating system swapping will occur.

Most server memory resources are used by the:

- Operating system (usually around 30 MB)
- Database system (See the database documentation)
- SAP System

Procedure

To check the memory usage, from the initial screen, choose *Administration* → *System administration* → *Monitor* → *Performance* → *Setup/Buffers* → *Buffers* → *Goto* → *Current local data* → *Additional functions* → *Storage*.

You should allocate as much memory as possible to ensure that good buffer quality.

It is usually more effective to allocate memory to the SAP buffers than to the database buffers. However, once the buffers are optimized (that is, no more object swaps occur), it does not make sense to increase buffer size further, since this will not improve performance.

See also:

[How Much Operating System Paging is Acceptable? [Seite 1088]](#)

[Directory Space and Data Space [Seite 1089]](#)

Changing Buffer Sizes

Use

You change buffer sizes by adjusting their profile parameters [Seite 247].

Procedure

Use the CCMS profile maintenance [Seite 278] tool to change these parameters.

Result

The new buffer sizes (that is, the change in the profile parameters) take effect when you restart the system. When you use the CCMS profile maintenance tool, you should also activate the profiles [Seite 287].

Adjusting Pool Sizes

Use

Many of the SAP buffers [Seite 1047] are contained in shared memory pools. When the sizes of these buffers are changed, it is necessary to adjust the size of the corresponding pool as well.

The CCMS profile maintenance [Seite 278] tool makes these adjustments automatically.

Procedure

To display information about the shared memory segments and the pools, from the initial screen, choose *Tools* → *Administration* → *Monitor* → *Performance* → *Setup/Buffers* → *Buffers* → *Goto* → *Current local data* → *Additional functions* → *Storage* → *Shared memory detail*.

You can change the pools and the shared memory segments with parameter **ipc/shm_psize_<two-digit shared-memory id>**.

You can change both the size and the location of the shared memory segment. The shared memory regions with the keys 10, 20, and 40 are used as shared memory pools by the SAP System. The size of these pools is determined by the values of the system profile parameters ipc/shm_psize_10, _20, and _40.

ipc/shm_psize_10 =12000000

This parameter sets the size of shared memory pool 10 to 12,000,000 bytes. The shared memory pools must be large enough to hold the shared memory regions that belong to them. If you change the size of a region that belongs a pool, you must also change the size of the pool. You must also change the size of a pool if you add a new shared memory region to it.

For all other keys, this parameter can be used to define the position of the shared memory region. In this case the shared memory parameters have the following format:

ipc/shm_psize_nn = value

where **nn** is the SAP shared memory key and *value* is the attribute of the shared memory region.

Shared memory parameters

The SAP System uses the default disposition (in pool or directly allocated) of the shared memory region.

Table: Assignment of pools to keys

Object key	Pool
11 – 19	10
21 – 29	20 (not used currently)
41 - 49	40

All other objects are not located in a pool, as specified by the default settings.

A shared memory parameter `ipc/shm_psize_nn` can have the following values:

The value 0	The shared memory region is removed from its standard pool (if any) and is allocated directly by the operating system. `ipc/shm_psize_19 =0` removes the 100%-resident table buffer from its default pool 10. If your host operating system does not limit the number of shared memory allocations, you can remove shared memory regions from pools. Removing a region from a pool simplifies the maintenance of the system profile because you no longer need to adjust the size of a pool when you adjust the size of a region. If a region does belong to a pool, you must change the pool size in accordance with changes that you make to the shared memory regions in the pool.
A negative value	The shared memory region is assigned to the shared memory pool whose shared memory key is the same as the value. `ipc/shm_psize_33 =-10` adds the partial table buffer to shared memory pool 10.

Shared memory parameters other than those for keys 10, 20, and 40 should not have values greater than 0. Otherwise, the SAP System treats the shared memory region as a pool.

The default settings for the shared memory parameters will not be automatically found in the profile. They will only appear if they have been inserted manually, either to confirm or to change the default setting.

See also:

[Tuning SAP Buffers [Seite 1071]]

[Typical Parameter Settings for SAP Buffers [Seite 1080]]

[Special Aspects of Tuning [Seite 1073]]

How Much Operating System Paging is Acceptable?

Definition

How much operating system paging is acceptable depends largely on the database server and the application servers.

On the database server, the I/O activity to the disks is normally quite high due to accesses to the database. Additional operating system paging should therefore be avoided to prevent an additional bottleneck. On the application server, disk accesses are usually less frequent. Therefore, operating system paging using a large buffer can be more efficient than a buffer which is too small which would lead to multiple database accesses and increased network traffic.

Directory Space and Data Space

Definition

For most buffers [Seite 1047] you can adjust the number of directory entries and the size of the user data area separately. SAP recommends that you increase both components, since an increase in one component may lead to a bottleneck in the other.

Microsoft Management Console: Windows NT

Use

The Microsoft Management Console (MMC) provides a common framework for system management. It allows various tools that were previously implemented separately to be integrated in a common user-interface enabling centralized system management. The MMC itself does not offer any management functionality, however it provides an environment for incorporating and accessing tools. Tools are integrated in the MMC in the form of **snap-ins** that allow standardized access to functions.

SAP has developed the R/3 Manager snap-in to allow the R/3 System to be monitored or started and stopped centrally from the MMC. The snap-in should be used instead of the old Service Manager [Seite 34] that is still available in the system. SAP recommends the use of the MMC because it simplifies system administration and provides many new features.

Features

The MMC offers the following features:

- You can monitor and control any number of R/3 Systems and application servers with a single tool as long as they are running under Windows NT.
- Once you have configured the console, you can save the configuration in a file and easily forward it to other users.
- The dual DCOM interface ISAPControl of the new start service lets a third party use the functionality, for example, from C, Java or Visual Basic, and you can integrate it into your own tools or scripts without having to use SAP proprietary mechanisms.
- The MMC allows you to:
  - Display all R/3 trace files, the system environment and SAP environment
  - Display the most serious alerts in the alert tree (Transaction RZ20)
  - Display the current status tree (Transaction RZ20)
  - Display and acknowledge current alerts (Transaction RZ20)
  - Display the R/3 syslog for a functioning, offline or malfunctioning R/3 System (Transaction SM21)
  - Display the R/3 process overview (Transaction SM50)
  - Display the queue statistics (dpmon)
  - Start or stop all NT application servers of a system with a mouse-click
  - Restart the service without having to restart the application server
  - Log on to an application server
  - Start analysis tools in R/3 for nodes in the alert tree
  - Start the following third-party tools to manage an application server, if they are available:
    - Telnet
    - Windows Terminal Server Client

- pcANYWHERE
- Computer Management (only for Windows 2000)

See also:

Starting the MMC [Seite 1092]

Layout of the R/3 Manager Snap-In [Seite 1093]

Context Menu of the R/3 Manager Snap-In [Seite 1095]

Configuring MMC for Active Directory Services [Seite 1099]

Starting the MMC

Use

There are three ways in which you can start the MMC:

- From the NT *Start* menu
- At the command prompt
- With a configuration file

Choose the method that best meets your requirements.

Procedure

NT Start Menu

Normally you start the MMC from the NT *Start* menu:

Choose *Start* → *Programs* → *SAP System Management Console*

The MMC opens. When you expand the *SAP R/3 Systems* node, all the instances running on the current host are displayed. If you want to change the configuration so that the instances on other hosts of your system are also displayed, follow the instructions given in Changing the Configuration of the MMC [Seite 1097].

Command Prompt

To start the MMC at the command prompt enter the following:

```
mmc.exe <MMC_configuration_file>
```

Where `<MMC_configuration_file>` is the name of a previously stored MMC snap-in configuration file. Use this method to start the MMC, if you have changed the configuration of the console and saved this in a file.

See also: Changing the Configuration of the MMC [Seite 1097].

Alternatively, you can specify an instance:

```
mmc.exe <MMC_configuration_file > -SAP_R/3_MANAGERHOST <instance _name>
```

Where `<instance_name>` is the instance that you want to monitor, for example, p18357_BIN_53. If other instances are already specified in the `<MMC_configuration_file>` file, the instance entered here will be added to the list.

Configuration File

If you have saved various configurations of the MMC, you can start the console simply by double-clicking on the appropriate configuration file.

See also:

Layout of R/3 Manager Snap-In [Seite 1093]

Context Menu of the R/3 Manager Snap-In [Seite 1095]

Layout of R/3 Manager Snap-In

Use

The R/3 Manager snap-in for the MMC provides essential information for monitoring the R/3 System and offers functions to start, stop and restart the system. All the information and functions available for R/3 can be accessed in the same way as the functions of other tools incorporated in the MMC. Therefore, before you begin working with the R/3 snap-in, it is useful to gain a general overview of the MMC layout and standard methods of accessing information and functions.

The MMC presents information on two panes of a window. The pane on the left is the scope pane. It displays available information in a tree structure that can be expanded and compressed. The pane on the right is the result pane. It shows detailed information about any item selected in the scope pane.

Features

When the MMC has been started, you can display the R/3 Manager snap-in by expanding the *Console Root*. The snap-in consists of a root node, *SAP R/3 System*, under which various R/3 Systems are listed. Under each of these systems, you can display individual instances. Each of the listed instances allows you to navigate to sub-nodes that provide detailed information about the system. The sub-nodes *Process List*, *Current State* and *Open Alerts* are always displayed. The additional nodes, *Syslog*, *Queue Statistic* and *WP Table*, are only available in *Expert user mode.*

The system icons and the nodes for instances, processes and alerts are displayed in different colors, depending on their state. The colors mean the following:

Gray:	Unknown state, only outdated values, system is offline
Green:	Error-free
Yellow:	Critical
Red:	Error

The R/3 snap-in periodically updates *Process List* information at a set time interval. Only the most critical state is passed on from the *Process List* to the parent instance node above it. The nodes at the R/3 System level contain all the states of the instance nodes. For example, if two instances are green, one is red and three are gray, the system node displays all 3 colors and thus reflects the states of all the instances.

To access detailed information about your R/3 system, you have to fully expand one of the R/3 instances listed in the MMC scope pane and then select an available item, for example, *Open Alerts.* Once an item has been selected, the corresponding information is displayed on the result pane. Usually you can navigate from here to further levels of detail or other categories of system information. The following table summarizes the available options:

Selected Item	Information Displayed

Process List	Displays the processes that have been started by the start service and their state.
Current Status	Displays detailed information on the current state of the system (analogous to transaction RZ20). It allows you to navigate to different categories of system information and levels of detail.
Open Alerts	Displays the most serious alerts in the system (analogous to transaction RZ20).
Syslog	Displays the system log, even if the R/3 system is not functioning properly or is offline (expert mode only).
Queue Statistic	Displays the state of the R/3 queues (expert mode only).
WP Table	Displays information on work processes (analogous to transaction SM50). The R/3 System has to be started, but does not have to be functioning properly as shared memory of the R/3 System is accessed (expert mode only).

You can choose to display the information you require in different ways using the *View* option. For example, the options *Large icons*, *Small icons* or *List* allow you to change the size and format of the display. Note that for the nodes *Alerts*, *Syslog*, *Queue Statistic*, *WP Table*, only the option *List* makes sense and is therefore the automatic default setting.

You can update the information displayed at any time using the *Refresh* option that is available on the context menu.

See also:

Context Menu of the R/3 Manager Snap-In [Seite 1095]

Starting the MMC [Seite 1092]

Context Menu of the R/3 Manager Snap-In

Use

The R/3 Manager snap-in includes context menus that offer a number of useful functions. In particular, it enables you to start and stop individual instances or the entire system simply by choosing the appropriate option. The context menu that is available differs, depending on the item that is selected in the scope pane.

Features

You can access the context menu of the R/3 Manager snap-in by selecting an item in the MMC scope pane and pressing the right mouse button. The context menu offers the following options:

Menu Option	Explanation	Scope Pane Node
Start	Starts the R/3 instance that is selected. If an R/3 System is selected, the entire system is started.	R/3 System or R/3 instance
Stop	Stops the R/3 instance that is selected. If an R/3 System is selected, the entire system is stopped.	R/3 System or R/3 instance
Logon	Logs on to the selected R/3 instance	R/3 System
All Tasks → View Start Profile	Displays the start profile	R/3 System
All Tasks → View Trace File	Displays the start service trace files	R/3 System
All Tasks → View Developer Traces	Displays developer trace files (e.g. dev_disp). provided that the user has the authorization. (*Expert User Mode* only)	R/3 System
All Tasks → View Environment	Displays the SAP environment. (Subdivided according to the system environment when the system is booted, the SAP Registry environment and the resulting total environment). (*Expert User Mode* only)	R/3 System
All Tasks → Restart Service	Restarts the start service of a selected R/3 instance without stopping the instance. This is useful, for example, to activate changes that have been made to profiles. (*Expert User Mode* only).	R/3 System

Context Menu of the R/3 Manager Snap-In

All Tasks → Telnet	Starts Telnet if it has been installed.	R/3 instance
All Tasks → Terminal Server	Starts the Terminal Server if it has been installed.	R/3 instance
All Tasks → pcANYWHERE	Starts pcANYWHERE if it has been installed.	R/3 instance
All Tasks → Manage	Adds the Computer Management snap-in for Windows 2000	R/3 instance
Analyse	Starts the analysis tool in R/3	Some subnodes of *Open Alerts* or *Current Status*
All Alerts	Displays all open alerts of a selected node and its subnodes in a separate popup. The popup allows the acknowledgement of alerts.	*Open Alerts* or *Current Status*
Properties	Displays technical data on the selected item	R/3 System R/3 instance

Take care when you select an instance that you want to start or stop. If the dialog instance and central instance are running on the same host, you can only distinguish between the two on the basis of their instance number. For example, the names of the central instance and dialog instance displayed could be: BIN p18357 53 and BIN p18357 54.

A useful feature of the MMC allows you to create subdirectories in the scope pane using the context menu of the *Console Root*. By choosing the option *New → Folder* you can create subdirectories under *Console Root* or under other directories already created. The root node of the SAP snap-in can also be moved simply by using drag and drop. This enables you to create any complex structure you wish.

See also:

Layout of R/3 Management Snap-In [Seite 1093]

Starting the MMC [Seite 1092]

Changing the Configuration of the MMC

Use

You can change the configuration of the R/3 Manager snap-in on the MMC to meet your requirements. In particular, you can extend the list of systems and instances displayed so that you can monitor all systems and instances from a single console.

Procedure

To extend the list of instances displayed in the MMC, do the following:

1. Open the MMC. Choose *Start* → *Run* from the NT main menu and enter `MMC`.

   The MMC window opens.

2. Select *Console root* and choose *Console* → *Add/Remove Snap-In.*

   The *Add/Remove Snap-In* dialog box opens.

3. Choose *Add* and then select the *SAP R/3 Manager* snap-in. Confirm your selection with *Add.*

   The *General Settings* dialog box opens.

4. Select the options you require:

Options	Select *Use R/3 instance fix list*
Auto Refreshment	Enter the periods in which you want system information to be updated automatically
Options	Select *Expert user mode* if you want to access more detailed system information later when you start working with the R/3 snap in.
	Select *Always show local R/3 instances,* if you want the instances installed on your local machine to be displayed.

5. Choose *Next* to close the *General Settings* dialog box.

   The *Fixed Server List* dialog box opens.

6. Create a list of the instances you want to monitor from the R/3 snap-in:

   a. In the *System* field, enter or select the name of the R/3 system that the instance belongs to.

   b. In the *Instance* field, enter or select the name of the host on which the instance is running, and the instance number, for example, `p18357 53`

   c. Choose *Add* to include the instance in the *Fixed Instances List*.

      An entry in the instance list looks like this:

      `BIN p18357 53`

   d. When you have entered all the instances you want to monitor, choose *Finish*.

Changing the Configuration of the MMC

You have now changed the configuration of the R/3 snap-in for the MMC. You can save the configuration in a file when you exit the MMC. Whenever you need the newly created configuration, you can start it simply by double-clicking on the saved file.

See also:

Starting the MMC [Seite 1092]

Layout of R/3 Manager Snap-In [Seite 1093]

Context Menu of the R/3 Manager Snap-In [Seite 1095]

Configuring MMC for Active Directory Services

Use

The MMC can be configured so that its structure and some of the information it presents are taken directly from the *Windows 2000 Active Directory*. The *Active Directory* is an extensive hierarchical inventory of all the objects available in a network. Special services make it possible to access and manipulate the information stored in the directory whenever required.

Configuring the MMC on the basis of the *Active directory* has the following advantages:

- The MMC reads information on R/3 systems and instances directly from the *Active Directory* and is therefore always up-to-date.
- The MMC presents additional information that is stored in the *Active Directory*, for example, detailed up-to-date technical data on systems or instances.

The MMC can be configured to access LDAP directories other than the *Active Directory*. In this case, additional instructions have to be observed.

Prerequisites

The R/3 System must be configured appropriately to support LDAP services.

Procedure

6. Start the MMC [Seite 1092].
7. Select *Console root* and choose *Console* → *Add/Remove Snap-In.*

   The *Add/Remove Snap-In* dialog box opens.
8. Choose *Add* and then select the *SAP R/3 Manager* snap-in. Confirm your selection with *Add.*

   The *General Settings* dialog box opens.
9. Enter data as follows:

   For *Options*
   Select *Query LDAP Directory for R/3 Instances.*

   For *Auto Refreshment*
   Enter the periods in which you want system information in the MMC to be updated automatically.

   For *Options*
   Select *Expert user mode* if you want to access more detailed system information later when you start working with the R/3 snap in.
   Deselect *Always show local R/3 instances.* These are automatically displayed in the MMC when the configuration is complete.
10. Choose *Next* to close the *General Settings* dialog box.

    The *LDAP Directory Connection* dialog box opens.

6. Enter the following data:

Fields	Entry
LDAP Server	*Directory Type* Select *Windows 2000 Active Directory*. If you want to access a different *LDAP Server*, select *Generic LDAP Directory*
	SAP Root DN For *Windows 2000 Active Directory* no entry is required. For *Generic LDAP Directory*, enter the distinguished name of the SAP root node in the LDAP directory.
	Directory Server Enter the name of the server where the *Active Directory* is located. For *Windows 2000 Active Directory*, an entry is only required, if the MMC is running on an NT 4.0 host. You can enter several host names separated by spaces. In this case, the first functioning host is the one that will be accessed.
LDAP Authentication	Select the mode of authentication for accessing the directory services. For *Windows 2000 Active Directory* any of the offered modes can be selected. For *Generic LDAP Directory*, select *anonymous log in* for read-only access of the directory. For read and write access, specify a user and password.
	Use Secure Socket Select this to encrypt the information that is passed between the MMC and *Active Directory* machine.

7. Choose *Next*.

   The *Directory Settings* dialog box opens.

8. Specify the information contained in the *Active Directory* that you want to be visible in the MMC:

   a. Qualify the systems and instances you want to see by selecting or entering data in the fields *Management Domain, System, Host and Instance No*. The wild card * is a valid entry.

   b. Choose *Show* to add entries to the *LDAP Search filter* text box. Use the *Hide* and *Remove* options as required:
   Hide allows you to exclude selected items that appear in the *LDAP Search filter* box from the MMC display.
   Remove allows you to delete selected entries from the *LDAP Search filter* box.

   c. Repeat the procedure until you have specified all the management domains, hosts and instances you want to display.

d. Select *Hide non DCOM manageable instances*, to exclude any machines from the MMC display that cannot be monitored because they do not support DCOM communication with an R/3 instance. Currently only Windows NT and Windows 2000 support DCOM.

9. Choose Finish.

   The MMC fetches the information specified in the *LDAP Search filter* from the *Active Directory* and configures the console accordingly.

10. Close all dialog boxes that are still open.

See also MMC Enhancements for Active Directory Services [Seite 1102]

MMC Enhancements for Active Directory Services

Use

When the MMC is configured on the basis of the *Windows 2000 Active Directory* it offers a number of extra features. These take advantage of the information stored in the directory to give the user additional up-to-date technical data.

Prerequisites

The enhanced MMC is only available if you have configured your R/3 System to integrate LDAP services and the MMC has been set up for *Active Directory Services.*

For more information see Configuring MMC for Active Directory Services [Seite 1099].

Features

Special features that are incorporated in the enhanced MMC are:

- Display of management domains
- Direct access to database administration functions
- Display of additional technical data for systems and instances

Management Domains

The enhanced version of the MMC contains additional hierarchical levels in the console tree. The new levels are at the top of the hierarchy and reflect the structure of the management domains in the *Active Directory*. For example, if the *Active Directory* groups R/3 systems into management domains that reflect the location of systems, the MMC is able to reflect this organizational structure.

Database Administration Functions

The new version of the MMC tree includes a node for the database server. This is marked with a special blue icon and can be expanded to view all available database server logs. In addition, the context menu that is available for the database server offers direct access to a number of database administration functions.

At present, the database logs and administration functions are only available for SQL Server. The context menu for the SQL Server lets you directly access the Enterprise Manager to perform a number of tasks:

- Start SQL Server
- Stop SQL Server
- Generate SQL Scripts
- Backup database
- Restore database
- Shrink database

Additional Technical Data

When you choose the *Properties* option in the context menu for an R/3 System or instance, data about the corresponding system or instance is displayed. In the enhanced MMC, this data is more extensive than in the standard version. In addition, it allows you to enter information in predefined categories that is subsequently stored in the *Active Directory*. For example, when you look at the *Properties* for a selected R/3 system, you can enter a meaningful text for the following:

Information Type	Possible Entry
Status	Downtime due to maintenance or failure
Administrators	Names of system administrators and contact information
Category	Test and development system
Description	Purpose of system

Database Administration in CCMS

Purpose

This component enables you to manage your database using the Computing Center Management System (CCMS). With CCMS, you get extensive support in database administration (DBA) and can perform many DBA functions from within the R/3 System.

Implementation Considerations

SAP recommends you to use CCMS for DBA where possible. The advantage of using CCMS is that you have a central point for managing your database. You also get the full advantage of the R/3 System graphical user interface (GUI), for example, when displaying space usage in your database.

Integration

CCMS is a standard part of the R/3 System. Therefore, you can use it without problem as part of your overall DBA activities.

Features

The main functions for DBA in CCMS are as follows:

- DBA Planning Calendar
- Update statistics, using the cost-based optimizer
- Database System Check
- Database Monitor
- Database Alert Monitor

Since the functionality and appearance differs slightly from database to database, the functions are described separately for each database.

Constraints

You have to perform certain DBA functions outside the R/3 System, that is, using tools supplied by the database manufacturer or developed for each database by SAP (for example, for doing an offline database backup). An example of a DBA tool outside the R/3 System is SAPDBA, available for the Informix and Oracle databases.

SAP/Oracle DBA in CCMS

Database Administration Tasks Supported by the CCMS

You can perform most regularly-recurring database administration tasks from within the R/3 System by using the DBA tools of the CCMS. As the table below shows, after installation, reorganizations and certain other tasks are the only tasks that cannot be carried out from within R/3.

Database Task:	Executed using:
Installation	Operating system
Data backup Backup database files and redo log files	R/3 (CCMS), operating system (SAPDBA)
Performance optimizing Monitor and optimize database performance	R/3 (CCMS), operating system (SAPDBA)
Access optimizing Configure cost-based optimizer, create and update statistics	R/3 (CCMS), operating system (SAPDBA)
Monitoring Monitor database using alert monitor, check database and SAPDBA action logs	R/3 (CCMS), operating system (SAPDBA)
Check Analyze space (number of objects, fragmentation, bottlenecks), check configuration (control file, online redo log files, INIT.ORA parameters)	R/3 (CCMS), (SAPDBA)
Reorganization Reorganize database	Operating system (SAPDBA)
Other tasks For example, database recovery (restore/recovery)	Operating system (SAPDBA)

CCMS Database Administration

The main CCMS tools for database administration are:

- Displaying Backup Logs and Status [Seite 1125] (Transaction *DB12*): Check the status of database backups
  - The DBA Planning Calendar [Seite 1107] (Transaction *DB13*): Automatically schedule and execute database operations (for example, database backups and analyses)
- Displaying SAPDBA Logs [Seite 1131] (Transaction *DB14*): Access action logs created by SAPDBA, BRBACKUP, BRARCHIVE, BRRESTORE or an external program
- The Data Archiving [Seite 1386] administration tool (Transaction *DB15*): Systematic display function for managing archived R/3 data

- The Database System Check [Seite 1142] (Transactions *DB16* and *DB17*): Check important database parameters to be able to recognize critical database situations in time.

- The Cost-based Optimizer [Seite 1132] administration tool (Transactions *DB20* and *DB21*) Configure the cost-based optimizer, and create and update optimizer statistics
- The DB Operations Monitor [Seite 1235] (Transaction *DB24*): Online monitoring of database operations (for example, data backups, database checks, and so on)
- Online Maintenance of Database Parameters [Seite 1170] (Transaction *DB27*): Details and history of parameters
- Maintaining Database Connection Information [Seite 1168] (Transaction DBCO): Information about additional, non-standard database connections
- The Database Performance Monitor [Seite 329] (Transaction *ST04*): Detailed database performance information and histories

For more information, see BC SAP Database Administration: ORACLE [Extern]

The SAPDBA program is a tool for administering the R/3 database (Oracle) and is implemented outside the R/3 System. Use SAPDBA for all database administration tasks, even for those tasks that cannot be administered in the R/3 System using the CCMS (for example, reorganization).

Using the DBA Planning Calendar: Oracle

The DBA Planning Calendar is used to automate database administration. This includes implementing, executing and checking actions.

You can use the DBA Planning Calendar for almost all regular database administration actions. This includes tasks for which the Oracle database system must be stopped, such as offline backups.

There is a restriction on use of the DBA Planning Calendar. You can only use the DBA Planning Calendar to start actions, if the R/3 System is active and available. Tasks, such as a recovery, for which the R/3 System must be inactive, cannot be executed using the *SAPDBA* program.

All DBA activities which stop the database, such as offline backups, will terminate active R/3 transactions. Schedule such activities for night runs, and warn users of the interruption using Utilities: System Messages [Extern].

In all such tasks, the DBA Planning Calendar stops the database and starts the action automatically. The R/3 System is not available as long as the database is stopped. However, the R/3 System itself is not stopped. Once the database is available again, the R/3 System is automatically reconnected.

Prerequisites

Before you use the DBA Planning Calendar for the first time, check the configuration of the following:

- Authorizations:
  Make sure you have the required R/3 authorizations. Also, ensure that the Oracle and operating system users are set up correctly.
- Alerts:
  You can change the default values for backup-related alerts.
- Profiles:
  Ensure that the BRBACKUP and BRARCHIVE profiles are set up correctly.
- Tape drives and tapes:
  Ensure that the required tape drives are available, and the required tapes are initialized.

For more information, see Prerequisites for Using the CCMS DBA Tools: Oracle [Seite 1110]

Procedure

1. **Start the DBA Planning Calendar**
   Choose *Tools* → *CCMS* → *DB administration* → *DBA Planning Calendar* (or call Transaction `DB13`).
2. **Use a predefined action pattern for database administration**
   Choose *Calendar* → *Action pattern*

   An action pattern implements a backup strategy and other database administration activities that must be regularly performed. Using a pre-defined action pattern ensures you are following the SAP standards for database activities.

Once you choose a pre-defined action pattern, the system adds the corresponding activities to the DBA Planning Calendar, and plans the background jobs that will execute the activities.

For more information, see Selecting Action Patterns: Oracle [Seite 1112]

3. **Edit the pre-defined action pattern, to make any required changes to activities or start times**

   For more information, see:

   ! Adding Actions to Planning Calendar (Oracle) [Seite 1115]

   ! Editing/Deleting Actions (Oracle) [Seite 1117]

4. **Analyze the DBA Planning Calendar**
   Check the calendar daily to be sure that scheduled activities have been executed correctly.

   Here is a suggested cycle of activities for using the Calendar:

   - Daily:
     - Start the Planning Calendar and check that the actions scheduled for the previous day were executed successfully.

       For more information, see Checking the Results of Actions: Oracle [Seite 1119]

       Also choose *Environment* → *Backup logs to* verify that your backups are adequate for a database recovery.

       For more information, see Displaying Backup Logs and Status: Oracle [Seite 1125]

     - Make sure that the tapes for the next scheduled backup are installed, initialized and ready to go.

       To determine which tapes are required for the next data backup and offline redo log backup, choose the *Tapes needed* function in the Planning Calendar.

       This function always displays the information for the next database backup or offline redo log backup, but not the information for completed or future actions.

   - As required:
     - Warn users before an offline backup. Transactions active in the R/3 System are canceled when the backup starts.
       For more information, see Utilities: System Messages [Extern]

     - If any action did not run successfully, analyze the problem and repeat the action if necessary.
       For more information, see Troubleshooting: Oracle [Seite 1123]

For more information about the DBA Planning Calendar, see:

- SAP/Oracle DBA in CCMS [Seite 1105]

  This gives you an overview of CCMS database administration tools.

- BC – SAP Database Administration:Oracle [Extern]

  This gives you an overview of the SAPDBA program for managing Oracle databases.

Prerequisites for Using the CCMS DBA Tools: Oracle

Hardware (Tape drives and tapes):

Database backups and offline redo log backups are not interactive. When they are executed from the DBA Planning Calendar, the operator is not informed when a tape needs to be changed.

Therefore, you need one of the following:

- The number of tape drives required to hold all tapes needed. The advantage of this is that BRBACKUP uses tape drives in parallel thus increasing the speed of the backup

  If you start BRBACKUP and BRARCHIVE as separate actions, assign them to separate tape drives using the initialization profile `init<DBSID>.sap`. This allows you to use different tapes for database backups and offline redo log backups

- An automatic tape changer, supported by SAPDBA. This devices changes the tapes as required by BRBACKUP and BRARCHIVE

  For more information, see Initialization Profile init<DBSID>.sap [Extern], parameter descriptions for **backup_dev_type** and **rewind_offline**

Enter the tape drives to use in the BRBACKUP/BRARCHIVE initialization profile `init<DBSID>.sap`

For more information, see Volume Management [Extern].

R/3 System: Authorizations

To use the DBA Planning Calendar, you need authorizations for database administration and background job scheduling. The profiles **S_RZL_ADMIN** and **S_BTCH_ALL** must be entered for the administrator.

For more information, see:

- Profile Maintenance [Seite 278] (authorization object **S_RZL_ADM**)
- Authorizations for Background Jobs [Seite 87]

External programs must be able to run on the database server so that actions affecting the database can be executed from other application servers.

Database System / Operating System: Users and Authorizations

DBA Planning Calendar actions are executed under the authorizations of the host system user with which an R/3 application server was started. This is usually the user **<SID>adm** on UNIX, or **SAPService<SID>** on Windows NT.

The user **<SID>adm** or **SAPService<SID>** must be specified as a trusted user **OPS$** in the Oracle database. Make sure this prerequisite is met. For more information, see the Oracle database documentation.

These prerequisites are normally met in the standard R/3 System.

BRBACKUP and BRARCHIVE Configuration

The backup programs BRBACKUP and BRARCHIVE use the backup parameter file `init<DBSID>.sap`.

If you are using different tape drives for BRBACKUP and BRARCHIVE, you must adjust the backup parameter file `init<DBSID>.sap`. If you are using multiple tape drives, you must also specify these in the configuration file.

Set the following parameters:

- For BRBACKUP: `tape_address, tape_address_rew`
- For BRARCHIVE: `tape_address_arch, tape_address_rew_arch`

> If you have scheduled BRBACKUP and BRARCHIVE to start in **one** action, the system only uses the tape drives defined for BRBACKUP.

For more information, see Backing Up the Database [Extern] in the *SAP Database Administration: Oracle* documentation.

Selecting Action Patterns: Oracle

The DBA Planning Calendar includes the following action patterns pre-defined by SAP for use in planning backups:

- Full online DB backup and offline redo log backup, daily Monday to Friday, in one run (one month period)
- Full online DB backup and offline redo log file backup daily from Monday to Thursday, in one run. Also includes full offline DB backup and offline redo log backup on Friday (one month period)
- Full online DB backup and offline redo log backup, daily Monday to Friday, in one run (two-week period)
- Full online DB backup and offline redo log file backup daily from Monday to Thursday, in one run. Also includes full offline DB backup and offline redo log backup on Friday (two-week period)
- Incremental DB backup daily Monday to Thursday as well as full online DB backup and offline redo log backup in one run on Friday (one month period)
- Incremental DB backup daily Monday to Thursday as well as full offline DB backup and offline redo log backup in one run on Friday (one month period)
- Incremental DB backup daily Monday to Thursday as well as full online DB backup and offline redo log backup in one run on Friday (two-week period)
- Incremental DB backup daily Monday to Thursday as well as full offline DB backup and offline redo log backup in one run on Friday (two-week period)

You can change or delete activities in an action pattern after selecting the pattern. A pre-defined action pattern provides an optimal backup strategy.

Once you choose a particular action pattern, the activities are entered into the DBA Planning Calendar on the corresponding dates. The background jobs that perform these activities are also automatically scheduled at the appropriate repetition intervals.

Procedure

1. In the DBA Planning Calendar (Transaction `DB13`), choose C*alendar* → *Action pattern*
2. Choose one of the predefined action patterns from the displayed list
3. Enter the time at which the key action in the pattern is to be executed The system suggests an appropriate time, which you can accept or change
4. Optionally, choose the ID of your corresponding factory calendar from the calendar list The factory calendar affects the backup activities. It is used, for example, so that backups are not started on public holidays, when no tapes are available.
5. Specify the profile for the full online DB backup and the offline redo log backup The profiles contain BRBACKUP parameters that influence the backup Usually, the initialization profile `init<DBSID>.sap` is sufficient.
6. The activities contained in an action pattern are automatically added to the DBA Planning Calendar. The system also schedules background jobs for carrying out the activities. All jobs in the action pattern are scheduled to be repeated periodically (each week).

If there are conflicts between the action pattern you have chosen and activities that are already scheduled in the Planning Calendar, the system presents a list of the conflicts. Review and eliminate the conflicts before trying to choose the action pattern again. Usually, you only have to change the start time.

Notes

- The default option used in starting offline redo log backups is `copy_delete_save`.
  For more information, see Command Options for BRBACKUP, BRARCHIVE and BRRESTORE [Extern]
- The database backup on Fridays is executed with verification using `DB_VERIFY`.
  For more information, see Backup Verifications [Extern]
- Before actions are started, you can automatically notify users informing them the R/3 System will be unavailable. This concerns actions that require the database be stopped, such as offline backups.
  It is recommended you notify users about these types of actions, as they may be running transactions that will be terminated abnormally when the system goes offline.
  To automate system message warnings, create an ABAP program that issues a system message. Then schedule the program for regular execution before the database action. For more information, see Utilities: System Messages [Extern]
- Offline redo log files are deleted from the redo log archiving directory after two successful redo log archiving sessions (option `copy_delete_save`).
- The database is checked daily for errors and possible bottlenecks (`sapdba –check`).
- The value `NEXT` is automatically adjusted in all tables every week in order to optimally adjust the extent size to the actual requirements (`sapdba –next`).
- Every Saturday, the system determines the need for update of statistics for all tables listed in the table `DBSTATC` (`sapdba –checkopt PSAP%`).
- Every Sunday, the system updates statistics for all tables that need statistics as determined by the system on Saturday (`sapdba –analyse DBSTATCO`).
  - The pre-defined action patterns offer you backup schedules with two different tape retention periods.
    - **One month retention period**

      This variant requires that backup tapes be retained for at least 28 days. The tapes can be used again after they are older than 28 days. Make sure the parameter for the tape retention period (`expir_period`) in the backup parameter file `init<DBSID>.sap` is set to *28*.

      Tapes that are used daily for data backup have the name `<SID>B<DATE>`. Replace `<SID>` with the database name (for example, `C11`) and `<DATE>` with the current day of the month.
    - **Two week retention period**

      This variant requires that backup tapes be retained for least 14 days. Tapes can be used again after they are older than 14 days. Make sure the parameter for the tape retention period (`expir_period`) in the backup parameter file `init<DBSID>.sap` is set to *14*.

Name tapes that are used daily `<SID>B<DAY><WK%2>`. Replace `<SID>` with the database name (for example `C11`), `<DAY>` by the number of the current day of the week (Monday=1,...,Sunday=7) and `<WK%2>` by the current calendar week using the numbers 0 and 1 (0 for even weeks and 1 for odd weeks).

- Number of tapes required: For each day of the retention period, you need one tape for the data backups (database and offline redo log backups in one run) named according to the naming convention. Tapes must be initialized as required by BRBACKUP and BRARCHIVE.
- Backups requiring more than one tape: For large databases, one tape is not enough for a complete backup. Change the action in the Planning Calendar as required (add additional tape names). In this case, the backup can run in parallel on multiple tape drives. Initialize additional tapes according to the naming conventions.

For more information, see:

  - Editing/Deleting Actions: Oracle [Seite 1117]
  - Limitations of the Database System [Extern]
  - Prerequisites for Using the CCMS DBA Tools: Oracle [Seite 1110]
- Volume Management [Extern]

Adding Actions to DBA Planning Calendar: Oracle

Do the following to add database operations to the Planning Calendar:

1. Place the cursor on a day to which you want to add a new action. Double-click an empty line or choose *Create action*. The system displays a dialog box listing the actions supported by the Planning Calendar.
2. In the dialog box, enter the start time for the action.
   You can trigger an action immediately for the current day by choosing *Start immediately*. This allows you to start an action manually if, for example, the planned action was unsuccessful.
3. Optionally, enter the repeat interval, in weeks, for an automatic repeat of an action. Without a repeat period, the action is run only once.
4. Optionally, enter the ID of the factory calendar to use for the action The action will not be executed on public holidays.
5. Finally, select the activity you want to schedule. Choose from the following activities:

   Full database offline + redo log backup
   Full offline backup (level 0) of the database using BRBACKUP, as well as archiving of any offline redo log files not yet archived, using BRARCHIVE. BRARCHIVE is automatically started from BRBACKUP. After the offline redo log files have been successfully backed up, you can delete them from the redo log archiving directory according to the BRARCHIVE function you have selected.

   Full database offline backup
   Full offline backup (level 0) of the database using BRBACKUP

   Full database online + redo log backup
   Full online backup (level 0) of the database using BRBACKUP, and then archiving of the offline redo log files, using BRARCHIVE. BRARCHIVE is automatically started from BRBACKUP. After the offline redo log files have been successfully backed up, you can delete them from the redo log archiving directory according to the BRARCHIVE function you have selected.

   Full database online backup
   Full online backup (level 0) of the database using BRBACKUP

   Redo log backup
   Backup of the redo log files from the archiving directory, using BRARCHIVE After the offline redo log files have been successfully backed up, you can delete them from the redo log archiving directory according to the BRARCHIVE function you have selected.

   Partial database offline backup
   Offline backup of the specified tablespaces using BRBACKUP

   Partial database online backup
   Online backup of the specified tablespaces using BRBACKUP

   Check optimizer statistics
   Determines whether new optimizer statistics are needed (`sapdba -checkopt`)

   Update optimizer statistics
   Analyzes either the tables in table DBSTATC or the specified tablespaces, using

SAPDBA (`sapdba -analyze`)
For more information, see SAPDBA Command Mode [Extern]

Adapt next extents
Adaptation of the values for the NEXT extent of the tables in a specified tablespace using SAPDBA (`sapdba -next`).

Check database
Checks the database system using SAPDBA (`sapdba -check`) For more information, see Using the Database System Check: Oracle [Seite 1162]

Incremental offline + redo log backup
Incremental offline backup (level 1) of the database using BRBACKUP, as well as archiving of any offline redo log files not yet archived, using BRARCHIVE. BRARCHIVE is automatically started from BRBACKUP. After the offline redo log files have been successfully backed up, you can delete them from the redo log archiving directory according to the BRARCHIVE function you have selected.

Incremental database offline backup
Incremental offline backup (level 1) of the database using BRBACKUP

Incremental online + redo log backup
Incremental online backup (level 1) of the database using BRBACKUP, as well as archiving of any offline redo log files , using BRARCHIVE. BRARCHIVE is automatically started from BRBACKUP. After the offline redo log files have been successfully backed up, you can delete them from the redo log archiving directory according to the BRARCHIVE function you have selected.

Incremental database online backup
Incremental online backup (level 1) of the database using BRBACKUP

6. Define parameters for the particular action
7. Choose *Continue* to finish scheduling the action. The system checks for conflicts with previously-scheduled activities. If any are found, the conflict are displayed and the action is not scheduled. Otherwise the action is added to the Planning Calendar and the required background job is scheduled.

For more information, see SAP Backup Utilities [Extern]

Editing/Deleting Actions (Oracle)

Use this procedure to change the scheduled time of an action or the command arguments used to execute it (including the names and number of tapes available for the backup or archiving).

You can also delete individual actions. However, you cannot delete all actions of an action pattern as a unit.

Prerequisites

Be sure that the changes you plan to make do not compromise critical database maintenance activities. For example, be sure that you do not alter or delete backup actions in such a way that you are no longer able to recover the database in the event of an error.

Procedure

1. Select a particular day, for which you want to change or delete an action, by double-clicking the header of the action, or the + hotspot on the header.

   Alternatively, select the action you want to change or delete and choose *Change action* or *Delete action*.

2. Select the action and choose *Delete action* or *Change action*

   Delete action: The system asks you to confirm the deletion. When you do so, the action is deleted from the DBA Planning Calendar. The corresponding background job is also deleted. Deletion of the job also stops automatic periodic repetition of the action, if scheduled. The procedure is finished.

   Change action: The system displays a dialog showing the actions supported by the DBA Planning Calendar. The action you chose is selected. Continue with the rest of this procedure.

3. Change the start time or repeat period for the action, as required

4. Choose *Parameters*, if you want to change the attributes for the action. This allows you to change SAPDBA command arguments, profiles, or tablespaces, or add tapes and so on, depending on the selected action.

   Are you changing the number of tapes required for a backup? Then you need to make the change only once. The change is automatically activated for all repetitions of the action.

5. Choose *Continue* to finish scheduling the action.

Result

The system checks for conflicts with previously-scheduled activities. If any are found, the conflicts are displayed and the action is not scheduled.

Otherwise the action is added to the Planning Calendar and the required background job is scheduled.

Checking the Results of Actions: Oracle

The CCMS offers three ways to check the results of actions:

- In the DBA Planning Calendar, for all scheduled actions (described here):
  - The color-coding in the DBA Planning Calendar provides an overview of the status of scheduled actions.
  - For more detailed information about an action, display the database action log and the R/3 background job log.
- Also, Displaying Backup Logs and Status: Oracle [Seite 1125] (Transaction *DB12*):

  This provides you with three additional tasks you can execute that are not possible using the Planning Calendar. You can:
  - Check the status of your backups
  - Display all backup logs and a history of backup activity in your R/3 System.
  - Use these to determine whether the backups you have could be used to recover the database after a database error (*Recovery report*)
- Use the DB Operations Monitor [Seite 1235] (Transaction DB24) for online monitoring of database operations within both internal and external database tools. You can also monitor runtime and remaining time of operations that are running. The DBA operations monitor can be used regardless of which database you are using. It provides historical as well as current (online) information

For more information, see Troubleshooting: Oracle [Seite 1123]

Color-coding of headers

Color-coding of headers in the DBA Planning Calendar is as follows

Background color	**Text color**	**Meaning**
Light blue	Black	There is no action scheduled for today, or no action has failed.
Dark blue	Black	At least one action failed today.
Light turquoise	Black	All actions scheduled for today completed successfully, or no action was executed.
Dark turquoise	Black	At least one action executed today failed.

Color-coding of action lines

Color-coding of action lines in the DBA Planning Calendar is as follows:

Background color	**Text color**	**Meaning**
White (light gray in the Windows NT GUI)	Black	Action scheduled for execution today.

Light gray (gray in the Windows NT GUI)	Black	Action scheduled for execution on a day in the future.
Violet	Black	The action is currently running
Green	Black	The action has been completed successfully.
Yellow	Black	The action completed successfully, but warning messages were issued.
Red	Black	Errors occurred during execution of the action, or the database action log is missing. Analyze the problem and repeat the action if necessary to ensure availability of a complete backup or continued safe operation of the database.
Gray	Red	The action was or will not be executed due to a public holiday.

Displaying the Status of an Action

1. Select a particular day, for which you want to display the status of an action, by double-clicking the header of the action, or the + hotspot on the header.
2. The system displays the status of each action. Possible statuses are as follows:

SCHED	The action has been scheduled, but the scheduled time has not yet been reached. Color: Gray
START	The action started and is still running Color: Violet
SUCC.	The action has been completed successfully. Color: Green
WARN.	The action completed successfully, but warning messages were issued. Color: Yellow
ERROR	The action has either terminated abnormally or is currently running Color: Red
UNKN.	The status of the action is unavailable Color: Red

Displaying Action Logs and Job Logs

From the DBA Planning Calendar, you can display the logs generated by Planning Calendar actions. These logs detail the results of an action. You can also display the job logs generated by the background jobs that carry out Planning Calendar activities

- To determine whether a database log (action log) has been generated for a particular action, select a day by double-clicking the header for the day, or the + hotspot on the header.

  The *Prot.*field is selected if a log was written.

- To display available action logs and job logs, put the cursor on a particular day.

  Choose *Action logs* to display a list of the database action logs generated on that day.

Choose *Job logs* to see the background processing job logs generated by the activities on that day.

Log and action timestamps are used to associate database logs and scheduled actions. An action log is assigned to an action that has the same type and the closest corresponding timestamp. In rare cases, this assignment method may incorrectly associate a log and an action.

For more information, see Background Processing [Seite 74]

Displaying Database Alerts for the DBA Planning Calendar: Oracle

To call the alert monitor, choose *Tools* → *CCMS* → *Control/Monitoring* → *Performance menu* → *Database* → *Alert monitor*.

The CCMS Alert Monitor warns you if a critical situation has occurred during database backup.

For more information, see SAP/Oracle Database Monitor: Introduction [Seite 329].

Initiating an Offline Redo Log File Backup Manually

If the offline redo log files are not being backed up automatically, and you realize that the available space in the archiving directory is too low, backup and then delete the offline redo log files from the archiving directory manually. Use the *Immediate start* mode in the DBA Planning Calendar for this.

For more information, see Adding Actions to the DBA Planning Calendar: Oracle [Seite 1115].

Troubleshooting: Oracle

On rare occasions, an action planned using the DBA Planning Calendar may not execute correctly, for a variety of reasons. You should therefore perform daily checks of at least the critical actions, such as database backups.

To find out why the failure occurred:

1. Was the background job executed correctly?
   Consult the job log. If no job log exists, the background job was probably not started. You can get more details using the job overview in Transaction *SM37* (note that the names of all jobs scheduled in the Calendar start with `DBA...`). The job log will also tell you whether an external program was started.
2. Consult the action log (if one is available) if you are sure that the background job ran successfully.
3. If available, also display the detail log. This log contains details of the external program, and any errors which occurred during execution.

After you have fixed the error, restart the action manually. Choose *Start immediately*. Make sure the action does not conflict with other planned actions.

For more information, see Checking the Results of Actions: Oracle [Seite 1119]

Common errors when using a tape drive include:

- No tape inserted in the tape drive
- The tape in the tape drive is write protected.
- The tape in the tape drive has not been initialized.
- The tape drive contains the wrong tape.
- The tape is already full
- An error has occurred in the tape drive

For more information, see BC SAP Database Administration: Oracle [Extern]

Backup on a standalone Windows NT database server

You must activate the `SAP Gateway`, before starting a backup on a standalone Windows NT database server, using BRBACKUP. Do the following:

1. Choose *Tools → System administration → Administration → Network → RFC destinations*.
   Alternatively use Transaction *SM59*.
2. Click + to open the `TCP/IP connections` node.
3. Double-click `SAPXPG_DBDEST<hostname>`
4. Choose *Destination → Gateway options*
5. Specify a `Gateway host` and a `Gateway service`
6. Choose *O.K.* and save the transaction.

Displaying Backup Logs and Status: Oracle

Process Flow

1. Choose *Tools* → *CCMS* → *DB Administration* → *Backup logs*
   Alternatively: Call transaction `DB12`
2. Choose from one of the three functions as needed:
   – *Recovery report*
     Use this to check whether your backups are adequate for a database recovery; display the backup and redo logs currently required to recover the database.

     See Using the Recovery Report: Oracle [Seite 1130]
   – *Database backups*
     Display the BRBACKUP log for the last successful / unsuccessful backup or for all backups.

     See Displaying Backup Logs: Oracle [Seite 1126]
   – *Redo log backups*
     Display the location and status of the redo log directory as well as the amount of free space available in the directory; the archive status of offline redo logs; and BRARCHIVE archiving action logs.

     See Displaying Redo Log Backups: Oracle [Seite 1128]

Displaying Backup Logs: Oracle

The *Database backups* area in Transaction *DB12* (see Displaying Backup Logs and Status: Oracle [Seite 1125]) provides information about backups of database data files. You can find out when the last unsuccessful backup (return code unequal to 0 or 1) and last successful backup occurred.

You thus get an overview of how current your backups and can judge whether they are adequate. You should have several generations of successful backups whose dates are not too far in the past.

- To display the database log of the last failed backup, choose ***Last unsuccessful backup*** To analyze the error more closely, display the file system log by selecting the *Detail log* pushbutton.

  This information is only displayed, if an unsuccessful backup exists.

- To display the database log of the last successful backup, choose ***Last successful backup***

  The following information is displayed:

Tape name	Tape name
Pos.	Position of the file on the tape
Backup time	Time the backup of the file ended
Compr.	Compression rate (if compression was used)
File ID	ORACLE file ID
File name	File name
Tablespace	Tablespace name
RedoNo.	Log sequence number of the online redo log file
TS status	ORACLE tablespace status
File status	ORACLE file status
Backint ID	Backup ID of Backint interface (only if external backup program was used)

  For more information, see Detail Log [Extern]

- For an overview of all existing backup logs, choose ***Overview of database backups*** Backups that ran incorrectly are highlighted in red.

  The following information is displayed:

Backup function	Brief information about the backup performed
Start of backup	Time the backup started
End of backup	Time the backup ended
RC	BRBACKUP For more information, see Return Codes [Extern]

Log	File system log name (`<Action ID>.<Function ID>`, see Names of the Detail Logs [Extern])

For further analysis of a backup, double-click the appropriate line to display the database log. Choose *Detail log* to look at the corresponding file system log.

For more information, see Summary Log [Extern] and Log Types [Extern].

Displaying Redo Log Backups: Oracle

The area *Redo log backups* in Transaction *DB12* (see Displaying Backup Logs and Status: Oracle [Seite 1125]) provides information about archiving of the offline redo log files. The system displays the amount of free space available in the archiving directory and the number of offline redo log files from the last generation that have not yet been archived.

- Choose **Archiving directory status** for the name of the database host, the name of the archiving directory, the status of the directory, and the amount of freespace in the directory.

- To display a list of offline redo log files including their archiving statuses, choose ***Overview of Redo log files***

The following information is displayed:

RedoNo.	Log sequence number
Time created	Time the offline redo log file was created
Tape name	Tape name
Pos	Position of the file on the tape
Backup time	Time the offline redo log file was archived
Backint ID	Backup ID of Backint interface (only if external backup program was used)
Offline redo log file name	Directory and name of the redo log file

- For an overview of all existing archiving logs, choose ***Overview of redo log backups*** Archiving sessions with errors are highlighted in color (red).

The following information is displayed:

Backup function	Brief information about the archiving performed
Start of backup	Time archiving started
End of backup	Time archiving ended
RC	BRBACKUP For more information, see Return Codes [Extern]
Log	File system log name (`<Action ID>.<Function ID>`, see Names of the Detail Logs [Extern])

For more information, see Summary Log [Extern]

For further analysis of an archiving session, double-click the appropriate line to display the database log. To display the file system log, choose *Detail log*.

The following information is displayed:

Function	Backup functions: *Save*: Creates first copy of the offline redo log file *Copy*: Creates second copy of the offline redo log file *Del*: Deletes offline redo log file from the archiving directory
RedoNo.	Log sequence number
Time created	Time the offline redo log file was created
Tape name	Tape name
Pos.	Position of the file on the tape
Backup time	Time the offline redo log file was archived
Compr.	Compression rate (if compression was used)
Backint ID	Backup ID of Backint interface (only if external backup program was used)
Offline redo log file name	Directory and name of the redo log file

For more information, see Detail Log [Extern] and Log Types [Extern].

Using the Recovery Report: Oracle

Choose *Recovery report* in transaction DB12 (see Display Backup Logs and Status: Oracle [Seite 1125]).

The system display information about the last successful backup (kind of backup, tape names); this tells you which backup you could use if a database error occurs making a recovery necessary. The report also checks whether the required redo log files are available (archived on tape or in the archiving directory). You thus know which files to restore in the event of a recovery.

Check the recovery report regularly in order to detect possible gaps in your backup mode. Note the following:

- Missing redo log files mean that if an error occurs, the database can no longer be restored to the current time point. In this case, you should perform a full database backup as soon as possible.
- If the list of redo log files is too long, a recovery to the current time point may take a long time. In this case, you should perform a full database backup as soon as possible.

For more information, see:

- SAPDBA Database Recovery [Extern]
- SQLDBA Database Recovery [Extern]
- Backup mode [Extern]

SAPDBA Logs: Oracle

1. Choose transaction `DB14`.
2. Choose from the pushbuttons *SAPDBA*, *DB OPTIMIZER, BRBACKUP* and *BRARCHIVE* to look at the logs for operations created using these programs.

The logs contains the following information:

Begin of action	Start of the operation (date and time)
End of action	End of the operation (date and time)
Fct	Type of operation
Object	Database object or keyword
RC	Return codes For more information, see Return Codes [Extern] and RC [Extern].

For detailed information about an individual database operation, double-click it.

Choose *Function IDs* for a list of all possible database operations (function ID), their respective assignment to the executing program, and a short explanation of the operations.

For operations not contained in *Function IDs* , choose *Others*.

For a list of all operations executed, choose *All*.

For more information, see:

- Displaying Backup Logs and Status: Oracle [Seite 1125]
- BRBACKUP, BRARCHIVE and BRRESTORE Logs [Extern]
- SAPDBA Logs [Extern]

CCMS Support for the Cost-Based Optimizer: Oracle

From Release 4.0, the cost-based optimizer (CBO) is a standard part of the R/3 System. If statistics are available for a table, the database system uses the cost-based optimizer. Otherwise, it uses the rule-based optimizer.

The CCMS includes the follow functions for the cost-based optimizer:

- **Automatically recognize if table statistics need to be updated:**
  To check the database for tables with old CBO statistics, call the DBA Planning Calendar (Transaction *DB13*) and use the *Check optimizer statistics* function (for example: `SAPDBA -CHECKOPT PSAP% -METHOD E`). This operation is the first step required to update CBO statistics.

  See Updating CBO Statistics Automatically [Seite 1134]

- **Update Optimizer Statistics Automatically:**
  To automatically update CBO statistics, call the DBA Planning Calendar (Transaction *DB13*) and use the *Update optimizer statistics* function (for example, `SAPDBA -ANALYZE DBSTATCO`) This operation is the second step required to update CBO statistics.

  See Updating CBO Statistics Automatically [Seite 1134]

- **Update Statistics for a Single Table:**
  Use *Edit DB Statistics* (Transaction *DB20*), to create new CBO statistics, or update or delete statistics, for a particular table.

  See: Checking and Updating Statistics for a Single Table: Oracle [Seite 1136]

- **Execute Standard CBO Operations:**
  Use *Standard operations* (Transaction *DB20*) to:
  – Analyze the need for statistics for all database tables
  – Create statistics according to control table `DBSTATC`
  – Create statistics for tables without statistics
  – Delete all illegal statistics

  See: Standard Operations for CBO Statistics [Seite 1141]

- **Maintain the Optimizer Control Table `DBSTATC`:**
  Use *DB Optimizer Control* (Transaction *DB21*) to configure automatic updates of CBO statistics with the control table `DBSTATC`. Once you schedule the recommended CBO actions in the DBA Planning Calendar, the R/3 System will administer the control table on its own. Therefore, do not add, delete, or change table entries unless you are aware of the consequences of doing so.

  See: Maintaining DBSTATC Control Table [Seite 1138]

CBO statistics, installations, and release upgrades:
Generating CBO statistics for all tables in the R/3 system is an activity you will need to perform in association with an installation or release upgrade. These actions are described in the relevant installation or upgrade guide.

Updating Cost-Based Optimizer Statistics Automatically

The DBA Planning Calendar completely supports the updating of CBO statistics. Use the DBA Planning Calendar to not only automatically check for old or missing statistics (phase 1), but also to automatically create statistics (phase 2).

Procedure

1. Add action `Check optimizer statistics` with the option `PSAP%: All SAP tablespaces` and the method `E` to the DBA Planning Calendar (*DB13*)

   This action is not one of the predefined action patterns. It must be added separately to the Planning Calendar. For more information, see Adding Actions to the DBA Planning Calendar: Oracle [Seite 1115].

   Planning Strategy:

   We recommend one of two planning strategies for carrying out phases 1 and 2: Plan the check for old or missing statistics (phase 1) to run once a week (for example, on Saturdays). Plan the update of statistics (phase 2) to run after phase 1 has finished (for example, on Sundays). In this model, CBO statistics are checked and refreshed completely once a week. Updating CBO statistics is also important (even if only for individual tables) after significant imports of data (batch input, for example).

   Runtime:

   The check of the statistics may take several hours depending on the number and size of the tables in the database. `SAPDBA -CHECKOPT` places a load on the database system and should normally only be run when database load is low.

   Background:

   The action executes the function `SAPDBA -CHECKOPT` with the option `PSAP%` and the method `E` and checks the validity of all CBO statistics for all tables in the database. Tables requiring new statistics are entered in the control table `DBSTATC` (if not already in the control table) or flagged, if already in the table. In phase 2 (`SAPDBA -ANALYZE DBSTATCO`), the statistics are updated.

2. Add action `Update optimizer statistics` with the option `DBSTATCO: All tables marked in DBSTATC` to the DBA Planning Calendar (*DB13*).

   This action is not one of the predefined action patterns. It must be added separately to the Planning Calendar. For more information, see Adding Actions to the DBA Planning Calendar: Oracle [Seite 1115].

   Do not use the other options for this action.

   Planning Strategy:

   This action should be planned to run once a week after `SAPDBA -CHECKOPT PSAP% -METHOD E` (phase 1) (see above).

   Runtime:

The update of the statistics may take several hours depending on the number and size of the tables in the database. SAPDBA -ANALYZE places a load on the database system and should normally only be run when database load is low.

Prioritize the statistics for critical tables by changing the priority in the control table DBSTATC. For more information, see Maintaining the DBSTATC Control Table [Seite 1138]

3. Check the results of the actions in the DBA Planning Calendar in the DBA Operations Monitor.

   For more information, see Using the DBA Operations Monitor [Seite 1235].

Result

The CBO statistics for the tables in the DBSTATC control table will be regularly checked and updated.

At delivery, the control table contains more then 400 critical R/3 tables (these are tables that grow quickly, whose CBO statistics need to be updated regularly).

SAPDBA -CHECKOPT automatically updates the entries in the control table DBSTATC and changes the method used to create the statistics, once a table reaches a certain size. In addition, SAPDBA -CHECKOPT adds new entries as needed, when tables that are not yet in the control table grow or shrink, and therefore require new CBO statistics. SAPDBA administers the control table and automatically deletes table entries, once the size of a table has stabilized. Automatic administration of the control table provides a better overview.

In rare exceptions, you could also manually add entries to the control table DBSTATC. For more information, see Checking and Updating Optimizer Statistics for a Single Table (Oracle) [Seite 1136] and Maintaining the DBSTATC Control Table [Seite 1138]

For more information about the SAPDBA commands -CHECKOPT and –ANALYZE, see SAPDBA Command Mode [Extern] .

Checking and Updating Statistics for a Single Table: Oracle

Prerequisites

Use this function only when you determine there is an immediate need for updated CBO statistics for a particular table.

For example, use this function when:

- You or SAP's EarlyWatch determine that a particular table needs updated CBO statistics. That is, you have a CBO-related performance problem relating to a table.
- You import a large amount of data into a particular table (for example, using batch input). The additional records make the CBO statistics inaccurate.

Procedure

1. *Choose Tools → CCMS → DB Administration → Cost-based optimizer → Create statistics.* Alternatively, use Transaction *DB20*.
2. Enter the name of the table (do not enter a generic name or pattern) and choose *Refresh information*.
3. Choose *Number of table entries* to determine whether the table's CBO statistics need refreshing. Then do the following:
   – Confirm the warning message. Depending upon the size of the table, this check may take a few seconds or up to several minutes or even longer.
   – After the check is complete, look at the field *Deviation old/new*. The contents of this field indicate how the CBO statistics deviate from the current table status. If the deviation is 10% or greater, update the statistics for the table. Otherwise, the deviation between the CBO statistics and the actual state of the table is not significant enough to require new statistics.

     The field *Old value* displays the number of table entries before the statistics were created. The field *New value*, displays the current number of table entries.
4. Do the table's statistics need to be refreshed? If yes (deviation ≥ 10%), do the following:
   – Set *Accuracy* to *Low* or *High*.

     If the runtime is acceptable, choose *High* accuracy.
   – Choose *Create statistics*.

     The R/3 System updates the CBO statistics for the table as well as the fields *Time* and *Date*.

     This action does not add the table to the control table DBSTATC. If the table is already in the control table and accuracy is set to *High*, a set TODO flag is canceled, thus preventing an unnecessary update of the statistics during the next SAPDBA -ANALYZE run.
5. Do the table's statistics need to be deleted? If yes, choose *Delete statistics*.

For more information, see Updating Optimizer Statistics Automatically (Oracle) [Seite 1134]

Maintaining the Control Table DBSTATC

The table DBSTATC controls the update of statistics for the cost-based optimizer. The entries in the control table DBSTATC are administered according to the SAPDBA Internal Rules for Determining Statistics [Extern]. Only change the control table under special circumstances.

This function allows you to:

- Display the control table, to check the status of the CBO update Use this, for example, to check if a statistics update is planned, the last time statistics were updated, or which method was used to update the statistics.
- Change the default method for analyzing a table and creating statistics
- Prevent SAPDBA from changing the method for creating statistics entered in the table
- Give a table higher priority for update of statistics

At delivery, the control table DBSTATC contains more than 400 entries. The tables in the control table are those (among others) that:

- Require regular updates of their CBO statistics
- Belong to the application monitor
- Are to be excluded from statistic updates
- Require a special analysis method

The R/3 System administers the control table. It adds tables to the control table, whose statistics need to be updated, and also deletes tables, whose size has not changed over a long period of time. In addition, the R/3 System determines the best method for updating the statistics depending on how the size of the table has grown.

If you are not completely certain a change is required, do not add, delete or edit any of the entries in the control table. Deleting entries may delay updating of old CBO statistics. Adding and editing entries may unnecessarily increase runtimes for updates of CBO statistics, or reduce the effectiveness of the statistics.

Procedure

1. Choose *Tools* → *CCMS* → *DB Administration* → *Cost-based optimizer* → *Configuration.* Alternatively, call Transaction *DB*21.
2. Check the control table entries The table includes the following columns:
   - ***DB object:*** Name of the R/3 table, whose statistics are to be checked and updated
   - ***Use:*** Use of the table in the R/3 System The use types are:

     A: For the R/3 application monitor (*ST07*) and also possibly for the optimizer

     O: Only for the optimizer (default setting)
   - ***Active***: Control flag. It indicates if and how the statistics are updated or created, and in which context they are used. The possible settings are:

A (Active)
Statistics are created or updated, if the TODO flag is set (default setting)

N (No)
Statistics must not be created or updated; SAPDBA deletes existing statistics during the next analyze session

P (Priority)
Statistics that have the priority (P) are updated or created before those statistics with lower priority (A), if the TODO flag is set (otherwise, the same as A)

R (Restrictive)
Short-term statistics can be updated or created. SAPDBA deletes existing statistics during the next analyze session.

U (Unconditioned)
Statistics are updated or created during every analyze session, even if the TODO flag is not set in the control table DBSTATC.

Only use the setting *N* in emergencies. In terms of the CBO, the setting *R* is sufficient. It also allows reorganization, which reduces the amount of space required.

– **TODO:** When it is set (X), the statistics for the table will be updated or created during the next `analyze` session (phase 2). The TODO flag is automatically set by SAPDBA during the `checkopt` session. The field remains empty, if no update is required. To force an update of statistics for a table, set the TODO flag yourself.

– **TODO chg. Date:** Date when the TODO flag was set or deleted.

– **Analysis method:** Analysis method used to update or create the statistics: Automatically set by SAPDBA. You can manually change the setting if required. The possible analysis methods are:

E
Estimate table statistics, including the index (default setting)

EI
Estimate table statistics, check index structure

C
Calculate exact table statistics, including the index

CI
Calculate exact table statistics, check index structure

SAPDBA sets the appropriate analysis method, depending on whether a table grew or shrunk. For more information, see SAPDBA Internal Rules for Determining Statistics [Extern]

3. Choose *New entries*, to add tables to the control table `DBSTATC`. You can edit the following fields:

– ***DB object:*** table name (for example, `APQD`)

– ***DB object type:*** database object type (default setting: `01` for table)

– ***Owner:*** Owner of the table (default setting `SAPR3`)

- ***Database:*** database system used (for example, ORACLE)
- **Use**: see above
- ***Active***: see above
- ***History:*** history flag: if set (X), the statistic results are archived (table DBSTATHORA)
- ***Analysis method:*** see above
- ***Analysis option:*** Specification of the analysis method (only relevant for E and EI)

  Pxx
  xx percent of table entries used for the analysis

  Rx
  x thousand of table records used for the analysis
- ***Cust.:*** Customer flag: If set (X), the settings cannot be changed by SAPDBA or other tools This allows customers to maintain their own settings for existing tables. These settings might be: do not create statistics (Active=N), or always calculate exact statistics (analysis method=C).
- ***Changed on:*** date when one of the settings was changed
- ***TODO:*** see above
- ***Changed on:*** see above

Save the new entry and exit the table maintenance function.

An alternative method to add entries to the control table is to choose *Copy as...*. Choosing *Copy as...* means you only have to modify and save the copied entry.

4. Choose *Details* to edit a selected table entry.
5. Choose *Delete* to delete a selected entry from the table DBSTATC.

For more information, see Updating Optimizer Statistics Automatically [Seite 1134] and Support for the Cost-Based Optimizer [Extern].

Standard Operations for CBO Statistics

Use

Execute the most important actions for updating statistics in the CCMS (Transaction *DB20*). If you have created new tables or executed a large data transfer, you must create the missing statistics or update the current statistics in addition to the normally scheduled statistic operations.

Features

Choose *Tools* → *CCMS* → *DB Administration* → *Cost-based optimizer* → *Create statistics.* Alternatively, call Transaction DB20.

Then choose *Standard operations*. Choose between the following functions:

- CHECKOPT

  Analyzes all database tables (PSAP%) to determine which tables need statistics. It then updates the corresponding entries in the control table DBSTATC.

- ANALYZE

  Creates current statistics for all tables whose TODO flag is set in the control table DBSTATC. This implicitly creates statistics for all tables and indexes which do not have statistics.

- NOOPTSTAT

  Creates statistics for all tables and indexes which do not have statistics.

- HARMFUL

  Deletes illegal statistics.

Choose *Start* to activate the function selected.

Database System Check

Use

You can use the database system check to identify existing and approaching critical situations in your Oracle database system in time. The database system check can be executed with the database in the *OPEN* state as well as in the *MOUNT* state. The database system check examines the following:

- Space (fill level and fragmentation)
- Physical consistency (availability of data files, control files, and redo log files)
- Specific problems based on DBA experience
- Oracle alert messages (`<SID>alert.log`)
- Oracle profile parameters (*init<SID>.ora*)

Activities

To optimize the check of the Oracle database system, do the following:

- **Configure** the database system check according to your requirements

  For more information, see Configuring Database System Check [Seite 1143]

- **Automate** the database system check in the DBA Planning Calendar (*DB13*). If you have selected an action pattern in the DBA Planning Calendar, the database system check (Action *Check database*) is automatically planned to run daily every morning.

  For more information, see Using the DBA Planning Calendar [Seite 1107] and Adding Actions to the DBA Planning Calendar [Seite 1115]

- **Monitor** the results and **take action** for all alerts from the database system check

  For more information, see Displaying Alert Messages from Database System Check [Seite 1162] and Alerts from Database System Check [Seite 1150].

Configuring Database System Check (Oracle)

Use

Use this function to configure the database system check in order to:

- Add new parameters of type *ORA* or *PROF*
- Exclude individual parameters from the check
- Specify threshold values for the parameters
- Create object-specific parameters to exclude them from the check
- Create object-specific parameters to set individual threshold values
- Set the check interval for the parameters
- Specify corresponding corrective actions
- Maintain the parameter documentation

Procedure

To configure the database system check, choose *Tools* → *CCMS* → *DB Administration* → *DB System Check* → *Configuration*. Alternatively: Call Transaction `DB17`.

The configuration data for the database system check is stored in the table *DBCHECKORA*.

The group box ***Number of SAPDBA check parameters*** provides an overview of the current status of the parameters (such as the current number of active parameters. You configure the individual parameters in the table below the box. The table includes the following columns:

Column	Description

Configuring Database System Check (Oracle)

Typ	Type of alert the parameter is assigned to There are four alert types: ***DBA*** Includes 19 parameters at delivery. These check memory space (fill level and fragmentation), physical consistency (availability of data) and DBA-specific problems, in the database system ***DBO*** Includes eight parameters at delivery. These check database operations that: • Terminated or had errors between the time of the last check and the current time A terminated operation or one with errors is reported only once. • Have not been processed successfully since a specific (time) period ***ORA*** Includes 13 parameters at delivery. If there is an error, these are read from the Oracle alert file, `<SID>alert.log`, and an alert is reported Any number of new parameters (Oracle error messages or Oracle strings) may be defined. ***PROF*** Includes 25 parameters at delivery. These compare the values of the Oracle database parameters with the values in the configuration table *DBCHECKORA*. Any number of new parameters (Oracle profile parameters) may be defined.
Parameter	Parameters and Alerts for Database System Check [Seite 1150]
Object	Object-specific instance of a parameter of type *DBA* or *DBO.* The instance can be used to exclude the object from the check or to individualize the object settings. In order to specify object-specific instances of the parameter, a characteristic master parameter must exist. This master parameter is the standard for all specific instances of the parameter. This master parameter does not contain any object specifications. When it is set to inactive, all specific instances of the parameter are set to inactive (this does not apply to parameters of type *DBO*) Only *Operation objects* may be specified for parameters of type *DBO*. These *operation objects* are entered as objects in the XDB tables *DBAREOL*, *DBAOPTL* or *DBASPAL*.
Actv.	Indicates whether the parameter is active (*green*) or inactive (*red*)

Sev.	Severity of the alert message: ***A*** Exception due to a deviation from the SAP standard (for example, a parameter for a *PROF* alert message is not within the value range we recommend) ***E*** Error ***W*** Warning
Operand, **Val.** **Unit**	Definition of the threshold value using operand, value and unit. Possible operands are: ***equal to***, ***not equal to***, ***less than***, ***less than or equal to***, ***greater than***, ***greater than or equal to*** and ***greater than and less than***. The possible units are: ***D*** Days ***G*** Giga ***K*** Kilo ***M*** Mega ***P*** Percent ***R*** Rate, only for type *ORA* (Number of alerts for each check session) ***S*** Seconds With *Operand* = >, V*alue* = *80* and *Unit* = *P*, an alert is triggered when the value of the parameter checked exceeds 80%. With *Operand* = >, V*alue* = *1* and *Unit* = *R,* an alert is triggered if the Oracle alert corresponding to the *ORA* parameter, or the *ORA* parameter corresponding to the Oracle string, is logged in the Oracle alert file <SID>alert.log more than once during the check session.
Period	The minimum time period before the parameter will be checked again For parameters of types *DBO, PROF* and *ORA*: If a parameter has several object-specific instances for which individual periods are specified, the system will always use the largest period for the time interval. You want to check the parameter ARCHIVE_STUCK only once a week. However, the database system check is planned in the DBA Planning Calendar to run daily. For the parameter ARCHIVE_STUCK, enter 7 in the *Period* field and Days in the *Unit* field. The parameter is only checked once a week.

Unit	Unit for period (time interval) Possible units are *Seconds*, *Hours*, or *Days*
Date	Date the last time the parameter was changed
User	User who made the last changes to the parameter
CorrType	Tool to use for the corrective measure(s): ***E*** Text editor ***H*** Hotline ***P*** SAPDBA program ***R*** R/3 report ***T*** R/3 transaction
CorrMeasure	Measures to be taken to fix alert
Description	Description of parameter / alert SAPDBA uses placeholders (#1, #2) to correctly describe the parameters and alerts: Type *DBA*: Do not change these alerts, only add to them Type *DBO*: The placeholder *#1* is replaced by a detailed alert generated by SAPDBA Type *ORA*: The error alert *ORA-...* logged in the Oracle alert log is displayed (and not the existing documentation in the *Description* field Type *PROF*: The placeholder *#1* is replaced by an alert generated by SAPDBA If the placeholder *#1* is not used or the *Description* field is empty, the system displays the Oracle documentation for the database parameter

To display details about a selected parameter, choose `Parameter details`. To update the configuration table display, choose `Refresh`.

Activities

Adding new parameters of type *ORA* or *PROF*

To add a new parameter of type *ORA* or *PROF* to the database system check, do the following:

1. Choose `Create new check (with reference)`
2. In the `Typ` field, choose the type of alert (*ORA* or *PROF*). Enter *Database error message*, to add a new parameter of type *ORA* or *DB Profile parameter* to add a new parameter of type *PROF*
3. In the `Parameter` field enter the name of the parameter
4. In the `Actv.` Field, enter *Yes* to activate the parameter
5. In the `Condition` field, specify the threshold value (operand, value and unit)

6. In the `Description` field, enter a brief description of the parameter
7. In the `Repeat period` field, enter the time period and the unit
8. In the `Corrective measure` field, enter the type of corrective measure and a description of the corrective measure
9. Save the new parameter

The changes made take affect the next time the action *Check database* is executed by the DBA Planning Calendar (DB13) or the SAPDBA command `-CHECK` is executed.

Activating / Deactivating check parameters

Use the column *Actv.* to activate (green) or deactivate (red) parameters.

Specifying threshold values for the parameters

Threshold values can be specified for the following parameters:

Parameter	Error Type
ARCHIVE_STUCK	*DBA*
CRITICAL_SEGS	*DBA*
FS_FULL	*DBA*
MANY_EXTENTS	*DBA*
TSP_FULL	*DBA*
All parameters	*DBO*
Checkpoint not complete (or similar alerts)	*ORA*
All parameters	*PROF*

1. Select the parameter and choose `Change`
2. Specify the operand, value, and unit
3. Save the parameter

Create object-specific parameters to exclude them from the check

Object-specific instances can be created for the following parameters to exclude them from the check:

Parameter	Object	Type
CRITICAL_SEGS	*<Table>* or *<Tablespace>*	*DBA*
FS_FULL	*<SAPDATA>* or *<file system>*	*DBA*
MANY_EXTENTS	*<Table>* or *<Tablespace>*	*DBA*
MISSING_INDEXES	*<Table>*	*DBA*
NO_OPT_STATS	*<Table>*	*DBA*

TABLES_NOT_IN_TABLE_TABLSPACE	*<Table>*	*DBA*
TSP_FULL	*Tablespace*	*DBA*
All parameters		*DBO*

1. Select the parameter and choose `Create object check`
2. In the `Object` field enter the name of the object You can also specify the wildcard * (for example, *character string, *character string* or character string*)
3. In the `Actv.` field, enter *No* to deactivate the parameter
4. Save the new object-specific parameter

Creating an object-specific parameter to set an individualized threshold value

Individualized threshold values can be specified for the following object-specific parameters:

Parameter	Object	Type
CRITICAL_SEGS	*<Tablespace>*	*DBA*
FS_FULL	*<SAPDATA> directory or <file system>*	*DBA*
TSP_FULL	*<Tablespace>*	*DBA*
All parameters		*DBO*

1. Select the parameter and choose `Create object check`
2. In the `Object` field enter the name of the object
3. In the `Actv.` Field, enter *Yes* to activate the parameter
4. In the `Condition` field, specify the threshold value (operand, value and unit)
5. In the `Description` field, enter a brief description of the parameter
6. In the `Repeat period` field, enter the time period and the unit
7. In the `Corrective measure` field, enter the type of corrective measure and a description of the corrective measure
8. Save the new object-specific parameter

Setting parameter check periods

You can set the check period for every parameter:

1. Select the parameter and choose `Change`
2. In the `Repeat period` field, enter the time period and the unit
3. Save the changed parameter

Specifying corrective measures

You can specify a corrective measure for every parameter:

1. Select the parameter and choose `Change`

2. In the `Corrective measure` field, enter the type of corrective measure and a description of the corrective measure
3. Save the changed parameter

Maintaining parameter documentation

Documentation can be maintained for all parameters:

1. Select the parameter and choose `Change`
2. In the `Description` field, enter a brief description of the parameter
3. Save the changed parameter

Parameters and Alerts for Database System Check

Database checks are usually executed with the database in the *OPEN* state. In critical situations, the database system check can check certain parameters while the database is in the *MOUNT* state. These parameters are described in detail in the following tables:

Type *DBA*:

Parameter	Severity	Check during mount state
ARCHIVE_STUCK	**W**	**Yes**
Cause: The amount of archiving directory space for offline redo log files used is above the specified threshold level. The system will crash if the database system runs out of room to archive online redo log files. The operand **>** of the threshold value checks the amount of **used** space in the archiving directory. The operand **<** of the threshold value checks the amount of **free** space in the archiving directory. **Mount state:** Checks whether the percentage of free space in the archiving directory is less than 10%. **Action:** Archive the offline redo log files using SAPDBA (BRARCHIVE).		
CONTROL_FILE_MISSING	**E**	**Yes**
Cause: The system cannot access at least one of the control files (we recommend that three be accessible). If the database system is still running, at least one copy is still online. **Action:** Look for the parameter CONTROL_FILES, in the init<SID>.ora initialization file to determine which directories the control files are stored in. Make sure you can display all the files using the file manager (command ls). Restore access to any non-accessible files.		
CONTROL_MIRROR	**E**	**Yes**

Cause: You have entered less than two control files in the init<SID>.ora initialization file. For safe operation, we recommend a double mirror of the original file. This message is often generated because the control file is mirrored on the hardware level, but this fact is not indicated in the init<SID>.ora initialization file. For safe operation, we recommend that you either use Oracle mirroring (multiple init<SID>.ora entries) or both hardware and Oracle mirroring. **Action:** Edit the init<SID>.ora profile and add the additional control files under the parameter CONTROL_FILES. We recommend you have at least three copies of the file. Each copy of the file must be stored on a separate physical hard disk. After you edit the profile, restart the database to activate your changes. If you only use hardware mirroring, you can switch this check off by setting the parameter *check_control_mirror* in the *init<SID>.sap* SAPDBA initialization file to N. This overwrites the standard configuration (transaction DB17).		
CRITICAL_SEGS	**W**	**No**
Cause: If the number of displayed extents are assigned to the displayed table(s), this would cause a tablespace overflow error. **Action:** Overflow with one extent: Extend the tablespace using SAPDBA and *Tablespace Administration*. Overflow with two extents: Monitor the freespace in the tablespace.		
DF_OFFLINE	**E**	**Yes**
Cause: The system cannot access the data file specified in the message, because this file is offline. All files in a SAP database should normally be online. Files that are offline will not be part of database processing nor part of a database restore. **Action:** Close all applications and enter the following commands using the Oracle Server Manager: 1. *connect internal* 2. *shutdown immediate* 3. *startup mount* 4. *alter database datafile <data file> online* 5. *alter database open* Alternatively, use the SAPDBA function, see SAPDBA Check (and Repair) Database [Extern]		
FILE_MISMATCH	**E**	**Yes**

Cause: The file type for a database file specified in the control file does not match the entry in the Data Dictionary of the Oracle database system. This error is usually caused by errors during processing of the control file or during restoration of the control file. **Action:** Execute `CREATE CONTROLFILE` using the Oracle Server Manager, specifying either the correct file or no file at all.		
FILE_MISSING	**E**	**Yes**
Cause: At least one data file is missing (file cannot be accessed). This error occurs if a file has the status *Recovery* and the file either cannot be accessed or the file is too old. **Action:** Execute the following SAPDBA functions in the following order (Database Check [Seite 1142] and then SAPDBA Check (and Repair) Database [Extern]).		
FILE_TYPE_UNKNOWN	**E**	**Yes**
Cause: The file type (data file, raw device, dir, link) is unknown. **Action:** Troubleshoot to determine the cause of the problem. If you are not sure how to proceed, contact SAP technical support.		
FS_FULL	**W**	**No**
Cause: The amount of *SAPDATA* directory space used is above the specified threshold. The operand **>** of the threshold value checks the amount of **used** space in all *SAPDATA* directories. The operand **<** of the threshold value checks the amount of **free** space in all *SAPDATA* directories. **Action:** Increase the amount of space in the *SAPDATA* directories.		
MANY_EXTENTS	**W**	**No**
Cause: The tablespace(s) named in the alert message is (are) reaching the *MAXEXTENTS* limit or must be reorganized for efficient operation. **Action:** Reorganize the tablespace(s) using SAPDBA. You could also temporarily increase the *MAXEXTENTS* limit for the tablespace(s) using SAPDBA.		
MISSING_INDEXES	**E**	**No**
Cause: *UNIQUE* indexes for SAP tables are missing. SAP tables that by default do not have an index are listed in a table of exceptions. The SAPDBA database check takes this table into account. **Action:** Create the missing indexes using Transaction *SE11*. (*Database table → Indexes*)		
NOARCHIVELOG	**E**	**Yes**

Cause: The Oracle database system is not archiving any online redo log files. SAPDBA checks if the ARCHIVELOG mode and the automatic archiving of online redo log files are active. **Action:** Using SAPDBA, restart the database in ARCHIVELOG mode. Choose *Archive mode → Toggle database log mode*. SAPDBA shuts down the database, reconfigures the database mode, and then restarts the database.		
NO_OPT_STATS	**E**	**No**
Cause: Statistics for the cost-based optimizer are missing. SAP tables that by default do not have statistics are listed in the exception table *DBDIFF*. The SAPDBA database check takes this table into account. **Action (immediate):** Call Transaction *DB20*. Under *Standard operations*, run `NO_OPT_STATS`. **Action (longterm):** Schedule the action *Check optimizer statistics* in the DBA planning calendar.		
REDOLOG_MIRROR	**W**	**Yes**
Cause: The online redo log files are not mirrored. Mirroring is required for safe operation. We recommend activating Oracle mirroring even if you are already using hardware mirroring. **Action:** Using the Oracle Server Manager, activate mirroring of the online redo logs in your database, with the statement `ALTER DATABASE ADD LOGFILE`. If you do not want to use Oracle mirroring, you can switch this check off by setting the parameter *check_control_mirror* in the *init<SID>.sap* SAPDBA initialization file to `N`. This overwrites the standard configuration (transaction `DB17`).		
REDOLOG_MISSING	**E**	**Yes**
Cause: During the test to determine whether online redo logs (<n> groups, <m> members per group) can be accessed, R/3 determined that one online redo log cannot be accessed. This problem can have multiple causes. **Action:** First, check the file that cannot be accessed, and check the system for problems (network problems, server down, and so on). One possible action (**ONLY** for Oracle experts, as an error could result in an incomplete chain of offline redo logs which could prevent a full recovery of the database): If a second member of the group that cannot be accessed, exists, delete the non-accessible member. Using the Oracle Server Manager, execute `ALTER SYSTEM SWITCH LOGFILE`. Then, check if the status of the online redo logs in the table `V$logfile` is correct.		
TABLES_NOT_IN_TABLE_TABLESPACE	**E**	**No**

Cause: At least one of the tables is stored in a tablespace with type *INDEX*, *PURE INDEX*, *ROLLBACK*, *PURE ROLLBACK*, *TEMP* or *PURE TEMP*. This can become dangerous during certain database restoration actions (for example, for an ALL_DATA backup strategy) as these tablespaces can then no longer be easily restored. A tablespace is of the type *INDEX*, *ROLLBACK* or *TEMP*, if more than 75% of its segments are index, rollback, or temporary segments. A tablespace is of the type *PURE INDEX*, *PURE ROLLBACK* or *PURE TEMP*, if 100% of its segments are index, rollback, or temporary segments. **Action:** Use SAPDBA to reorganize (*Reorganize single table or index*) the affected tables. This will move the tables into a tablespace of type *TABLE* or *PURE TABLE*.		
TSP_BACKUP_MODE	**E**	**Yes**
Cause: The tablespace(s) (or their data files) listed in the message is (are) currently in backup mode. This reduces R/3 System performance. The check takes into account whether a BRBACKUP online backup is running. **Action:** Make sure no database backup (backup by BRBACKUP or an external tool) is running. Use the function *Check database (online quick check)* in SAPDBA to cancel the backup mode. SAPDBA will display a confirmation prompt for you to confirm you want to cancel the backup mode.		
TSP_FULL	**W**	**No**
Cause: The tablespace(s) listed in the message is (are) more than 90% full and must be extended in order to prevent a possible overflow. **Action:** Extend the tablespace using SAPDBA and *Tablespace Administration*.		
TSP_OFFLINE	**E**	**No**
Cause: The tablespace(s) named in the alert message is (are) offline. It (They) cannot be accessed. **Action:** Bring the corresponding tablespaces online by entering the command `ALTER TABLESPACE <tablespace> ONLINE` in the Oracle Server Manager.		

Type *DBO*:

Parameter	**Severity**	**Check during mount state**
DBO ALY	**E**	**No**
Cause: Errors occurred in `sapdba-analyze` sessions		

DBO ALY DBSTATCO	**E**	**No**
Cause: The age of the last successful `sapdba-analyze` session for the object `DBSTATCO` is older than the specified threshold value (default setting: >10 days)		
DBO NXT	**E**	**No**
Cause: The age of the last successful `sapdba-next` session is older than the specified threshold value (default setting: >10 days)		
DBO OPT	**E**	**No**
Cause: Errors occurred in `sapdba-checkopt` sessions		
DBO OPT PSAP%	**E**	**No**
Cause: The age of the last successful `sapdba-checkopt` session for the object `PSAP%` is older than the specified threshold value (default setting: >10 days)		
DBO RSI	**E**	**No**
Cause: There are errors in SAPDBA reorganizations of an individual table or an individual index.		
DBO RTC	**E**	**No**
Cause: There are errors in SAPDBA reorganizations of a tablespace without data files.		
DBO RTD	**E**	**No**
Cause: There are errors in SAPDBA reorganizations of a tablespace with data files.		

Action for all DBO alert messages: Check the schedule in the DBA Planning Calendar [Seite 1107] (Transaction `DB13`) and the operation logs in the DB Operations Monitor [Seite 1235] (Transaction `DB24`).

Type *ORA*:

Parameter	**Severity**	**Check during mount state**
ORA -272 error writing archive log	**E**	**Yes**
Cause: Error writing to archive log **Action:** Check to determine whether the storage medium is functioning as well as whether enough storage space is available.		
ORA -376 file <name> cannot be read at this time	**E**	**Yes**

Cause: The most likely cause is that the file is offline. **Action:** Use SAPDBA (function *Check database*) to check the status of the file. Bring the file online if necessary. Contact SAP technical support for additional help, as there are multiple potential causes for this problem.		
ORA -600 internal error code, arguments: [num], [?], [?], [?], [?], [?]	**E**	**Yes**
Cause: This is a general internal error for Oracle program exceptions. It indicates that a process has encountered a low-level, unexpected condition. This message can be caused by: timeouts, incorrect files, incorrect data checks in memory, hardware or I/O error, or incorrectly recovered files. **Action:** Report this message to Oracle's customer support service or to the SAP R/3 hotline.		
ORA –1113 file <name> needs media recovery	**E**	**Yes**
Cause: The database system attempted to open a data file that requires media recovery. The system could not access the file due to a physical hard disk drive error. **Action:** You must recover the database. However, choosing a recovery strategy requires a careful analysis of the failure. We recommend you contact SAP technical support for help with this problem.		
ORA –1115 IO error reading block from file <name> block <#> num	**E**	**Yes**
Cause: The disk where the file is stored is probably offline. **Action:** Bring the disk online. Then, execute the operation again. If you are unable to restore access to the file or are unsure how to proceed, we recommend you contact SAP technical support for help with this problem.		
ORA –1122 database file <name> failed verification check	**E**	**No**
Cause: The information in the datafile is not consistent with the information in the control file. **Action:** Make sure you are using the correct data file and control file for the database.		
ORA –1135 file name accessed for DML query is offline	**E**	**Yes**

Cause: A query failed because it attempted to access a data file belonging to an offline tablespace. An offline tablespace must be brought online to access its data. **Action:** Check the Oracle alert file in the directory displayed in the detail view of the database system check (Transaction *DB16*). Other Oracle messages may provide additional information about the problem. If you are unsure how to proceed, contact SAP technical support.		
ORA –1555 snapshot too old rollback segment number %s with <name> too small	**W**	**No**
Cause: Rollback segments required for consistent reading were overwritten by other database users. **Action:** Change the configuration of the rollback segments. Only execute this operation when the database load is low. For more information, see the SAPNet notes.		
ORA –1562 failed to extend rollback segment number <n>	**W**	**Yes**
Cause: The rollback segment named in the alert message could not be extended, as the tablespace `PSAPROLL` is full. **Action:** Use SAPDBA (*Tablespace extension*), to extend the tablespace `PSAPROLL`.		
ORA –1578 Oracle data block corrupted (file <name>, block <number>	**E**	**Yes**
Cause: The database system attempted to read a data file that contains corrupted data. **Action:** Recover the database. Choosing a recovery strategy requires a careful analysis of the failure. We recommend you contact SAP technical support for help with this problem.		
ORA –3113 end-of-file on communication channel	**E**	**No**
Cause: An unexpected end-of-file was processed on the communication channel. The problem could not be handled by the Oracle Net8, two task, software. This message could occur if the shadow two-task process associated with a Net8 connection has terminated abnormally, or if there is a physical failure (network or server offline). **Action:** If this message occurs during a connection attempt, check the setup files for the appropriate Oracle Net8 driver, and confirm Net8 software is correctly installed on the server. If the message occurs after a connection is established, and the error is not due to a physical failure, check if a trace file was generated on the server at the time of the failure. If a trace file exists, the error may be an internal Oracle error. In this case, you should contact Oracle's customer support or SAP technical support..		
Checkpoint not complete	**E**	**No**

Cause: After a log switch, data confirmed by *COMMIT* is written from the database buffer to the datafiles (synchronization). Oracle writes the alert message to the alert log, if the write session is still active at the next log switch. **Action:** If this alert message occurs often, there is a database performance problem. To correct the problem, enlarge the online redo log files. For more information, see the Oracle documentation.

Type *PROF*:

Parameter	Severity	Check during mount state
CONTROL_FILE_RECORD_KEEP_TIME	**E**	**No**
The amount of time (days) reusable areas in the control file are locked. Default setting: < 30		
CURSOR_SPACE_FOR_TIME	**W**	**No**
The setting, CURSOR_SPACE_FOR_TIME = FALSE, requires a considerably less amount of space in the Oracle shared pool. Default setting: <> FALSE		
DBWR_IO_SLAVES	**A**	**No**
Number of IO slaves used by the DBWR process. Default setting: <> 0		
DB_BLOCK_BUFFERS	**W**	**No**
Number of blocks in the database buffer cache. Default setting: < 8960		
DB_BLOCK_CHECKPOINT_BATCH	**W**	**No**
Number of buffers the DBWR process writes per write session during a checkpoint. Default setting: < 8		
DB_BLOCK_SIZE	**W**	**No**
Size (bytes) of the Oracle database blocks. Default setting: <> 8192		
DB_FILES	**W**	**No**
Number of data files Default setting: < 254		
DB_FILE_MULTIBLOCK_READ_COUNT	**W**	**No**
Number of blocks read during a complete tablescan I/O operation. Default setting: > 8		

DISK_ASYNCH_IO	**W**	**No**
The setting, DISK_ASYNCH_IO = TRUE, means that I/O operations are asynchronous for data files, control files and log files. Default setting: <> TRUE		
HASH_JOIN_ENABLED	**W**	**No**
The setting, HASH_JOIN_ENABLED = FALSE, means that the optimizer does not use any hash joins when calculating the access plan. Default setting: <> FALSE		
LOG_ARCHIVE_START	**E**	**No**
The setting LOG_ARCHIVE_START = TRUE means that automatic archiving is activated when the Oracle instance is started. Default setting: <> TRUE		
LOG_BUFFER	**W**	**No**
Size (KB) of the redo log buffer in the SGA. Default setting: >< 1100,300		
LOG_CHECKPOINT_INTERVAL	**W**	**No**
Number of the redo log blocks that trigger a new checkpoint. The default setting is extremely high, to ensure that a checkpoint is only triggered by a redo log switch. Default setting: < 3000000000		
LOG_CHECKPOINT_TIMEOUT	**W**	**No**
Time (seconds) between two checkpoints. Default setting: > 0		
LOG_SMALL_ENTRY_MAX_SIZE	**W**	**No**
Maximum size (bytes) of a copy in the redo log buffer that can be created using the redo allocation latch (without redo buffer copy latch). Default setting: <> 120		
OPEN_CURSORS	**W**	**No**
Maximum number of open cursors per user process. Default setting: >< 2000,800		
OPTIMIZER_FEATURES_ENABLED	**W**	**No**
Oracle release indicating which features of the optimizer are active. Default setting: <> 8.0.4		
OPTIMIZER_INDEX_COST_ADJ	**W**	**No**
Ratio (%) that influences the analysis of index scans Default setting: <> 10		

OPTIMIZER_MODE	**W**	**No**
Controls how the optimizer behaves (for example, rule-based = RULE, or cost-based = CHOOSE). Default setting: <> CHOOSE		
ROW_CACHE_CURSORS	**W**	**No**
Number of recursively-buffered cursors used by the row cache manager. Default setting: >< 1500,270		
SHARED_POOL_SIZE	**W**	**No**
Size (MB) of the shared pool in the SGA. Default setting: < 50		
SORT_AREA_RETAINED_SIZE	**E**	**No**
Size (KB) of the sort area in the PGA after a sort. This amount is the sort result. Default setting: >< 540,0		
SORT_AREA_SIZE	**E**	**No**
Size (KB) of the sort area in the PGA. Default setting: >< 10240,2048		
TIMED_STATISTICS	**E**	**No**
The setting, TIMED_STATISTICS = TRUE, creates time-related statistics. Default setting: <> TRUE		
TRANSACTION_AUDITING	**E**	**No**
The setting, TRANSACTION_AUDITING = FALSE, means no special REDO record (with session and user information) is created by the transaction layer. Default setting: <> FALSE		

Action for all PROF alert messages: Adjust the Oracle init<SID>.ora initialization file according to the *PROF* parameters types of the database system check (Transaction *DB17*).

SAPDBA stores comprehensive information about every database operation in a log. To display this log, use the DB Operations Monitor [Seite 1235]. Note any lines in the log that are highlighted in color as these indicate errors or warnings. In addition, the SAPDBA log also provides detailed information. For example, SAPDBA displays the complete distribution of tables for the check *TABLES_NOT_IN_TABLE_TABLESPACE*. This provides information to help you determine the correct target tablespace for a table.

Displaying Alert Messages from Database System Check

Use

Use the alert messages from the database system check to recognize existing and possible critical Oracle database situations in time.

Always deal with any alert messages from the database check system as quickly as possible. Alert messages with severity *E* are serious problems for which you must take immediate action.

Procedure

To display alert messages from the database system check, choose *Tools → CCMS → DB Administration → DB System Check → Display alerts.* Alternatively: Choose Transaction `DB16`.

In the group box:

- ***Check results*** includes the number of open warnings and errors.
- ***Default settings*** includes the current monitor options (such as update and delete cycles)
- In **View** you can change the display of alerts (*open* or *all*)

The alerts (warnings and errors) from the database system check is displayed in a table The table includes the following columns:

- **Result**

  Severity of the alert message:

  `A`: Exception due to a deviation from the SAP standard (for example, a parameter for a *PROF* alert message is not within the value range we recommend)

  `C`: An error occurred, but it has been fixed in the meantime

  `E`: Error (for errors of alert type *DBA* take immediate action)

  `W`: Warning

- **Date**

  Date of alert

- **Time**

  Time of alert

- **Days**

  Age (days) of alert

- **Error type**

Alert message type (*DBA*, *ORA* and *PROF*) to which the parameters of the database system check are assigned For more information, see Configuring Database System Check [Seite 1143]

- **Error name**

  Parameter for the database system check For more information, see Alert Messages from Database System Check [Seite 1150]

- **Text**

  Description of alert. This description is taken from the `SAPDBA -check` log. Long alert messages (such as those that contain lists of names) are displayed in shortened form. See the SAPDBA detail log or the Oracle alert log for the full message.

Activities

- **Automatically refreshing display**

  To automatically refresh the display of database operations, choose *Settings → Auto-refresh → Activate*. To deactivate automatic refresh of the display, choose *Deactivate*. To set the time interval for the refresh, choose *Settings → Auto-refresh → Time Interval*.

- **Starting database system check immediately**

  To start the database system check immediately, choose *Start check*

- **Configuring database system check (Oracle)**

  To configure the database system check, choose *Configure check*

- **Database operations monitor**

  To call the database operations monitor, choose *Database operations monitor*

- **Displaying details**

  To display detailed information about a line, place the cursor on the line and choose the *Details on selected line* icon

- **Displaying specific views**

  To display specific views (open alerts or all alerts), choose *Settings → View → Standard* or *History*. To set the time period for the view, choose *Settings → View → Time period*.

- **Displaying a print preview of the table**

  To display a print preview of the table, choose the *Print view* icon

- **Exporting table**

  To export the table to a different file format, choose the *Export* icon

- **Updating table display**

  To update the table display, choose the *Read check results again* icon

- **Manually deleting table entry**

  To manually delete a table entry, select the table entry and choose the *Delete selected alert* icon

- **Manually resetting a table entry**

To manually reset a completed error or warning alert, choose the *Reset selected alert*

- **Automatically deleting table entries**

  To automatically delete table entries, choose *Settings → Auto-delete → Activate* Choose *Deactivate* to deactivate automatic delete. To set the time interval for the delete, choose *Settings → Auto-delete → Time interval*.

- **Printing table entries**

  To print table entries, choose *Print*

- **Sort table entries**

  To sort table entries, choose either *Sort in ascending order* or *Sort in descending order*

- **Searching table entries**

  To search for table entries in the table, choose *Find*

- **Saving settings**

  To save all your settings (view, auto-refresh, and auto-delete), choose *Settings → Save*.

Using the DB Operations Monitor

Use

Use the DBA operations monitor for online monitoring of database operations within both internal and external database tools. You can also monitor runtime and remaining time of operations that are running. The DBA operations monitor can be used regardless of which database you are using. It provides historical as well as current (online) information about the following database operations:

- Backup/Recovery (for example, to back up or recover the database)
- Performance (for example, checking, creating, updating and deleting database statistics)
- The memory structure (for example, space information for database objects, reorganization of database objects, or extending and deleting database objects).
- Database checks (for example, checking the database for critical situations)
- Configuration (for example, configuring database parameters)

Prerequisites

To use the database operations monitor, the tables of the XDB interface must be correctly filled.

Features

To call the DBA operations monitor, choose *Tools* → *CCMS* → *DB administration* → *Operations monitor* Alternatively, call Transaction *DB24*.

In the group box *Overview* is an overview of the status of database operations that are running or have finished as well the current setup for the database operations monitor. The individual database operations are displayed in a table. The table includes the following columns:

Column	**Meaning**
RC	Result of the database operation: *Green* means “Everything OK” Yellow means “Warning” *Red* means “Error“
Status	Status of the operation: *STARTED* The operation started. However information about the operation is not yet available (for example, percent complete, or estimated remaining time). *RUNNING* The operation is running and information about the operation is available. *COMPLETED* The operation is finished
Date	Date of the operation

Time	Start time of the operation
FID	Type of operation
Object	Database objects that are affected by the operation or are key
Runtime	Runtime of the operation
Program	Program that executes the operation (for example, SAPDBA)
Description	Short description of the operation

Activities

Displaying specific database operations

To display specific database operations (for example, backup/recovery operations), choose the corresponding pushbutton, or the menu option under *DB operation*. Choose *All database operations* to display all DB operations.

Displaying specific views

To display specific views (for example, to display those database operations with status "warning", or those with status "error") choose *Setup* → *View*. Then choose either *Warning* or *Error*. Choose *All operations*, to display all database operations. To set the time period for the view, choose *Setup* → *View* → *Time period*.

Updating display

To update the display of database operations, choose *Edit* → *Refresh*, or choose *Refresh* on the application toolbar.

Automatically refresh display

To automatically refresh the display of database operations, choose *Setup* → *Auto-refresh* → *Activate*. To deactivate automatic refresh of the display, choose *Deactivate*. To set the time interval for the refresh, choose *Setup* → *Auto-Refresh* → *Time Interval*.

Deleting table entries

To delete database operations from the table, select the desired entries and choose *Edit* → *Delete*.

Automatically deleting table entries

To automatically delete table entries, choose *Setup* → *Auto-delete* → *Activate* Choose *Deactivate* to deactivate automatic delete. To set the time interval for the refresh, choose *Setup* → *Auto-delete* → *Time Interval*.

Sorting table entries

To sort columns in ascending or descending order, first select the relevant column. Then choose *Edit* → *Sort* and *Descending*, or *Ascending*. Alternately, choose *Sort in descending order* or *Sort in ascending order* on the application tool bar.

Printing table entries

To print table entries, choose *Print*

Saving settings

To save all your settings (view, auto-refresh, and auto-delete), choose *Setup* → *Save*.

Details about database operations

For details about operations (for example, remaining time for the operation, or directory and name of the log file), double-click the table entry, or select the table entry and choose *Display details* on the application toolbar. On the next screen, choose *Refresh* on the application toolbar to update the display. Use *Display action log* to display a detailed log for the database operation.

Maintaining Database Connection Information

Use

Use DB connection maintenance to maintain the R/3 table *DBCON*. This table contains information about additional non-standard database connections (such as database system, password, and so on).

Procedure

To call DB connection maintenance, choose *Tools* → *CCMS* → *DB administration* → *DB connection maintenance*. Alternatively, use Transaction `DBCO`.

A table for DB connection maintenance appears. The table includes the following columns:

- **Name of connection**

  Logical name of database connection

- **DBS**

  Database system

- **Database user name**

  Database user

- **Database password**

  Password to create the connection to the database

- **Connection information**

  Database system-dependent information about the database connection.

Displaying Details about the Database Connection

1. Select the entry
2. Choose `Details`

Adding New Information to Database Connections

1. To switch to change mode, choose `Table view` → `Display -> Change`
2. Choose `New entries` Alternatively, choose `Copy as...` to use an existing data connection as a template
3. In the `Connection name` field, enter a name you choose for the logical database connection
4. In the `DBMS` field enter the code for the database system (for example, *ORA* for Oracle)
5. In the `User name` field enter the name of the database user
6. In the `DB Password` field enter the password twice
7. In the `Conn.info` field enter the database-dependent information for the database connection

8. In the `Reconnect type` field enter the type of availability for an open database connection
9. Choose `Save`

Maintaining Existing Database Connection Information

1. To switch to change mode, choose `Table view` → `Display -> Change`
2. Select the entry
3. Choose `Details`
4. Change the existing information
5. In the `DB Password` field enter the password twice
6. Choose `Save`

Deleting Existing Database Connection Information

1. To switch to change mode, choose `Table view` → `Display -> Change`
2. Select the entry
3. Choose `Delete`

Maintaining Database Parameters: Oracle

Use

This function allows you to:

- Display and change the current Oracle database parameters online
- Check the status and validity of the Oracle database parameters
- Display the history of the Oracle database parameters

Procedure

To call DB parameter maintenance, choose *Tools* → *CCMS* → *DB administration* → *DB parameters*. Alternatively, call Transaction *DB27*.

The frame *Number of database profile parameters* includes the number of:

- All Oracle database parameters
- Oracle database parameters currently at their default value
- Database parameters modifiable by the system (command *ALTER SYSTEM ...*)
- Database parameters modifiable by the session (command *ALTER SESSION ...*)
- Parameters modified since the last database start
- Invalid database parameters whose current value is different than the value specified in the database system check

The individual database parameters are displayed in a table. The table includes the following columns:

Column	Meaning
Name	Name of the database parameters
Val. (value)	Value of the database parameter

Alarm	Status of the database parameter **Red** The value of the database parameter is different than the threshold value or not within the validity range specified in the database system check. The severity of this value is set to *Error*. **Yellow** The value of the database parameter is different than the threshold value or not within the validity range specified in the database system check. The severity of this value is set to *Warning*. **Green** The value of the database parameter is identical to the threshold value or within the validity range specified in the database system check or the parameter is not checked by the database system check. For more information, see Configure Database System Checks [Seite 1143]
Description	Oracle documentation for the database parameter
Modif.	Was the parameter modified since the last database start? **FALSE** The parameter has not been modified since the last database start. **MODIFIED** The parameter was modified using the command *ALTER SESSION ...* since the last database start. **SYS_MODIFIED** The parameter was modified using the command *ALTER SYSTEM ...* since the last database start.
Default	Is the parameter a default Oracle database system parameter? **TRUE** Default parameter in Oracle database system **FALSE** Not a default parameter in Oracle database system

Ses.modif.	Can the database parameter be modified during the session? **TRUE** The parameter can be modified during the session using the command *ALTER SESSION ...* **FALSE** The parameter cannot be modified during the session Modifying a parameter immediately affects the database user session in which the parameter was modified. In the SAP context this means that all SAP user sessions whose dialog steps are processed by this database work process will be affected by the change. Since this will not involve all SAP user sessions, changes to session-modifiable parameters are not supported.
Sys.modif	Is the database parameter system-modifiable? **IMMEDIATE** The parameter can be changed using the command *ALTER SYSTEM ...* meaning it takes affect for all current database user sessions and therefore for the entire SAP System. **DEFERRED** The parameter can be changed using the command *ALTER SYSTEM ...* and it takes affect for all future database user sessions (for example, for SAP application servers started after the change went into effect). **FALSE** The parameter is not system-modifiable In order to use the command *ALTER SYSTEM ...* you must have an authorization for *ALTER SYSTEM* for the database.
Adjust.	Was the parameter automatically modified by the database system? **TRUE** The parameter was automatically modified by the database system **FALSE** The parameter was not automatically modified by the database system

Activities

Sorting database parameters according to date

To move the last database parameters changed to the top of the table, choose *Sort date in ascending order*. To move the last database parameters changed to the bottom of the table, choose *Sort date in descending order*.

Displaying specific views of database parameters

Choose:

- To display all database parameters, choose *Selection criteria* → *All*
- To display all system-modified database parameters, choose *Selection criteria* → *System modifiable*
- To display all session-modified database parameters, choose *Selection criteria* → *Session modifiable*
- To display all default Oracle database system parameters, choose *Selection criteria* → *Default*
- To display all non-default Oracle database system parameters, choose *Selection criteria* → *Not default*
- To display all parameters modified since the last database start, choose *Selection criteria* → *Modified since start*
- To display all parameters checked by SAPDBA using the function *DB System Check* since the last database start, choose *Selection criteria* → *With SAPDBA check*
- To display all parameters with invalid values, choose *Selection criteria* → *Invalid values*

Updating database parameter display

To update the display of database parameters, choose *Refresh*

Displaying details and history for a database parameter

1. Select an entry
2. Choose *Details and history*

   In addition to the details and history for the database parameter, the system also displays the threshold value specified in the database system check.

Displaying history for all database parameters

Choose *Goto* → *Total history*

- To display only the database parameters modified since the SAP System was implemented, choose *With history*
- To display only the database parameters modified since a certain date, choose *Selection date*
- To display a history for all database parameters, choose *All parameters*

Changing database parameters

Only system-modifiable database parameters can be changed online. These parameters have either the value *IMMEDIATE* or *DEFERRED* in the *Sys.modif* column.

1. Select an entry
2. Choose *Change*
3. Change the value of the database parameter

4. Choose *Save*

To make the change effective even after a database restart, you must maintain the database parameter in the *init<SID>.ora* Oracle initialization file. This file is usually located in the directory *$ORACLE_HOME/dbs* (UNIX) or *%ORACLE_HOME%\dbs* (NT).

Displaying the *init<SID>.ora* Oracle initialization file

To display the Oracle initialization file *init<SID>.ora*, choose *Display Oracle profile*.

Monitoring the Oracle Database

Use

With the R/3 alert monitor [Seite 812], you can keep track of the following database-related issues:

- Space management [Seite 1176]**:** monitor tablespaces and segments.
- Performance [Seite 1179]**:** monitor how current the optimizer statistics are, as well as Oracle caches
- Backup/restore [Seite 1181]**:** monitor the amount of space remaining in the archiving directory and also whether backup is running
- R/3 consistency [Seite 1182]**:** check the consistency between the indexes, database objects, and database tables included in the ABAP Dictionary and those in the Oracle data dictionary.
- Running jobs [Seite 1184]**:** monitor jobs registered in the XDB interface
- Health [Seite 1185]**:** monitor miscellaneous system checks including `init<SID>.ora` settings and `alert_<SID>.log` messages

By configuring data collection tools, or methods, to run periodically, alerts are automatically updated and fed into the monitoring architecture. Analysis tools provide additional information about the alert conditions, and auto-react tools can be configured to automatically respond when an alert occurs.

Procedure

Starting the Oracle Database Monitor

1. To reach the alert monitor, use Transaction RZ20.
2. Expand the monitor set "SAP CCMS Monitor Templates."
3. Load the "Database" monitor (Use either *Monitor* → *Load Monitor*, F8, or double-click).
4. Expand the Oracle database monitor.

Each element of the Oracle database that is monitored is displayed as part of a hierarchical tree and is called a Monitor Tree Element (MTE), or node. Follow links from this page to see brief description of each of these nodes.

Some data collection tools, or methods, provide data for more than one node. Thresholds for these methods and how frequently they are scheduled to run can be configured through the node to which the method is attached. Unless otherwise noted, a node's collection and analysis methods are available from the node itself.

Space Management Alerts

Tablespaces

fullest tablespace

Cause of alert: A tablespace has a higher percentage full than the percentage specified in *Properties → Performance Attribute → Thresholds*. The collection method scans all tablespaces and reports the smallest free space percentage here.

What to do: If the reported free space percentage is too low, start the analysis method to identify which tablespace has the least amount of free space left. Add another datafile to this tablespace.

<tablespace name>

Cause of alert: Each tablespace is scanned on startup, by `SOTREE_build_space_tree`. This node's collection method determines each tablespace's percentage of free space. You can configure a different free space percentage threshold for each tablespace. For example, for a read-only tablespace that won't need extra free space, you could safely set this threshold to 0.

What to do: Determine the anticipated growth of an existing tablespace and add an additional datafile of an appropriate size.

If you add a tablespace, run the RSDBMON0 report from Transaction SE80 so the new tablespace will be recognized and represented as a new node in the Tablespaces subtree.

If you drop a tablespace, just delete the node representing that tablespace.

Segments

Segments with too few allocatable extents

Cause of alert: The number of segments with too few allocatable extents has exceeded the configured threshold. 'Too few' is defined in the properties of another monitor node, 'fewest allocatable extents for a segment.' All segments that would raise a yellow or red alert in that node ('fewest allocatable extents for a segment') will be counted and included in this node. Only segment's in SAPR3's schema whose `PCTINCREASE` is set to zero are considered.

To allow for a number of read-only segments, increase the threshold so an alert will not be raised if the number of segments with too few allocatable extents is equal to or less than the number of read-only segments.

What to do: To determine which segments are affected, run the analysis method, which will list all segments with fewer allocatable extents than specified in the node's thresholds. If the `NEXT` storage parameter for this segment is set too high, decrease it to allow the segment to allocate more extents. If `NEXT` is set as intended, add another datafile to the tablespace.

The collection method that resides at this node feeds other nodes as well.

Running the collection and analysis methods associated with this node is very resource intensive. To decrease impact on performance, don't run these methods more often than every few hours. Also, on larger databases, consider running the collection method as a background job to avoid the possibility of the method timing out in a dialog process.

Fewest allocatable extents for a segment

Cause of alert: The smallest number of extents left for any segment has fallen below the configured threshold. Only segment's in SAPR3's schema whose `PCTINCREASE` is set to zero are considered.

What to do: To determine which segments are affected, run the analysis method, which will list all segments with fewer allocatable extents than specified in the node's thresholds. If the `NEXT` storage parameter for this segment is set too high, decrease it to allow the segment to allocate more extents. If `NEXT` is set as intended, add another datafile to the tablespace.

Running the collection and analysis methods associated with this node is very resource intensive. To decrease impact on performance, don't run these methods more often than every few hours. Also, on larger databases, consider running the collection method as a background job to avoid the possibility of the method timing out in a dialog process.

Segments approaching MAX_EXTENTS

Cause of alert: The number of segments that trigger red or yellow alerts for the node 'fewest extents before max_extents' has exceeded the threshold configured for that.

If you have a certain number of read-only segments, you might want to increase the threshold, so an alert will not be raised if the number of segments approaching `MAX_EXTENTS` is equal or smaller to the number of read-only segments.

What to do: Increase the segment's storage parameter `MAX_EXTENTS` if you expect the segment to grow.

Running the collection and analysis methods associated with this node is very resource intensive. To decrease impact on performance, don't run these methods more often than every few hours. Also, on larger databases, consider running the collection method as a background job to avoid the possibility of the method timing out in a dialog process.

Fewest extents left before MAX_EXTENTS

Cause of alert: The smallest number of allocatable extents for any segments before `MAX_EXTENTS` is reached has fallen below the configured threshold.

What to do: If you expect the segment to grow and, therefore, allocated more extents, increase the segment's `MAX_EXTENTS` storage parameter.

Running the collection and analysis methods associated with this node is very resource intensive. To decrease impact on performance, don't run these methods more often than every few hours. Also, on larger databases, consider running the collection method as a background job to avoid the possibility of the method timing out in a dialog process.

Segments with non-zero PCTINCREASE

Cause of alert: Segments have a `PCTINCREASE` value other than zero. Because Oracle now allows a `MAX_EXTENTS` setting of “unlimited”, setting `PCTINCREASE` to something other than zero no longer makes sense. Setting `PCTINCREASE` to non-zero can potentially introduce unnecessary fragmentation into your tablespace. Only segments in SAPR3’s schema are considered.

(`PCTINCREASE` was important when the largest number for `MAX_EXTENTS` was still relatively small and based on Oracle’s block size. Since you should now always be able to avoid reaching `MAX_EXTENTS` simply by setting its value high enough, you no longer need `PCTINCREASE` for allocating larger and larger extents to avoid reaching `MAX_EXTENTS`. This no longer makes sense since you should always be able to avoid reaching `MAX_EXTENTS` by simply setting its value high enough.)

What to do: Evaluate whether the segment’s storage parameter can be set to zero.

Most allocated extents in any segment

Cause of alert: The number of allocated extents any segment has exceeded the configured threshold. Having few allocated extents per segment was significant when Oracle’s maximum `MAX_EXTENTS` value was relatively low based on Oracle’s block size. `MAX_EXTENTS` can now have a value of “unlimited,” eliminating this concern. Having many extents should not cause problems, assuming none of the segments in this tablespace have `PCTINCREASE` set to non-zero and that the `NEXT` storage parameter settings for all segments in this tablespace are either all the same or multiples of one another.

What to do: If you prefer segments with fewer allocated extents, you can use SAPDBA to reorganize the object.

Rollback segment extension failed

Cause of alert: The rollback segment named in the alert message tried unsuccessfully to allocate an extent of the size specified by the rollback segment’s `NEXT` storage parameter. This allocation failed because the rollback segment is stored in a tablespace with no free extent large enough. Since rollback segments should be stored in tablespace `PSAPROLL`, this is most likely the tablespace affected.

What to do: If the rollback segment’s `NEXT` storage parameter is set higher than intended, decrease its value. Otherwise, use SAPDBA to add a datafile to the affected tablespace (most likely `PSAPROLL`).

Data is supplied to this node by a pair of methods. The first, which is very resource intensive and should be run sparingly, runs `sapdba -check`, completing checks that have been activated in Transaction DB17 [Seite 1143]. The second method runs frequently to ensure that the latest results of these checks, including results from runs of `sapdba -check` started outside the monitor, will be fed to this monitor node.

Database Performance Alerts

Optimizer

last statistic check

Cause of alert: Number of days since the last `sapdba -checkopt PSAP%` has been run exceeds the configured threshold. If this has never been run, an additional node is created under optimizer to represent this since reporting 0 days would be misleading/incorrect.

What to do: Run `sapdba -checkopt PSAP%`. This determines the need for new optimizer statistics. For more information, see CCMS Support for the Cost-Based Optimizer [Seite 1132].

last analyze

Cause of alert: The number of days since the last `sapdba -analyze PSAP%` or `sapdba -analyze DBSTATCO` was run, exceeds the configured threshold. If this has never been run, an additional node is created under optimizer to represent this since reporting 0 days would be misleading/incorrect.

What to do: Run `sapdba -analyze PSAP%` or `sapdba -analyze DBSTATCO` to update optimizer statistics. For more information, see CCMS Support for the Cost-Based Optimizer [Seite 1132].

Buffers

buffer cache

Cause of alert: The buffer cache hit ratio fell below the configured threshold.

What to do: Increase the size of the buffer cache only if the previous size increase improved the buffer cache hit ratio. For most applications, the buffer cache hit ratio should be above 90%. Applications that mostly execute long table scans cannot benefit as much from the buffer cache since such applications tend to overwrite the buffer cache. For additional information on how to tune the buffer cache, refer to Oracle's documentation.

library cache

Cause of alert: The library cache hit ratio fell below the configured threshold.

What to do: Increase the size of the shared pool in which the library cache resides until the `V$LIBRARYCACHE.RELOADS` value is near 0. This is done by increasing the value of the `init<SID>.ora` parameter `SHARED_POOL_SIZE`. The application should use identical SQL statements whenever possible. For additional information, refer to Oracle's Tuning guide.

redo log buffer

Cause of alert: The number of redo entries per redo log space request fell below the configured threshold. Oracle recommends at least 5000 redo entries per redo log space request.

What to do: Increase the size of the redo log buffer by increasing the `init<SID>.ora` parameter log_buffer until the number of redo entries per redo log space request stops increasing.

Checkpoints

checkpoint not complete

Cause of alert: After a log switch, data is written from the database buffer into the datafiles, or synchronized. This error occurs if the next log switch is activated before the data has been written completely to the data files.

What to do: If this error occurs often, there is a database performance problem. To correct the problem, enlarge the online redo log files. For more information, see the Oracle documentation.

Database Backup/Restore Alerts

Archiving

archiver destination full

Cause of alert: The number of archive logs that will still fit into the archiver destination is below the preconfigured threshold.

What to do: Either move some of the archive logs to a different location or add more space to the archiver destination to accommodate more archive logs.

archiving off

Cause of alert: Your database is running, but online redo logs are not being archived.

What to do: Using SAPDBA, restart the database in archivelog mode. *Choose Archive mode → Toggle database log mode*. SAPDBA shuts down the database, reconfigures the archiving mode, and the restarts the database.

Backup status

tablespace in backup mode

Cause of alert: The tablespaces named in the alert message are currently in backup mode, which reduces R/3 performance.

What to do: Confirm that no database backup is running, whether BRBACKUP or an external tool. Then use the function *Check database (online quick check)* in SAPDBA to cancel the backup mode.

Database R/3 Consistency Alerts

Objects missing in database

primary indexes, secondary indexes, tables, views

Cause of alert: Some ABAP Dictionary objects were not found in the database. If an object was intentionally deleted from the databases during development or if a database error occurred during creation, this missing object list can be significant for customer objects. Objects not being found in the database can lead to incomplete or incorrect transports and cancelled conversions.

What to do: Unless there has been either an incomplete transport or a canceled conversion, simply create the object in the database. For more information, see Consistency Checks [Seite 227].

Unknown objects in ABAP Dictionary

database tables, database indexes, database views, database tables without unique index

Cause of alert: Objects were found in the database that are not defined in the ABAP Dictionary. Since these objects do not belong to the R/3 System or cause problems, this list is primarily informational. These objects are usually created directly via the database. If the author is unknown, the purpose of the object can sometimes be determined using the display function. Because the data display in R/3 is based on ABAP Dictionary definitions, data can be displayed only via the database.

What to do: Customer tables, views and indexes should always be created with the ABAP Dictionary. This is the only way to ensure that the objects are recognized by the R/3 System and that problems such as collisions can be avoided during upgrading.

Inconsistent objects

primary indexes, secondary indexes, tables, views

Cause of alert: Objects exhibit inconsistencies between their ABAP Dictionary definitions and their database data dictionary definitions.

What to do: Compare the object's definition in the ABAP Dictionary with that in the database data dictionary and update the incorrect one.

Other checks

primary indexes, secondary indexes, tables, views, primary index not unique

Cause of alert: Several checks are performed, including whether the primary indexes of tables defined in the ABAP Dictionary were created uniquely in the database and whether there are objects in the R/3 base tables that either cannot be described at all or cannot be completely described in the ABAP Dictionary.

Optional indexes

too many indexes created, indexes not created

Cause of alert: Indexes defined in the ABAP Dictionary as optional database indexes have been found.

All the indexes which are defined in the ABAP Dictionary but for which a flag is set that they should not be created on the database to their system configuration or database system are displayed here as well.

What to do: Determine if these indexes have either been created or not been created as intended.

Running Jobs Alerts

Cumulative job log

job log

Cause of alert: This summary node provides a quick, high-level view of all jobs that have run or are currently running.

What to do: Open this node to review the single message and alert status of any job. To follow up on a problem with any particular job, note that job's unique identifier* and review the Job details node below for that identifier.

* *nnn.YYYYMMDDhhmmss,* where *nnn* is the abbreviated name of the function type and the remaining part is a time stamp.

Job details

<nnn.YYYYMMDDhhmmss>

Cause of alert: Any job that has not completed successfully, including jobs that are still in progress, will have a node here. Each of these, in turn, has two subnodes: the first describes the three-letter function code from the beginning of the job's identifier, the second describes in more detail the job's status, including providing access to a list of all alerts generated for that job.

What to do: Once the alerted problem is resolved, acknowledge the alert here, then delete the node.

Database Health Alerts

Database files

missing control file

Cause of alert: The system cannot access at least one of the control files (SAP recommends that three be accessible). If the database system is still running, at least one copy is still online.

What to do: Look for the parameter CONTROL_FILES, in the init<SID>.ora initialization file to determine which directories your control files are stored in. Make sure you can display all the files using the file manager (command ls). Restore access to any non-accessible files.

controlfile mirror missing

Cause of alert: There are fewer than the three recommended copies of the control file in the init<SID>.ora initialization file. SAP recommends you make two mirrored copies of the control file for safe operation. This message is often generated because there is a copy of the control file on the hardware level, but this fact is not indicated in the init<SID>.ora file. For safe operation, SAP recommends that you either use Oracle mirroring (multiple init<SID>.ora entries) or both hardware and Oracle mirroring.

What to do: Edit the init<SID>.ora file and add the additional control files under the parameter CONTROL_FILES. SAP recommends you have at least three instances of the file. Each instance of the file must be stored on a separate physical hard disk. After you edit the profile, restart the database to activate your changes.

dbfile offline

Cause of alert: The system cannot access the data file specified in the message, because this file is offline. All files in a SAP database should normally be online.

Files that are offline will not be part of database processing nor part of a database restore.

What to do: Close all applications and enter the following commands using the Oracle Server Manager:

```
1. connect internal
2. shutdown immediate
3. startup mount
4. alter database datafile <data file> online
5. alter database open
```

Alternatively, use the SAPDBA function, see SAPDBA Check (and Repair) Database [Extern]

file mismatch

Cause of alert: The file type for a database file specified in the control file does not match the entry in the Data Dictionary of the Oracle database system. This error is usually caused by errors during processing of the control file or during restoration of the control file.

What to do: Execute `CREATE CONTROLFILE` using the SQLDBA tool, specifying either the correct file or no file at all.

dbfile missing

Cause of alert: At least one data file is missing (file cannot be accessed). This error occurs if a file has the status *Recovery* and the file either cannot be accessed or the file is too old.

What to do: Using SAPDBA, execute a `Restore/recovery` to partially or fully recover the database.

file type unknown

Cause of alert: The file type (data file, raw device, dir, link) is unknown.

What to do: Troubleshoot to determine the cause of the problem. If you are not sure how to proceed, contact SAP technical support.

redolog not mirrored

Cause of alert: Your online redo logs are not mirrored. Mirroring is required for safe operation.

What to do: Using the *SQLDBA* tool, activate mirroring of the online redo logs in your database, with the statement `ALTER DATABASE ADD LOGFILE`.

redolog missing

Cause of alert: During the test to determine whether online redo logs (<n> groups, <m> members per group) can be accessed, R/3 determined that one online redo log cannot be accessed. This problem can have multiple causes.

What to do: First, check the file that cannot be accessed, and check the system for problems (network problems, server down, and so on). One possible action (**ONLY** for Oracle experts, as an error could result in an incomplete chain of offline redo logs which could prevent a full recovery of the database): If a second member of the group that cannot be accessed, exists, delete the non-accessible member. Using the *SQLDBA* tool, execute `ALTER SYSTEM SWITCH LOGFILE`. Then, check if the status of the online redo logs in the table `V$logfile` is correct.

tablespace offline

Cause of alert: The tablespace(s) named in the alert message is (are) offline. It (They) cannot be accessed.

What to do: Bring the tablespace(s) online, using the SQLDBA tool with the command `SET ONLINE` (`DROP IF NOT NEEDED`).

dbfile offline (ORA-1135)

Cause of alert: A query failed because it attempted to access a data file belonging to an offline tablespace. An offline tablespace must be brought online to access its data.

What to do: Check the Oracle alert file in the directory displayed in the detail view of the database system check (Transaction DB16). Other Oracle messages may provide additional information about the problem. If you are unsure how to proceed, contact SAP technical support.

Read/write errors

I/O error

Cause of alert: The disk where the file is stored is probably off line.

What to do: Bring the disk online. Then, execute the operation again. If you are unable to restore access to the file or are unsure how to proceed, SAP recommends you contact technical support for help with this problem.

file failed verification

Cause of alert: The information in the datafile is not consistent with the information in the control file.

What to do: Make sure you are using the correct data file and control file for the database.

corrupted block

Cause of alert: The database system attempted to read a data file that contains corrupted data.

What to do: Recover the database. Choosing a recovery strategy requires a careful analysis of the failure. SAP recommends that you contact technical support for help with this problem.

error writing archive log

Cause of alert: An error occurred while writing to the archive log.

What to do: Check to determine whether the storage medium is functioning as well as whether enough storage space is available.

file cannot be read

Cause of alert: The most likely cause is that the file is offline.

What to do: Use SAPDBA's *Check database* function to check the status of the file. Bring the file online if necessary. Contact SAP technical support for additional help, as there are multiple potential causes for this problem.

Init.ora settings

dbwr_io_slaves

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Number of IO slaves used by the DBWR process. Default setting: <> 0

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

db_block_buffers

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Number of blocks in the database buffer cache. Default setting: < 8960

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

db_block size

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Size (bytes) of the Oracle database blocks. Default setting: <> 8192

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

db_file_multiblock_read_count

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Number of blocks read during a complete tablescan I/O operation. Default setting: > 8

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

log_archive_start

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

The setting `LOG_ARCHIVE_START = TRUE` means that automatic archiving is activated when the Oracle instance is started. Default setting: <> `TRUE`

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

log_buffers

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Size (KB) of the redo log buffer in the SGA. Default setting: >< 1024,320

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

log_checkpoint_interval

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Number of the redo log blocks that trigger a new checkpoint. The default setting is extremely high, to ensure that a checkpoint is only triggered by a redo log switch. Default setting: < 3000000000

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

log_checkpoint_timeout

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Time (sec.) between two checkpoints. Default setting: > 0

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

open_cursors

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Maximum number of open cursors per user process. Default setting: >< 2000,800

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

optimizer_mode

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Controls how the optimizer behaves (for example, rule-based = RULE, or cost-based = CHOOSE). Default setting: <> CHOOSE

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

row_cache_cursors

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Number of recursively-buffered cursors used by the row cache manager. Default setting: >< 1500,270

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

shared_pool_size

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Size (MB) of the shared pool in the SGA. Default setting: < 50

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

sort_area_retained_size

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Size (KB) of the sort area in the PGA after a sort. This amount is the sort result. Default setting: >< 540,256

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

sort_area_size

Cause of alert: Current setting is outside the range of values preconfigured for this check in Transaction DB17 [Seite 1143].

Size (KB) of the sort area in the PGA. Default setting: >< 10240,2048

What to do: Verify that this value is set as intended. If so, readjust the preconfigured range of values in Transaction DB17.

Miscellaneous errors

DBA_600

Cause of alert: This is a general internal error for Oracle program exceptions. It indicates that a process has encountered a low-level, unexpected condition. This message can be caused by: timeouts, incorrect files, incorrect data checks in memory, hardware or I/O error, or incorrectly recovered files.

What to do: Report this message to Oracle's customer support service or to SAP technical support.

snapshot too old

Cause of alert: Rollback segments required for consistent reading were overwritten by other database users.

What to do: Change the configuration of the rollback segments. Only execute this operation when the database load is low. For more information, see notes in SAPNet.

end-of-file communication channel

Cause of alert: An unexpected end-of-file was processed on the communication channel. The problem could not be handled by the Oracle Net8, two task, software. This message could occur if the shadow two-task process associated with a Net8 connect has terminated abnormally, or if there is a physical failure (network or server offline).

What to do: If this message occurs during a connection attempt, check the setup files for the appropriate Oracle Net8 driver, and confirm Net8 software is correctly installed on the server. If the message occurs after a connection is established, and the error is not due to a physical failure, check if a trace file was generated on the server at failure time.

If a trace file exists, the error may be an internal Oracle error. In this case, you should contact Oracle's customer support or SAP technical support..

file needs media recovery

Cause of alert: The database system attempted to open a data file that requires media recovery. The system could not access the file due to a physical hard disk drive error.

What to do: You must recover the database. However, choosing a recovery strategy requires a careful analysis of the failure. SAP recommends that you contact technical support for help with this problem.

SAP/Informix DBA in CCMS

Purpose

This component enables you to manage your Informix database using the Computing Center Management System (CCMS). With CCMS, you get extensive support in database administration (DBA) for the Informix database and can perform many DBA functions from within the R/3 System.

Implementation Considerations

SAP recommends you to use CCMS for Informix DBA where possible. CCMS is supported for Informix databases on both UNIX and Windows NT platforms.

For each area of administration, the table below in "Integration" shows the available tools. In general, you should use CCMS or SAPDBA as first choice, followed by the other Informix tools. The reasons for this are as follows:

- CCMS deploys the familiar R/3 interface, can be used directly from your R/3 session and is perfectly adequate for many routine functions.
- SAPDBA is tailored for use with the R/3 System running on Informix databases and can also be used when the R/3 System is down. With SAPDBA, you can perform a wide range of DBA functions (but not archive and backup).
- The Informix tools have the disadvantage that they are not designed specifically to run with the R/3 System, and furthermore some of these tools have a less advanced interface than SAPDBA or CCMS.

  There are some overlaps in functionality between CCMS and SAPDBA. In general, however, they complement one another, as their strengths lie in different areas. For example, CCMS is more suited for shared memory parameters, whereas SAPDBA is better for monitoring and tuning in the area of space management.

  These are only guidelines as to the best tool for the task. The exact nature of the task determines which tool you should use.

Integration

There are many different tasks involved in Informix database administration, only some of which you can carry out using CCMS, as shown in the following table:

Area of administration	Can be performed using
Installation	R3SETUP
Archive and backup	CCMS, ontape, ON-Archive, ON-Bar
Reorganization	SAPDBA
Update statistics	CCMS, SAPDBA, Informix tools
Performance tuning	CCMS, SAPDBA, Informix tools
Monitoring	CCMS, SAPDBA, Informix tools

Space management	SAPDBA, Informix tools
System checks	SAPDBA, Informix tools, CCMS
Other	SAPDBA, Informix tools

Features

The main features for Informix DBA in CCMS are as follows:

Area of administration	**Can be performed in CCMS using**
Scheduling archive, backup, update statistics, DB system checks, physical consistency checks, and other tasks	DBA Planning Calendar [Seite 1195]
Reviewing results of archive and backup	Archive and Backup Monitor in CCMS [Seite 1232]
Update Statistics	Update Statistics [Seite 1220]
Performance tuning and monitoring	Database Monitor [Seite 424] and Database Alert Monitor [Seite 1238]
System checks (that is, configuration and performance)	DB System Check [Seite 1226]

There is some overlap between these tools.

Constraints

You have to perform certain DBA functions outside the R/3 System, that is, using tools supplied by Informix and SAP. For example, to perform a reorganization, you have to use SAPDBA.

See also:

BC R/3 Database Guide: Informix [Extern]

Getting Started in CCMS with Informix DBA

Use

Before starting to use the Computing Center Management System (CCMS) for Informix database administration (DBA) tasks you need to set up certain things.

Procedure

1. In the R/3 System, check that:
   a. You have authorization for database administration and background job scheduling.

      The profiles `S_RZL_ADMIN` and `S_BTCH_ALL` must be entered for the administrator. Refer to Profile Maintenance [Seite 278].

   b. External programs are able to run on the database server so that actions on the database can be performed from other application servers.

      Refer to Background Processing [Seite 74] and Note 8523 in SAPNet – R/3 Frontend.
2. In the Informix database management system (DBMS), check the following:

   The DBA Planning Calendar in CCMS uses the Informix data recovery tool `ON-Bar` or `ON-Archive`. Therefore, you must correctly configure your chosen tool to successfully use the DBA Planning Calendar:

   – `ON-Bar`

     Refer to Configuring ON-Bar [Extern].

   – `ON-Archive`

     Refer to Configuration of On-Archive [Extern].

     You must make sure that storage devices (for example, tape drives) for archive and backup always have the correct empty tape volume. Otherwise, even if all other preparations and schedules are correct, no data can be secured and you risk losing data in the event of a system failure.

Result

You can now use CCMS to perform tasks for the Informix database. For example, you can now use the DBA Planning Calendar (Informix) [Seite 1197].

See also:

ON-Bar for Data Recovery [Extern]

ON-Archive for Data Recovery [Extern]

DBA Planning Calendar

Use

The DBA Planning Calendar is part of the Computing Center Management System (CCMS) in the R/3 System. You can use it to schedule and execute certain database administration (DBA) tasks, such as backups, updating statistics for the cost-based optimizer, database system checks, and so on. The tasks available differ according to the database platform but the method of use is the same.

The main functions of the DBA Planning Calendar are as follows:

- Scheduling of DBA tasks using the following methods:
  - Predefined action patterns
  - Scheduling that you define yourself
- Change or deletion of tasks already scheduled
- Execution of DBA tasks as follows:
  - At scheduled time
  - Immediately
- Viewing of result logs (not available for all actions) and job logs

Integration

The DBA Planning Calendar is one of a number of tools available in CCMS for performing DBA tasks. Refer to Database Administration in CCMS [Extern]. The DBA Planning Calendar is available for all database platforms supported by SAP with the exception of DB2/400 (which does not require the Planning Calendar, because DB2/400 provides good equivalent tools.) It is delivered as a standard part of the R/3 System.

Features

The features implemented for the Informix database management system (DBMS) are as follows:

DBMS	Feature
Informix	• Database archive (with ON-Archive) or database backup (with ON-Bar) • Logical-log backup (with ON-Archive or ON-Bar) • Check and update statistics for cost-based optimizer (CBO) • System checks (that is, configuration and performance checks) • Physical consistency checks

Activities

1. You choose a predefined action pattern (or make up your own schedule).
2. If you want to alter the schedule (for example, add new tasks, delete or change existing tasks), you can do so at any time before the scheduled time.

3. You make sure that the resources required to complete each task (for example, tape devices) are correctly set up.

   The system automatically performs the task at the scheduled time (or immediately if you want).

4. You look at the action and job logs to check the results.

   For more information about activities, see DBA Planning Calendar: Informix [Seite 1197].

DBA Planning Calendar (Informix)

Purpose

You can schedule a variety of database administration (DBA) tasks with the Informix database. For more information, see "Features" in DBA Planning Calendar [Seite 1195].

Prerequisites

For more information about what you need to do before using the DBA Planning Calendar for the first time, see Getting Started in CCMS with Informix DBA [Seite 1194].

Process Flow

1. You schedule the activities you need, preferably choosing a predefined action pattern. Refer to Choosing an Action Pattern in the DBA Planning Calendar (Informix) [Seite 1199]. If you want to set up your own schedule, see the next step.

   By using an action pattern, you are following SAP's guidelines for database activities that can be carried out from the CCMS. An action pattern implements a backup strategy and other database administration activities that need to be regularly performed.

   When you choose an action pattern, the system enters the required activities into the DBA Planning Calendar. It also schedules the background jobs that carry out the activities.

   You should perform the following tasks **daily**.

2. If needed, you edit the schedule to make any required changes to activities or start times. For example, you might want to add a new job on a certain day or time, or delete a pre-scheduled job. Refer to Scheduling Actions in the DBA Planning Calendar [Seite 1201].

3. You make sure that the resources required to complete each task are available. Typically, you need to make sure that:

– Storage devices (for example, tape drives) are correctly set up.

– New media (for example, tapes) are correctly initialized.

– The required media are inserted in the correct storage devices.

4. You check the result by looking at the action and job logs. Refer to Checking the Results of Actions in the DBA Planning Calendar [Seite 1204]. For archives and backups, you can also use the archive and backup monitor. Refer to Using the Archive and Backup Monitor in CCMS (Informix) [Seite 1233].

   If any action did not run successfully, analyze the problem and repeat the action if necessary. Refer to Troubleshooting in the DBA Planning Calendar (Informix) [Seite 1219].

The information above is an overview of how to use the DBA Planning Calendar. For more information about how to perform actions specific to the Informix database, see the following:

- Archiving or Backing Up the Database in the DBA Planning Calendar (Informix) [Seite 1207]
- Backing up the Logical Log in the DBA Planning Calendar (Informix) [Seite 1209]
- Backing up the Logical Log (Automatic) in the DBA Planning Calendar (Informix) [Seite 1211]
- Checking Statistics in the DBA Planning Calendar (Informix) [Seite 1213]
- Updating Statistics in the DBA Planning Calendar (Informix) [Seite 1214]
- Checking Physical Consistency in the DBA Planning Calendar (Informix) [Extern]
- Checking the DB System in the DBA Planning Calendar (Informix) [Seite 1217]

See also:

SAP/Informix DBA in CCMS [Seite 1192]

BC R/3 Database Guide: Informix [Extern]

Choosing an Action Pattern in the DBA Planning Calendar

Use

The DBA Planning Calendar [Seite 1195] provides easy-to-use predefined action patterns specific to each database platform. You specify a reference time, on the basis of which all schedules are defined. It is possible later to delete an action pattern. For more information about how to change actions in a pattern, see Scheduling Actions in the DBA Planning Calendar [Seite 1201]. However, SAP recommends you to use a predefined action pattern.

Prerequisites

You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].

Procedure

1. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar* → *Calendar* → *Action Pattern*.
2. Select a predefined action pattern.
3. Enter the time at which the key action is to be carried out.

   The system suggests an appropriate time, which you can accept if you want. The system uses this time to work out the schedule for the activities in the action pattern.

   If there are conflicts between the action pattern you have chosen and activities that are already scheduled in the Planning Calendar, then the system presents a list of the conflicts.
4. If there are conflicts, do the following:

   a) Print the list with the `Shift-F1` key combination. Then choose *Cancel*. No activities from the new action pattern are scheduled.

   b) Review and eliminate the conflicts before trying to schedule the action pattern again.
5. To delete a predefined action pattern, you have to delete the next scheduled occurrence of the action that was scheduled as part of an action pattern. All future scheduling of the action is deleted.
6. Make sure that the required resources are available when an action is scheduled to run.

Result

The activities in the action pattern are automatically inserted into the planning calendar. The system also schedules background jobs for executing the activities. All jobs are scheduled for periodic repetition according to the schedule in the action pattern.

See also:

Checking the Results of Actions in the DBA Planning Calendar [Seite 1204]

DBA Planning Calendar (Informix) [Seite 1197]

Scheduling Actions in the DBA Planning Calendar

Use

This section tells you how to schedule actions in the DBA Planning Calendar [Seite 1195], which is part of the Computing Center Management System (CCMS).

Prerequisites

- You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].
- If you want to change or delete an action, it must be in the state *SCHED* (that is, not already executed).
- If you want to insert an action, you must choose today or a later day, and if you choose today, you must choose a time after the current time.
- If an action has already been executed, you can only display it. See Checking the Results of Actions in the DBA Planning Calendar [Seite 1204].

Procedure

1. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar* to start the DBA Planning Calendar.
2. Choose the day you want, by double clicking on the day's header bar.

   The system displays actions already scheduled on the chosen day.
3. To insert a new action, do the following:

   a) Choose *Create action*.

   The system displays the actions supported by the Planning Calendar for your database platform.

   b) Select the action you want to schedule.

   The system shows the basic parameters currently set for the action.

   c) Enter the basic parameters for the action as follows:

Parameter	What to enter	For example
Start time	• The time when the action is to start, using 24-hour clock notation. • Choose *Start immediately*, if you are entering an action for today and want to start the action immediately.	`17:00` This means the job is to be executed at 5 o'clock in the afternoon.
Period	The interval for the action, in weeks. The action is repeated at the interval you enter. If you do not enter a value, the action is run once only.	`2` This means the action is to be repeated on the same day and time every two weeks.

Calendar	Select the calendar for your country or area.	US This means the calendar for the United States is to be used.

The system warns you if there is a conflict with an existing action. If so, you must choose another time for the action.

Depending on the action you are inserting, the system may prompt for further input parameters.

4. If necessary, enter further input parameters.
5. Choose *Continue* to insert the action.
6. To change an existing action, do the following:

   a) Select the action you want to change.

   b) Choose *Change action*.

   The system shows the basic parameters currently set for the action.

   c) If required, change the basic parameters for the action. Refer to the table shown in the previous step.

   The system warns you if there is a conflict with an existing action. If so, you must choose another time for the action.

   d) If required, choose *Parameters* to change the parameters specific to the action (for example, the tape volume to use for a backup).

   e) Choose *Continue* to save your changes.
7. To delete an existing action, do the following:

   a) Select the action you want to delete.

   b) Choose *Delete action*.

   The system asks you to confirm the deletion.

   c) Confirm the deletion.

   The system deletes the action, including the corresponding background job. Deletion of the job also stops automatic periodic repetition of the action, if that was scheduled.

   If an action is one of a sequence, you can only change or delete the **next** scheduled occurrence of the action. If you do this, the system also deletes all future occurrences of the action in the same sequence.

   For example, you cannot change or delete an action scheduled to run in six weeks' time, if the next action of the same sequence is scheduled to run next week. Instead, you have to change or delete the occurrence for next week.
8. If you have inserted a new action or changed an existing one, make sure that any resources required by your change or insertion are available.

Result

The schedule of the DBA Planning Calendar is updated with the results of your insertion, change, or deletion.

See also:

Choosing a Predefined Action Pattern in the DBA Planning Calendar [Seite 1199]

DBA Planning Calendar (Informix) [Seite 1197]

Checking the Results of Actions in the DBA Planning Calendar

Use

This section tells you how to check the results after an action has been executed in the DBA Planning Calendar [Seite 1195]. Actions executed in the DBA Planning Calendar generate the following:

- Actions logs (not available for some actions)
- Job logs (available for all actions)

You can view both of these in the DBA Planning Calendar.

Background jobs are used to schedule actions and these background jobs generate job logs. You can view all the information using the Calendar. In addition, you can view all previous archive and backup actions with the archive and backup Monitor.

Prerequisites

You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].

Procedure

To display the current state of an action (past, current, or previous) in the Calendar, do the following:

1. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar* to start the DBA Planning Calendar.

   The system displays the calendar, with a list of actions for each day. The actions are color-coded, with the following meanings:

Action Color Coding in the DBA Planning Calendar

Color	Meaning
Blue	Action scheduled for execution today
Green	Action completed successfully
Yellow	Database activity log missing or warning issued
Red	Action failed – analyze problem and repeat action if necessary to ensure availability of a complete backup or continued safe operation of the database

2. Choose the day for the action you want to look at.

   The system displays a list of actions for the day you selected.

3. Look at the field *State* for the action.

   The following values are possible:

Action States in the DBA Planning Calendar

Status	Meaning
SCHED	Action is scheduled but the scheduled time has not yet arrived
OK	Action was completed successfully
ERROR	Action was not completed successfully
EXEC	Action is being carried out
N.A.	No information available on state (for example, if no action log normally written for the action)

The statuses *ERROR* and *N.A.* are highlighted in red in the Calendar. If there is one *ERROR* in one day and no OKs, then the header for that day is also highlighted.

4. Choose *Action Logs* to look at the action log.

   The system displays the action log, if there is one. The system does not generate an action log for every type of action.

   A job log (see next step) contains general information, whereas an action log contains specific information about the database aspect of the action. You normally check the action log. If there is a problem with the job, you should also check the job log.

   Action logs are sometimes assigned to a later action than they should be, due to problems with the timestamp used to assign logs to scheduled actions. In some cases the action is delayed and even postponed until after the next scheduling time.

   This occurs, for example, if no batch work process is available. An action log is then assigned to the action that has the same type and the closest corresponding timestamp. Unfortunately this means that the action log is then assigned to the next scheduling time and the log for the originally scheduled time is incorrect.

5. Choose *Job Logs* to display the job log for an action. For example, this is recommended if there is a problem.

The system displays the job log.

6. Choose *Environment* → *Backup Logs* for the archive and backup monitor, where you can see the results of archives and logical-log backups. For more information, see Using the Archive and Backup Monitor in CCMS: Informix [Seite 1233].

7. If there is a problem, see Troubleshooting in the DBA Planning Calendar (Informix) [Seite 1219].

Result

By checking the results of actions executed in the DBA Planning Calendar, you can make sure that your database is functioning optimally. For example, you can tell if a database backup has failed, fix the problem, and schedule it again immediately. Therefore, you can avoid downtime for your R/3 System.

See also:

Choosing a Predefined Action Pattern in the DBA Planning Calendar [Seite 1199]

Scheduling Actions in the DBA Planning Calendar [Seite 1201]

Background Processing [Seite 74]

DBA Planning Calendar (Informix) [Seite 1197]

Archiving or Backing Up the Database in the DBA Planning Calendar (Informix)

Use

You can use the DBA Planning Calendar [Seite 1195] in the Computing Center Management System (CCMS) to schedule database backups (ON-Bar) or archives (ON-Archive) for the Informix database. For more information about database backup or archive, see:

- Database Backup (ON-Bar) [Extern]
- Archive (ON-Archive and ontape) [Extern]

Prerequisites

- You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].
- You know how to use the DBA Planning Calendar. For more information, see:
  - DBA Planning Calendar [Seite 1195] for an overview
  - DBA Planning Calendar (Informix) [Seite 1197] for Informix-specific aspects

  For more information about scheduling an action (for example, a database backup) in the DBA Planning Calendar, see Scheduling Actions in the DBA Planning Calendar [Seite 837].
- Your storage devices are ready and have media loaded with enough available space. For example, if using tape, your tape device is ready to receive data and you have loaded a tape with enough available space.

Procedure

1. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar*.
2. Choose the day when you want the database backup or archive to take place.
3. Choose *Create action*.
4. Select an *Action* from the list as follows:

ON-Archive

Select	To perform
Database full archive	A full archive of all dbspaces of the database
Incremental archive level 1	An incremental archive of all dbspaces changed since the last full archive
Incremental archive level 2	An incremental archive of all dbspaces changed since the last incremental archive at level-1

For more information about archiving with ON-Archive **outside** CCMS, see Creation of an Archive (ON-Archive) [Extern].

ON-Bar

Select	To perform
Database backup (dbspaces)	A full backup of all or selected dbspaces of the database
Incremental database backup (dbspaces)	An incremental backup of dbspaces changed since the last database backup
Whole system backup (serial)	A full backup of all dbspaces and the logical log, executed serially
Incremental whole system backup (serial)	An incremental backup of all dbspaces changed since the last database backup and a logical-log backup, executed serially

For more information about backing up the database with ON-Bar **outside** CCMS, see Creation of a Database Backup (ON-Bar) [Extern].

5. Enter data as required in the fields *Start Time*, *Period (weeks)*, and *Calendar*. For more information, see Scheduling Actions in the DBA Planning Calendar [Seite 1201].

6. Choose *Continue*.

   If your chosen action requires more parameters, the system prompts for them.

   For example, with ON-Archive, you have to select the *Vset name* (that is, volume set name) for the archive. SAP recommends that you use volume set `DBTAP` for archives. For more information, see Volume Sets and Volumes for ON-Archive [Extern].

7. Enter data as required and choose *Continue*.

   If either of the following conditions applies, a full (level-0) database backup or archive is executed, even though you scheduled an incremental (level-1 or level-2) backup or archive:

   - There is no successfully executed level-0 database backup or archive.
   - You have altered the dbspace structure since the last level-0 database backup or archive. In other words, you have added or deleted non-temporary dbspaces. A prompt warns you of this when you start the DBA Planning Calendar.

   If you enter more than one volume set for an archive with ON-Archive, the archive runs in parallel. CCMS automatically determines the allocation of dbspaces to volume sets. For more information about planning parallel archives, see Parallel Archive Approach (ON-Archive) [Extern].

Result

The database backup or archive is now scheduled. It will be created at the scheduled date and time. For more information about looking at the results of the database backup or archive, see Using the Archive and Backup Monitor in CCMS (Informix) [Seite 837].

Backing Up the Logical Log in the DBA Planning Calendar (Informix)

Use

You can use the DBA Planning Calendar [Seite 1195] in the Computing Center Management System (CCMS)to schedule logical-log backups for the Informix database. You can follow this procedure if you use the Informix data recovery tools ON-Bar or ON-Archive.

You can create a logical-log backup for the Informix database with the DBA Planning Calendar in the Computing Center Management System (CCMS) by using the following methods:

- Normal scheduled logical-log backup

  With this you can back up the logical log including the currently used log file at the scheduled time. This is the method described in this section.

- Automatic logical-log backup

  With ON-Archive you can use this method to trigger logical-log backup. This works by detecting the fill level of the logical log. When a pre-defined level is reached, the backup job triggered to run. Therefore, you must make sure that the correct tape volume is always mounted. This method offers you an extra level of security to avoid the logical log filling up.

  For more information, see Backing Up the Logical Log (Automatic) in the DBA Planning Calendar (Informix) [Seite 1211].

SAP recommends that you use both methods for extra security. Always keep a dedicated tape drive free when backing up logs automatically.

> If the logical logs are not backed up before they completely fill, you need to perform an emergency backup. This is complex, time-consuming and leads to unplanned downtime for your system. You can avoid this by devising a sensible backup schedule with the Calendar. Always make sure that the correct empty tape is loaded in the appropriate tape drive. See Preventing Emergency Logical-Log Backup [Extern].
>
> If you need to execute a logical-log backup immediately, see Scheduling Actions in the DBA Planning Calendar [Seite 1201].

For more information about logical-log backup, see Logical-Log Backup [Extern].

Prerequisites

- You are ready to use CCMS. Refer to Getting Started in CCMS with DBA Tasks for Informix [Seite 1194].
- You know how to use the DBA Planning Calendar. For more information, see:
  - DBA Planning Calendar [Seite 1195] for an overview
  - DBA Planning Calendar (Informix) [Seite 1197] for Informix-specific aspects

For more information about scheduling an action (for example, a logical-log backup) in the DBA Planning Calendar, see Scheduling Actions in the DBA Planning Calendar [Seite 837].

- Your storage devices are ready and have media loaded with enough available space. For example, if using tape, your tape device is ready to receive data and you have loaded a tape with enough available space.

Procedure

8. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar*.
9. Choose the day when you want the logical-log backup to take place.
10. Choose *Create action*.
11. Select *Logical-Log Backup* (ON-Bar) or *Logfile Backup* (ON-Archive).

    For more information about creating a logical-log backup **outside** CCMS, see:

    - Creation of a Logical-Log Backup (ON-Bar) [Extern]
    - Creation of a Logical-Log Backup (ON-Archive) [Extern]
12. Enter data as required in the fields *Start Time*, *Period (weeks)*, and *Calendar*. For more information, see Scheduling Actions in the DBA Planning Calendar [Seite 1201].
13. Choose *Continue*.

    If your chosen action requires more parameters, the system prompts for them.

    For example, with ON-Archive, you have to select the *Vset name* (that is, volume set name) for the logical-log backup. SAP recommends that you use volume set `LOGTAP` for logical-log backups. For more information, see Volume Sets and Volumes for ON-Archive [Extern].
14. Enter data as required and choose *Continue*.

Result

The logical-log backup is now scheduled. It will be created at the scheduled date and time. For more information about looking at the results of the logical-log backup, see Using the Archive and Backup Monitor in CCMS (Informix) [Seite 837].

Backing Up the Logical Log (Automatic) in the DBA Planning Calendar (Informix)

Use

To avoid the logical-log files of an Informix database filling up, you can activate a triggered automatic logical-log backup in the DBA Planning Calendar [Seite 1195], which is part of the Computing Center Management System (CCMS). You can **only** follow this procedure if you use the Informix data recovery tool ON-Archive.

You can create a logical-log backup for the Informix database with the DBA Planning Calendar in the Computing Center Management System (CCMS) by using the following methods:

- Normal scheduled logical-log backup

  With this you can back up the logical log including the currently used log file at the scheduled time. For more information, see Backing Up the Logical Log in the DBA Planning Calendar (Informix) [Seite 1209].

- Automatic logical-log backup

  With ON-Archive you can use this method to trigger logical-log backup. This works by detecting the fill level of the logical log. When a pre-defined level is reached, the backup job triggered to run. Therefore, you must make sure that the correct tape volume is always mounted. This method offers you an extra level of security to avoid the logical log filling up.

  This is the method described in this section.

SAP recommends that you use both methods for extra security. Always keep a dedicated tape drive free when backing up logs automatically.

If the logical logs are not backed up before they completely fill, you need to perform an emergency backup. This is complex, time-consuming and leads to unplanned downtime for your system. You can avoid this by devising a sensible backup schedule with the Calendar. Always make sure that the correct empty tape is loaded in the appropriate tape drive. See Preventing Emergency Logical-Log Backup [Extern].

If you need to execute a logical-log backup immediately, see Scheduling Actions in the DBA Planning Calendar [Seite 1201].

For more information about logical-log backup, see Logical-Log Backup [Extern].

Prerequisites

- You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].
- You know how to use the DBA Planning Calendar. For more information, see:
  - DBA Planning Calendar [Seite 1195] for an overview
  - DBA Planning Calendar (Informix) [Seite 1197] for Informix-specific aspects

For more information about scheduling an action (for example, a logical-log backup) in the DBA Planning Calendar, see Scheduling Actions in the DBA Planning Calendar [Seite 837].

- Your storage devices are ready and have media loaded with enough available space. For example, if using tape, your tape device is ready to receive data and you have loaded a tape with enough available space.

Procedure

15. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar*.
16. Choose *Calendar* → *Automatic Logsave*.
17. Enter the percentage fill level to trigger a backup and choose *Continue*.

    The percentage fill level is how full the logical-log files must be in order for a backup to be performed. A typical value might be 50%. To turn off the automatic logsave, enter 0%.
18. Choose the *Vset name* (that is, the volume set name) to be used for the triggered backup.

    For more information, see Volume Sets and Volumes for ON-Archive [Extern].
19. Choose *Continue*.

Result

The automatic logical-log backup is now scheduled. A logical-log backup will be created when the logical log reaches the fill level you specified. For more information about looking at the results of the logical-log backup, see Using the Archive and Backup Monitor in CCMS (Informix) [Seite 837].

Checking Statistics in the DBA Planning Calendar (Informix)

Use

You can use the DBA Planning Calendar [Seite 1195] in the Computing Center Management System (CCMS) to schedule a check to see whether the statistics on the Informix database need updating. You can only schedule check statistics for **all** tables in the database.

The information generated by this function is evaluated for the "Optimizer Statistics" alert. Refer to Monitoring Optimizer Statistics (Informix) [Seite 458]. For more information, see Note 64210 in SAPNet – R/3 Frontend.

You can also schedule an **update** statistics in the DBA Planning Calendar. Refer to Updating Statistics in the DBA Planning Calendar (Informix) [Seite 1214].

For more information about update statistics – which you can also perform with SAPDBA (that is, outside CCMS) – see Update Statistics with SAPDBA [Extern].

Prerequisites

- You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].
- You know how to use the DBA Planning Calendar. For more information, see:
  - DBA Planning Calendar [Seite 1195] for an overview
  - DBA Planning Calendar (Informix) [Seite 1197] for Informix-specific aspects

    For more information about scheduling an action (for example, check statistics) in the DBA Planning Calendar, see Scheduling Actions in the DBA Planning Calendar [Seite 1201].

Procedure

20. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar*.
21. Choose the day when you want the statistics to be checked.
22. Choose *Create action*.
23. Select *Check: upd. stat. needed (for all tabs)*.
24. Enter data as required in the fields *Start Time*, *Period (weeks)*, and *Calendar*. For more information, see Scheduling Actions in the DBA Planning Calendar [Seite 1201].
25. Choose *Continue*.

Result

The check statistics is now scheduled. It will be executed at the scheduled date and time. For more information about looking at the results of the check statistics, see Checking the Results of Actions in the DBA Planning Calendar [Seite 1204].

Updating Statistics in the DBA Planning Calendar (Informix)

Use

You can use the DBA Planning Calendar [Seite 1195] in the Computing Center Management System (CCMS) to schedule update statistics for the Informix database. You can update statistics for all tables in the database or only for one table.

You can also schedule a **check** statistics in the DBA Planning Calendar. Refer to Checking Statistics in the DBA Planning Calendar (Informix) [Seite 1213].

For more information about update statistics – which you can also perform with SAPDBA (that is, outside CCMS) – see Update Statistics with SAPDBA [Seite 837].

Prerequisites

- You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].
- You know how to use the DBA Planning Calendar. For more information, see:
  - DBA Planning Calendar [Seite 1195] for an overview
  - DBA Planning Calendar (Informix) [Seite 1197] for Informix-specific aspects

    For more information about scheduling an action (for example, check statistics) in the DBA Planning Calendar, see Scheduling Actions in the DBA Planning Calendar [Seite 837].

Procedure

26. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar*.
27. Choose the day when you want the statistics to be updated.
28. Choose *Create action*.
29. Select one of the following:
    - *Update Optimizer Statistics (all tables)*
    - *Update Optimizer Statistics (one table)*
5. Enter data as required in the fields *Start Time*, *Period (weeks)*, and *Calendar*. For more information, see Scheduling Actions in the DBA Planning Calendar [Seite 1201].
6. Choose *Continue*.
7. If you are scheduling update statistics for one table, enter the *Table* name.
8. Enter the parameters to specify the update statistics:

Parameter	Meaning

Threshold for Update Statistics	The statistics for a table are only updated when the optimizer value for the number of rows in the table deviates from the current (that is, correct) number of rows in the table by more than a certain value, that is, the "threshold". You can either use the default value of 10% or enter a value of your own.
Execution strategy	To optimize the run time for update statistics, you can choose an execution strategy that performs update statistics in parallel. You can determine how many parallel processes are started for update statistics. The default for this parameter is the number of CPU VPs configured in your system. The number of CPU VPs is specified in the `ONCONFIG` parameter `NUMCPUVPS`. If you choose a value less than 2, update statistics is performed sequentially. For more information, see NUMCPUVPS (Informix) [Seite 487]. With update statistics for all tables, processing is performed in parallel at table level. If the statistics for an individual table need to be updated, processing is performed in parallel at column level.
Application Monitor Statistics	The default for this parameter is “no” (that is, the box is not selected). If you select this to activate the parameter, additional space statistics are calculated for each table. These can be displayed by the application monitor. Since this calculation is very time-consuming, SAP recommends that you only activate this parameter if you work with the application monitor.
Maximum Runtime	The default for this parameter is “no limit”. If you want to make sure that the update statistics does not last too long, you can specify a maximum runtime in minutes. When this limit is reached, the update statistics ends after the current table.
Log file	Using this parameter, you can enter the directory and file name of the log file for update statistics. The default is `$INFORMIXDIR/sapreorg/updstat_<SID>.log`. If the directory you enter does not exist, the default directory is used. If the default directory does not exist, the log is written to `/tmp/updstat_<SID>.log`.
Detailed	The system writes additional information for each table to the log. The default is to write only overview information to the log.

If you perform update statistics with an R/3 release prior to 3.1G, it runs as follows:

- Default threshold value (10%)
- No parallel processing
- Calculation of application monitor data is activated
- No runtime limit
- Log file defaults to `$INFORMIXDIR/sapreorg/updstat_<SID>.log`, or (if the required directory does not exist) to `/tmp/updstat_<SID>.log`.

If you wish to alter this, you must delete the planned actions and re-schedule them with R/3 Release 3.1G or later. However, note that parallel processing and the specification of the log file are only available for jobs scheduled with R/3 Release 4.0B or later.

9. Choose *Continue*.

Result

The update statistics is now scheduled. It will be executed at the scheduled date and time. For more information about looking at the results of the update statistics, see Checking the Results of Actions in the DBA Planning Calendar [Seite 1204].

Checking the DB System in the DBA Planning Calendar (Informix)

Use

You can use the DBA Planning Calendar [Seite 1195] in the Computing Center Management System (CCMS) to schedule database system checks against your Informix database. These check the configuration and performance of the database.

For more information about configuration and performance checks, see DB System Checks in CCMS (Informix) [Seite 1226].

Prerequisites

- You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].
- You know how to use the DBA Planning Calendar. For more information, see:
  - DBA Planning Calendar [Seite 1195] for an overview
  - DBA Planning Calendar (Informix) [Seite 1197] for Informix-specific aspects

  For more information about scheduling an action (for example, check DB system) in the DBA Planning Calendar, see Scheduling Actions in the DBA Planning Calendar [Seite 837].

Checks that you schedule from the DBA Planning Calendar are executed as follows:

- Using the settings current at execution time

  For more information about how to modify the settings for database system checks before running them in the DBA Planning Calendar, see Configuring DB System Checks in CCMS (Informix) [Seite 1227] [Seite 1227].

- All checks are executed

  That is, you cannot specify that only certain checks are executed for a given run.

Procedure

30. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar*.
31. Choose the day when you want the check to be started.
32. Choose *Create action*.
33. Select *Database Configuration Check*.
34. Enter data as required in the fields *Start Time*, *Period (weeks)*, and *Calendar*. For more information, see Scheduling Actions in the DBA Planning Calendar [Seite 1201].
35. Choose *Continue*.

Result

The check is now scheduled. It will be executed at the scheduled date and time, using the settings current at execution time. For more information about looking at the results, see Checking the Results of Actions in the DBA Planning Calendar [Seite 1204].

For more specific information about how to see the results of the checks, see Viewing DB System Checks in CCMS: Informix [Seite 1230].

Troubleshooting in the DBA Planning Calendar (Informix)

Use

This section tells you what to do in the event of problems with the DBA Planning Calendar in the Computing Center Management System (CCMS) for the Informix database.

For information about troubleshooting with the alert monitor, see Monitoring the Database with the Alert Monitor (Informix) [Seite 1238].

Prerequisites

Any scheduled action can occasionally fail, so you must at least check the more critical actions such as archive and backup daily. Refer to Checking the Results of Actions in the DBA Planning Calendar [Seite 1204].

Procedure

1. Check that the background job was executed correctly.

   Consult the job log to check this. If no job log exists, the background job was probably not started. You can get more details using the job overview with transaction SM37 (note that the names of all jobs scheduled in the Calendar start with *DBA*). The job log also tells you whether an external program was started.

2. Consult the action log (if available) if you are sure that the background job ran successfully.
3. Display the operating system log (if available).
4. Check for common errors such as:
   - There is no tape in the tape drive
   - The tape in the tape drive is write protected.
   - The tape in the tape drive has not been initialized.
   - The tape drive contains the wrong tape.
   - The tape is full.
   - There has been an error with the tape drive.

Result

When you have corrected the error, reschedule the action to execute immediately. Refer to Scheduling Actions in the DBA Planning Calendar [Seite 1201].

See also:

DBA Planning Calendar (Informix) [Seite 1197]

Update Statistics in CCMS (Informix)

Purpose

You can update statistics on the Informix database using the Computing Center Management System (CCMS). The cost-based optimizer (CBO) for the Informix database needs current, up-to-date statistics about the number of data rows, size of indexes and the distribution of data in individual fields of a table. By running update statistics, you make sure that the CBO has up-to-date information so it can work out the best access path to get data for a query.

If the statistics held by the CBO are out of date or not present at all, the CBO might generate inappropriate access paths, for example, use of the wrong index or inappropriate use of a full table sequential scan when a suitable index exists. The result is poor database performance.

You can also run update statistics for your Informix database using SAPDBA. Refer to Update Statistics with SAPDBA [Extern].

Prerequisites

Before running update statistics in CCMS, check that you have met the R/3 System prerequisites in Getting Started in CCMS with Informix DBA [Seite 1194].

Process Flow

You have the following possibilities for running update statistics in CCMS:

1. You can use the DBA Planning Calendar in CCMS to schedule regular execution of check statistics and update statistics, as follows:

   a) You can schedule check statistics [Seite 1213] in the DBA Planning Calendar to regularly examine how far the actual statistics deviate from the stored statistics for a table or tables.

   b) You can schedule update statistics [Seite 1214] in the DBA Planning Calendar to ensure that the database statistics are regularly updated.

   If you are new to the DBA Planning Calendar, see DBA Planning Calendar (Informix) [Seite 1197].

2. You can also run one-off checks on individual tables to see if the table's statistics are out-of-date, and then run an update statistics for the table if required. This is useful, for example, if the data in a table has been significantly updated, but the next scheduled run of update statistics is not for some time. Refer to Checking and Updating Statistics for a Single Table in CCMS (Informix) [Seite 1221].

3. You can configure update statistics [Seite 1223] by amending the parameters in the control table `DBSTATC`. This control table contains a list of the database tables for which the default values for update statistics are not suitable. If you change this table, **all** runs of update statistics – in SAPDBA, CCMS, or the DBA Planning Calendar – are affected. Configuring update statistics makes sense with large tables, for which the default parameters might not be appropriate.

Result

The statistics used by the Informix CBO are kept up-to-date, resulting in optimal execution of queries and better performance.

Checking and Updating Statistics for a Single Table in CCMS (Informix)

Use

You can use this procedure to do the following for a **single** table in your Informix database:

- Check whether the table's statistics deviate significantly from the current value.
- Update the statistics for the table, if required.

Prerequisites

You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].

Use this procedure when you determine there is a need to update the statistics for a table. For example, use this procedure in the following situations:

- You or SAP's EarlyWatch determine that a table needs updated statistics.
- You change the data of a table significantly (for example, with batch input).

Procedure

1. Choose *Tools* → *CCMS* → *DB Administration* → *Cost based optimizer* → *Create statistics*.

   The table you enter in the next step should meet one of the qualifications listed above in "Prerequisites".
2. Enter the name of the table in *TableName* and choose *Continue*.

   You see the details of the last statistics update, that is, *Date*, *Time*, and *Accuracy*. The results of the last statistics update are displayed in *Statistic* under *Number of rows*.
3. To check whether the table's statistics need refreshing, choose *Count total number of rows*.

   You see the current number of rows in the table in the field *Total*. The deviation between the current number of rows in the table and the value held in the table's statistics is shown as a percentage in the *Deviation* field.
4. If the table's statistics need to be updated, select the *Accuracy* you require.
5. Choose *Refresh* to perform update statistics.

Result

The Informix optimizer now has up-to-date statistics for the table, resulting in optimal execution of queries against the table.

See also:

Update Statistics in CCMS (Informix) [Seite 1220]

Configuring Update Statistics in CCMS (Informix) [Seite 1223]

Checking and Updating Statistics for a Single Table in CCMS (Informix)

Update Statistics with SAPDBA [Extern]

Configuring Update Statistics in CCMS (Informix)

Use

This procedure enables you to change the table DBSTATC, used to control the update of cost-based optimizer (CBO) statistics for selected R/3 tables in the Informix database. SAP delivers the control table DBSTATC to you with several hundred pre-defined entries for R/3 Basis and Application tables that have special requirements for statistics updates. For all other tables (that is, for those not appearing in the control table), update statistics is performed with the default settings. If you make an entry in the control table DBSTATC, you are changing the default settings used for analyzing an R/3 table and updating its statistics.

If you change the control table DBSTATC, **all** runs of update statistics – in SAPDBA or CCMS (including in the DBA Planning Calendar) – use the changed settings. Do **not** change SAP's pre-defined entries without good reason.

Prerequisites

You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].

Procedure

1. Choose *Tools* → *CCMS* → *DB Administration* → *Cost based optimizer* → *Configuration*.

   The system displays the control table DBSTATC, showing configuration information for tables that have special requirements for update statistics. The most important fields are as follows:

Important Fields in DBSTATC Table

Field	Value	Comment
DB object	*<TABLE NAME>*	The name of the table with special requirements for update statistics (must be entered in uppercase)
DB object type	*01*	Indicates “table” – this is the default setting
Use	*O* *A*	R/3 Basis table – this is the default setting Refers to a table from application monitor (see transaction ST07) – do not normally use (if you use this setting, extra statistics are collected for the application monitor when update statistics is executed using SAPDBA from the command line, which greatly increases run times.)

Active	*A*	Update statistics is performed for table, if necessary – this is the default setting
	N	Excludes table from update statistics – do not use unless SAP advises you to
	U	Forces update statistics on table for every run of update statistics – do not use unless SAP advises you to, as it increases run times
Accuracy	*1*	Forces update statistics HIGH for all index columns
	2	Update statistics HIGH for leading index columns, MEDIUM for other index columns – this is the default setting
TODO flag	*X*	Forces update statistics of the table the next time update statistics is run – do not normally use, as this might cause unnecessary alerts to be produced
History flag	*X*	Causes historical data to be collected
TODO chg. date		The date on which the entry in DBSTATC was last changed – do not use

Do **not** enter or change any fields other than the ones shown above. Other fields are either not relevant to Informix databases, or are maintained automatically by the system.

2. To see full details of an entry in the control table DBSTATC, choose the entry you want to see and then choose *Detail*.

   The system displays all the fields for the entry you choose.

3. To add a new entry to the control table DBSTATC (that is, to configure the settings used for checking and updating statistics on an R/3 table), do the following:

   a) If necessary, choose *Change -> Display* to switch to change mode.

   b) Choose *New entries*.

      The system displays a dialog box for you to specify the new entry.

   c) Enter values as shown in the table in step 1. Use the default settings except for fields where you have good reason to use some other setting.

   d) Save the entry.

4. To change an existing entry in the control table DBSTATC (that is, to re-configure the settings used for checking and updating statistics on an R/3 table already in the control table), do the following:

   a) If necessary, choose *Change* → *Display* to switch to change mode.

   b) Choose the entry you want to change.

   c) Choose *Detail*.

      The system displays a dialog box for you to change the existing entry.

d) Enter values as shown in the table in step 1. Use the default settings except for fields where you have good reason to use some other setting.

e) Save the entry.

5. To delete an entry from the control table `DBSTATC` (that is, to use the default settings are used for checking and updating statistics on an R/3 table), do the following:

   Do **not** delete the entries made by SAP in the control table `DBSTATC` if you use the application monitor, because the statistics required by the application monitor can then no longer be collected.

   a) If necessary, choose *Change* → *Display* to switch to change mode.

   b) Choose the entry you want to delete.

   c) Choose *Delete*.

   d) Save the deletion.

   When you have deleted an entry from the control table `DBSTATC`, update statistics uses the default settings for the deleted R/3 table in future.

Result

The way the Informix optimizer updates statistics for tables with entries in the control table `DBSTATC` is changed. This takes effect the next time update statistics is run. Update statistics run from SAPDBA or CCMS (including from the DBA Planning Calendar) is affected by such changes.

See also:

Update Statistics in CCMS (Informix) [Seite 1220]

Checking and Updating Statistics for a Single Table in CCMS (Informix) [Seite 1221]

Update Statistics with SAPDBA [Extern]

DB System Checks in CCMS (Informix)

Purpose

You can use the Computing Center Management System (CCMS) to run a series of performance and configuration checks on your Informix database system, including the following:

- Parameters in the `ONCONFIG` file
- Chunk and disk layout
- Lock mode of tables
- Number of extents for a table
- Access rights for programs

The database system checks (shortened to "DB system checks") make **no** amendments to your database. The checks are easy to perform, are quickly finished, and have a minor effect on database performance. If problems arise, there is a detailed event log to help you find a solution.

You can also run the checks in SAPDBA. Refer to DB System Checks with SAPDBA [Extern].

Process Flow

SAP recommends you run the checks as follows:

Checks that are very important to database operation are run automatically without you having to do anything, and these checks raise alerts if required in the CCMS alert monitor. Some checks are run hourly, some daily. You can deactivate these checks if you want.

For more information, see Configuring DB System Checks in CCMS: Informix [Seite 1227] and Monitoring the Database with the Alert Monitor: Informix [Seite 1238].

1. If required, you configure the checks to better suit your system. Refer to Configuring DB System Checks in CCMS (Informix) [Seite 1227].
2. You schedule the checks to run regularly. Refer to Checking the DB System in the DBA Planning Calendar (Informix) [Seite 1217].
3. If you have a problem with your database or you have changed the configuration, you run the checks to help find what is wrong or to make sure the new configuration is acceptable. Refer to Running DB System Checks in CCMS (Informix) [Seite 1229].
4. You view the results. Refer to Viewing DB System Checks in CCMS (Informix) [Seite 1230].

Result

Since many database problems can be traced to incorrect settings, the DB system checks enable you to easily identify common problems. The result is improved performance of your Informix database and high availability for your R/3 System.

Configuring DB System Checks in CCMS (Informix)

Use

You can use this procedure to configure database system checks (shortened to "DB system checks") for your Informix database. This procedure is started from the Computing Center Management System (CCMS) in the R/3 System.

Prerequisites

For more information about what you need to do before using the Computer Center Management System (CCMS) for DBA tasks, see Getting Started in CCMS with Informix DBA [Seite 1194].

Procedure

1. Choose *Tools* → *CCMS* → *DB Administration* → *Check* → *Configuration*.

   The system displays a dialog showing the different kinds of check that you can configure.

2. Select the *Filter* you want.

   The filter determines what type of check the system displays for you to configure. If you want to filter for a particular class of check, select *origin class* and enter a valid class in the field *checks*. To filter for the checks that are automatically scheduled (that is, the checks that can lead to alerts being raised in the alert monitor), select *Alert schedule class* and enter *Daily* or *Hourly*.

3. Choose *Show checks* to display the checks in more detail.

   The system displays a summary list showing all single checks for the filter you have selected. The filter is displayed at the bottom of the screen.

   If there is an entry in the column *Schedule* (for example, *daily* or *hourly*), then this parameter is used to set an alert if necessary. For more information, see Monitoring the Database with the Alert Monitor: Informix [Seite 1238] and The Alert Browser: Working with Alerts [Seite 835].

4. To modify a single check, select a row and then choose *Modify row*.

   You cannot modify all single checks. For example, the check *Chunksize* cannot be modified.

   There are sometimes multiple rows for a single parameter. This is due to different system combinations. Make sure you change the correct row for your system combination – if in doubt, change all rows for the parameter.

   The system displays the parameters of the single check you have selected.

5. To find more information about the single check, choose *Information*.

   A help window is displayed showing detailed information about the single check that you are modifying.

6. Enter your requirements for the single check in the box *Customizable*, according to the table below:

Configuring DB System Checks in CCMS (Informix)

Customizable Parameters for a Single Check

Type of Check	How you can Customize the Check
Dynamic	Activate or de-activate check by selecting *Activated*? Alter *severity* of check, affecting how the check appears on the report Enter a comment to identify your change in *Modif. comment*
Static	As for dynamic checks plus the following: Enter an operator in the field *Checkop.* Enter a value in the field *Checkvalue*. Enter a unit in the field *Byteunit* to determine how *Checkvalue* is to be interpreted, if bytes are used for the check.

Extra information about the check is displayed in the boxes *Class*, *Last modification* and *Platform combination*.

7. If you have modified parameters for the single check and want to reset them to the values before you started, choose *Screen entry*. If you want to reset the parameters to the default values set by SAP, choose *SAP original*.

8. When the parameters for the single check are as required, save the check.

Result

You have configured the DB system check to meet your requirements.

See also:

Running DB System Checks in CCMS (Informix) [Seite 1229]

Viewing DB System Checks in CCMS (Informix) [Seite 1230]

Running DB System Checks in CCMS (Informix)

Use

You can use this procedure to run database system checks (shortened to "DB system checks") for your Informix database. This procedure is started from the Computing Center Management System (CCMS) in the R/3 System.

You can also run the DB system checks in the DBA Planning Calendar [Seite 1217] or SAPDBA [Extern].

Prerequisites

You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].

Procedure

1. Choose *Tools → CCMS → DB Administration → Check → Monitor*.
2. To run a check using the current configuration, choose *Run check*.

   The system prompts you to define the type of check you want to run.

   If you want to change the configuration, see Configuring DB System Checks in CCMS (Informix) [Seite 1227].
3. Select the type of check you want to run. You can run a particular class of check by selecting *Origin class* and entering a valid class. You can run a single check by selecting *Parameter* and entering the name of the check. You can re-run the automatically executed checks by selecting *Alert schedule class* and then entering *Hourly* or *Daily*.
4. If you want very detailed information to appear in the results log, select *store and show all single checks (verbose)*.
5. To execute the check, choose *run check*.

Result

The system executes the check, normally taking only a short time.

SAP recommends that you **always** look at the results of check runs as soon as possible, otherwise you might fail to react in good time to a warning or error. At worst, this might lead to data loss.

For more information about checking the results, see Viewing DB System Checks in CCMS (Informix) [Seite 1230].

See also:

Configuring DB System Checks in CCMS (Informix) [Seite 1227]

Viewing DB System Checks in CCMS (Informix)

Use

You can use this procedure to view the results of database system checks (shortened to "DB system checks") for your Informix database, or to delete results from previous checks. This procedure is started from the Computing Center Management System (CCMS) in the R/3 System.

We recommend that you **always** check the results of check runs as soon as possible, otherwise you might fail to react in good time to a warning or error.

Prerequisites

You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].

You must run a check before you can view the results. Refer to Running DB System Checks in CCMS (Informix) [Seite 1229].

Procedure

1. Choose *Tools* → *CCMS* → *DB administration* → *Check* → *Monitor*.

   The system displays a list showing the results of previous check runs. For each check run, the highest detected severity is displayed, using a color-coded scheme (that is, red for errors, yellow for warnings, green for OK):

DB System Check Results

Severity	Color	What to do
Abort	Red	You must **immediately** correct the severe error, otherwise data might be lost
Error	Red	You should correct the error **as soon as possible**, otherwise a severe error might occur leading to data loss
Warning	Yellow	You should correct the setting, but do not need to do so immediately
Not checked	Yellow	As the system could not perform the check, you should find out why not (this is usually given in the recommendation)
Nothing to complain	Green	As the check is OK, you need not do anything.

2. To display the results for a check run, select the row.

   The system displays entries for the following:

   – A *Summary* for the check run

   – Checks grouped by severity

3. Look at the *highest detected severity* in the summary.

   If this is *nothing to complain*, you need take no action, as your system has passed all checks. Otherwise, continue with this procedure.

4. Drill down to display the results of each individual parameter checked.
5. If the check is color-coded red or yellow, perform the necessary action, as identified in the table above.

   For each such check, the system displays the expected value and the current value, together with a recommendation, as shown in the following example:

   When looking at the detailed results for a check run, you might see something like the following for a single check:

   EXTENTS for table vrsx (class DBGENERAL) non-performance check

   expected... <= 200   (SAP original)

   found 228

   recommendation. section D-021 in note ‘64001’ ?

   please reorganize table immediately

   This means that the value in the variable *EXTENTS* found for the table *vrsx* exceeds the recommended maximum value of 200. Consult Note 64001 in SAPNet – R/3 Frontend, or choose *?* to display background information about this single check.

   You can also see the following information:

   - The *class* of the check is *DBGENERAL*, that is, general database checks.
   - The check is a *non-performance check*.
   - The expected value of the check is *SAP original*. This means that the expected value had not been amended before the check was run.
6. To find more information about the recommendation for a check, choose the question mark next to it.

   A help window is displayed showing detailed information about the single check to help you correct the problem.
7. To delete the results from a previous check run, select the row and choose *Delete Protocols*. You can delete the results of several check runs at the same time. If you choose *Delete Protocols* without having selected a row, you can delete all the results.
8. If there are more difficult problems, choose *Generate Debug Protocol* to look at a detailed event log.

Result

If you follow the recommendations given for checks color-coded red or yellow, you can make sure that your system remains as well tuned as possible. This leads to reduced downtime and better system performance.

See also:

Configuring DB System Checks in CCMS (Informix) [Seite 1227]

Archive and Backup Monitor in CCMS (Informix)

Use

The archive and backup monitor for the Informix database is part of the Computing Center Management System (CCMS) in the R/3 System. You can use it to view the results of the following:

- Database backups (ON-Bar) or archives (ON-Archive)
- Logical-log backups, including the status of the logical log

The advantage of using the archive and backup monitor is that all the information is in a single place. This lets you review past archive and backup activity without looking up the results individually.

Integration

The archive and backup monitor is one of a number of tools available in CCMS for performing database administration (DBA) tasks with the Informix database. Refer to SAP/Informix DBA in CCMS [Seite 1192].

Prerequisites

You have performed a database backup, an archive, or a logical-log backup. Otherwise, there is little relevant information to be displayed in the backup monitor (other than the status of the logical log).

You use ON-Bar [Extern] or ON-Archive [Extern] as your data recovery tool for the Informix database.

Features

For Informix, the following features are implemented:

- Details of the last successful database backup or archive
- Overview of all database backups or archives
- Status of the logical log, including percent used
- Details of the logical-log files that have been backed up recently
- Overview of all logical-log backups
- Volume report showing details of the volumes used for archives and backups (only available with ON-Archive)
- Recovery report from SAPDBA showing details required to restore the database (only available with ON-Archive)

Activities

- Schedule a database backup, an archive, or a logical-log backup in the DBA Planning calendar. Refer to DBA Planning Calendar (Informix) [Seite 1197].
- Look in the archive and backup monitor for the results. Refer to Using the Archive and Backup Monitor in CCMS (Informix) [Seite 1233].

Using the Archive and Backup Monitor in CCMS (Informix)

Use

You can use the archive and backup monitor [Seite 1232] in the Computing Center Management System (CCMS) to check in detail the results of an archive, a database backup, or a logical-log backup started in the DBA Planning Calendar [Seite 1195]. For more information on scheduling these actions, see Scheduling Actions in the DBA Planning Calendar [Seite 1201].

Prerequisites

You are ready to use CCMS. Refer to Getting Started in CCMS with Informix DBA [Seite 1194].

You have performed an archive, a database backup, or a logical-log backup in the DBA Planning Calendar.

You use ON-Bar [Extern] or ON-Archive [Extern] as your data recovery tool for the Informix database.

Procedure

1. Choose *Tools* → *CCMS* → *DB Administration* → *Backup Logs*.

   The system displays the archive and backup monitor, including a summary of the logical-log status.

2. If you are using ON-Bar, choose between *Dbspace backups* and *Whole system backups*, and between *All backups* and *CCMS backups only*.

   The system changes the display accordingly.

3. Choose the appropriate function to display the information you want to see, as shown in the following table:

   The functions available – reflected in the display – vary according to whether you are using ON-Bar or ON-Archive. With ON-Bar, the display also varies according to whether you have chosen whole-system backups or database backups, and CCMS backups or all database backups (see previous step).

Function	Description
Last database backup (all dbspaces) (ON-Bar) *Last whole system backup* (ON-Bar) *Last successful database archive* (ON-Archive)	Displays the date and time of the last database backup or archive. Choose to see further details of the database backup or archive (such as the dbspaces and so on).

Overview of database backups (ON-Bar) *Overview of whole system backups* (ON-Bar) *Overview of all database archives* (ON-Archive)	Choose to see an overview of all database backups and archives performed.
Status of the logical log	Displays percentage used of the logical log. Choose to see further details about the current status of the logical log (that is, total size in Kb, amount used in Kb, amount available in Kb, percent used).
Most recent logs backed up	Displays the number of the latest logical-log file *Not yet backed up*. Choose to see a recent history of logical-log backups.
Overview of all logical log backups	Choose to see a comprehensive overview of all logical-log backups.
CCMS backups only (ON-Bar) All backups (ON-Bar)	Choose to see all backups or only those performed in CCMS.
Volume Report (ON-Archive)	Choose to see full details of each archived dbspace or backed-up logical log. The information displayed includes the volume set used, label and volume number of the tape and its creation date.
Recovery Report (ON-Archive)	Displays the recovery reports created by SAPDBA, if any exist. You can also generate a recovery report immediately. Note that a recovery report created in CCMS is not automatically stored in the same file system as it is when created with SAPDBA. For further details on recovery reports in SAPDBA, see Recovery Report with SAPDBA [Extern].

Result

By checking the results of archives and backups you can make sure that they have completed successfully. Remember that unsuccessful archives and backups might mean that production data is lost in the event of failure, because you are not able to perform a complete database restore.

See also:

Archive and Backup Monitor in CCMS (Informix) [Seite 1232]

Using the DB Operations Monitor

Use

Use the DBA operations monitor for online monitoring of database operations within both internal and external database tools. You can also monitor runtime and remaining time of operations that are running. The DBA operations monitor can be used regardless of which database you are using. It provides historical as well as current (online) information about the following database operations:

- Backup/Recovery (for example, to back up or recover the database)
- Performance (for example, checking, creating, updating and deleting database statistics)
- The memory structure (for example, space information for database objects, reorganization of database objects, or extending and deleting database objects).
- Database checks (for example, checking the database for critical situations)
- Configuration (for example, configuring database parameters)

Prerequisites

To use the database operations monitor, the tables of the XDB interface must be correctly filled.

Features

To call the DBA operations monitor, choose *Tools* → *CCMS* → *DB administration* → *Operations monitor* Alternatively, call Transaction *DB24*.

In the group box *Overview* is an overview of the status of database operations that are running or have finished as well the current setup for the database operations monitor. The individual database operations are displayed in a table. The table includes the following columns:

Column	Meaning
RC	Result of the database operation: *Green* means “Everything OK” Yellow means “Warning” *Red* means “Error“
Status	Status of the operation: *STARTED* The operation started. However information about the operation is not yet available (for example, percent complete, or estimated remaining time). *RUNNING* The operation is running and information about the operation is available. *COMPLETED* The operation is finished
Date	Date of the operation

Time	Start time of the operation
FID	Type of operation
Object	Database objects that are affected by the operation or are key
Runtime	Runtime of the operation
Program	Program that executes the operation (for example, SAPDBA)
Description	Short description of the operation

Activities

Displaying specific database operations

To display specific database operations (for example, backup/recovery operations), choose the corresponding pushbutton, or the menu option under *DB operation*. Choose *All database operations* to display all DB operations.

Displaying specific views

To display specific views (for example, to display those database operations with status “warning”, or those with status “error”) choose *Setup* → *View*. Then choose either *Warning* or *Error*. Choose *All operations*, to display all database operations. To set the time period for the view, choose *Setup* → *View* → *Time period*.

Updating display

To update the display of database operations, choose *Edit* → *Refresh*, or choose *Refresh* on the application toolbar.

Automatically refresh display

To automatically refresh the display of database operations, choose *Setup* → *Auto-refresh* → *Activate*. To deactivate automatic refresh of the display, choose *Deactivate*. To set the time interval for the refresh, choose *Setup* → *Auto-Refresh* → *Time Interval*.

Deleting table entries

To delete database operations from the table, select the desired entries and choose *Edit* → *Delete*.

Automatically deleting table entries

To automatically delete table entries, choose *Setup* → *Auto-delete* → *Activate* Choose *Deactivate* to deactivate automatic delete. To set the time interval for the refresh, choose *Setup* → *Auto-delete* → *Time Interval*.

Sorting table entries

To sort columns in ascending or descending order, first select the relevant column. Then choose *Edit* → *Sort* and *Descending*, or *Ascending*. Alternately, choose *Sort in descending order* or *Sort in ascending order* on the application tool bar.

Printing table entries

To print table entries, choose *Print*

Saving settings

To save all your settings (view, auto-refresh, and auto-delete), choose *Setup* → *Save*.

Details about database operations

For details about operations (for example, remaining time for the operation, or directory and name of the log file), double-click the table entry, or select the table entry and choose *Display details* on the application toolbar. On the next screen, choose *Refresh* on the application toolbar to update the display. Use *Display action log* to display a detailed log for the database operation.

Monitoring the Database with the Alert Monitor (Informix)

Use

Using the alert monitor [Seite 812], you can monitor the following database alerts:

- Space management – reorganization and space monitoring
- DB system check – consistency and profile
- Backup/restore

The alerts described in this section are all triggered automatically from DB System Checks in CCMS (Informix) [Seite 1226].

Prerequisites

- Changing the default thresholds

  Although SAP recommends that you do not normally change the default thresholds, you can set thresholds yourself for most of the database alerts.
- Deactivating an alert

  You can deactivate an alert, but only do this for a particular reason and for a short time.

For more information, see Configuring DB System Checks in CCMS (Informix) [Seite 1227].

Procedure

1. Start the alert monitor. Refer to Starting the Alert Monitor [Seite 837].
2. Choose *SAP CCMS Monitor Templates*.
3. Choose *Entire System*.
4. Open the *Database* monitoring tree element (MTE).

   For each R/3 instance, the alerts are displayed with color coding to indicate severity. The following alerts are possible in the *Database* MTE:

MTE	Meaning	For more information, see
SpaceManagement	Monitors the space situation in your database	Management of Database Growth [Extern]
Reorganization	Checks if reorganization or application data archive required, due to tables running out of extents or running out of allocated pages	Reorganization with SAPDBA [Extern] Application Data Archiving [Extern] Analyzing Tables by Fill Level, Size, and Extents with SAPDBA [Extern]

SpaceMonitoring	Checks if dbspace fill level OK and if tables can be extended correctly	Listing Dbspaces with SAPDBA [Extern] Analyzing Tables for Critical Next Extent Size with SAPDBA [Extern] See also next step.
DBSystemCheck	Checks key aspects of your database system	DB System Checks in CCMS (Informix) [Seite 1226]
DB Consistency	Checks if chunk sizes are within limit, if raw devices are overlapping, and if logging mode OK	Listing Chunks with SAPDBA [Extern] Logging Mode with SAPDBA [Extern]
DB Profile	Checks value of settings in the `ONCONFIG` file	Editing the ONCONFIG File for ON-Archive [Extern]
Backup	Checks aspects affecting database backup and archive	Database Backup (ON-Bar) [Extern] Archive (ON-Archive and ontape) [Extern]
Restore	Checks number of chunks for a dbspace	Listing Chunks with SAPDBA [Extern]

To get up-to-date and detailed information about what the alerts mean and how you should react, use the online help in the R/3 System.

5. If you have the *SpaceMonitoring* alert for a dbspace, choose *Start analysis tool* to extend the dbspace. Refer to Extending a Dbspace with CCMS [Extern].

Result

By using the database alert monitor continually during productive database operation, you can find out quickly and easily whether your database has problems. The result is a more highly tuned database and reduced system downtime.

SAP/DB2 Universal Database DBA in CCMS

Purpose

This component enables you to manage your DB2 Universal Database using the Computing Center Management System (CCMS). With CCMS, you get extensive support for database administration (DBA) for the DB2 Universal Database and can perform many DBA functions from within the R/3 System.

Integration

You can perform most regularly-recurring database administration tasks from within the R/3 System by using the DBA tools of the CCMS.

Area of Administration:	Can Be Performed Using:
Installation	Operating system
Data backup Back up database files, archive logs	R/3 (CCMS), operating system, SAP-DB2admin
Performance tuning Analyze database performance data	R/3 (CCMS), operating system, SAP-DB2admin
Optimizer management Update optimizer statistics	R/3 (CCMS), operating system, SAP-DB2admin
Monitoring Track the database with the alert monitor, check database	R/3 (CCMS)
Database check Space analysis (number of objects, fragmentation, bottlenecks), configuration check (log directory, DB(M) parameters)	R/3 (CCMS), SAP-DB2admin
Tables/Indexes Analyze tables and indexes	R/3 (CCMS)
Reorganization	R/3 (CCMS)
Other (Recovery)	Operating system, SAP-DB2admin

Features

The main DBA features for DB2 Universal Database in CCMS are:

Area of Administration	Can Be Performed in CCMS Using:

Scheduling archive, backup, update statistics, and other tasks	DBA Planning Calendar [Seite 1244]
Update statistics	Update Statistics [Seite 1257]
Performance tuning and monitoring	Database Monitor [Seite 527] and Database Alert Monitor [Seite 812]
Checking backup results	Backup Logs [Extern] and Database Alert Monitor [Seite 812]
System checks (that is, configuration and performance)	Database Alert Monitor [Seite 812]
Disk space check	Database Alert Monitor [Seite 812]

Constraints

You have to perform certain DBA functions outside the R/3 System.

See Also

BC R/3 Database Guide: DB2 Universal Database [Extern]

Using the DBA Planning Calendar (DB2 UDB)

The DBA Planning Calendar is a tool for automating database operations. With the calendar, you can schedule operations such as online backups, have them automatically carried out, and then check that the operation was successful.

Use this procedure to set up and use the DBA Planning Calendar.

> You cannot perform all of the tasks required for database administration from the DBA Planning Calendar. For help with tasks that you must perform with the R/3 System down, see the links under **See Also**.

Prerequisites

Before using the calendar for the first time, check that these prerequisites have been met:

See: Prerequisites for Using the DBA Planning Calendar (DB2 UDB) [Seite 1246]

Procedure

1. Start the DBA Planning Calendar. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar*.
2. Choose from one of the predefined action patterns.

   Using a predefined action pattern ensures that you follow SAP's guidelines for database actions that can be carried out while R/3 is up.

   Choosing a Predefined Action Pattern (DB2 UDB) [Seite 1247]
3. Edit the action pattern to make any required changes to actions or start times. Click each action that you wish to change.

   See:
   - DBA Planning Calendar (DB2 UDB) [Seite 1244]
   - Scheduling Actions in the DBA Planning Calendar (DB2 UDB) [Seite 1249]
4. Use the DBA Planning Calendar. You must check the calendar daily to be sure that scheduled actions have been carried out correctly.
   - Checking the Results of Actions (DB2 UDB) [Seite 1254].
   - Displaying the Status of a Day's Actions (DB2 UDB) [Seite 1253]
   - Troubleshooting (DB2 UDB) [Seite 1256]

Result

At the end of this process, you will be able to use the DBA Planning Calendar to automate some of your database administration tasks.

See Also

SAP/DB2 Universal Database DBA in CCMS [Seite 1240]

SAP/DB2 UDB Database Monitor [Seite 527]

BC R/3 Database Guide: DB2 Universal Database [Extern]

DBA Planning Calendar (DB2 UDB)

Use

Some database administration activities, such as database backups, have to be performed regularly. You can schedule and coordinate these tasks using the DBA Planning Calendar.

The main function of the DBA Planning Calendar is to define the start times and parameters for database activities. Since these activities run without interaction with the administrator, you have to make sure in advance that the necessary resources will be available.

Integration

The DBA Planning Calendar is part of the Computing Center Management System (CCMS).

Features

Actions can be scheduled in advance using R/3 background processing. These actions are then executed automatically.

Activities

These plannable actions are currently available for DB2 Universal Database:

- *Update statistics and reorganization check (DBSTATC)*

  Updating of statistics and reorganization check for tables defined in DBSTATC.

- *Update statistics and reorganization check for all tables*

  Updating of statistics and reorganization check for all database tables.

- *Archive inactive log files onto device*
- *Archive inactive log files into ADSM*
- *Full database backup of the database*
- *Reorganize flagged tables and update statistics*

  A selection of tables (max. 100) requiring reorganization is displayed. Information on runtime and size is given for each table on the basis of which the approximate time required for the reorganization can be calculated.

- *Reorganize tables in tablespace(s)*

  Reorganizes all tables of a specific tablespace using primary indexes. The table statistics are updated.

- *Initialize tape in device*

See Also

Using the DBA Planning Calendar (DB2 UDB) [Seite 1242]

Prerequisites for Using the DBA Planning Calendar (DB2 UDB)

To make sure that database administration tasks are secure, authorization checks must be made for certain operations both in the R/3 System and in the database system.

- **R/3 System Prerequisites**

  You need authorization for database administration and background job scheduling. This is provided by the profiles `S_RZL_ADMIN` and `S_BTCH_ALL`.

  External programs must be able to run on the database server so that actions affecting the database can be carried out from other application servers.

- **Database System Prerequisites**

  Some actions require authorization to connect to the database. Others (such as online backup) require even higher levels of authorization.

  For more information about preparatory tasks that ensure correct setup of the environment, see R/3 Note 44190.

  Before scheduling database backups, see the information on backing up the database in BC R/3 Database Guide: DB2 Universal Database [Extern].

- **Hardware Prerequisites**

  The number of tapes required depends on the backup strategy you decide on. The tapes must already be in the specified tape drive or tape changer, so that the operator does not have to interrupt scheduled backup runs.

  Different tape drives are needed if logs are to be archived with periodic database backups.

Choosing a Predefined Action Pattern (DB2 UDB)

Use

The DBA Planning Calendar provides easy-to-use predefined action patterns for backups, archiving, statistics, and reorganization checks. You simply specify a reference time, on the basis of which all schedules are defined. You can change or delete plans in an action pattern at a later time. However, a backup is safer if you use a predefined action pattern.

The scheduled actions are checked for clashes with other existing actions. Any clashes found are listed, but no actions are actually activated. You must resolve the clashes yourself, either by changing the action pattern's reference time or changing the scheduling time of any actions that clash.

Prerequisites

Refer to Prerequisites for Using the DBA Planning Calendar (DB2 UDB) [Seite 1246].

Procedure

1. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar* → *Planning* → *Action Pattern*.
2. Choose a predefined action pattern from the list. The following predefined action patterns exist:
   - *Upd. Statistics + Reorgcheck all tables (weekly)*
   - *Archive inactive log files into ADSM (daily)*
   - *Initialize tape and archive inactive log files (daily)*

     The default is for the tape to be initialized with the name `<SID>A<DATE>` (<SID> is the SAP system ID and <DATE> the current day of the month) and the device `/dev/rmt0` (for NT `/dev/mt0`, and for Solaris `/dev/rmt/0`, and for Linux /dev/st0). You can either accept these proposed values or change them as appropriate for your naming conventions or hardware.
   - *Update Statistics and Reorgcheck on DBSTATC (daily)*
   - *Full online backup of database to device (daily)*
   - *Full online backup of database into ADSM (daily)*
   - *Full online backup of database with vendor library (daily)*
3. Accept or change the parameters required for the pattern you have selected.
4. Weekly action patterns:
   Select the day and time for which you want the pattern to be scheduled.
   Daily action patterns:
   Select the days for which you want the pattern to be scheduled. Select the time once for all the days selected.
5. If there are conflicts between the action pattern you have chosen and activities that are already scheduled in the DBA Planning Calendar, the system displays a list of those conflicts.

Print the list with the `Shift-F1` key combination. Then choose *Cancel*. No activities from the new action pattern are scheduled.

Review and eliminate the conflicts before trying to choose the action pattern again.

To delete a predefined action pattern, you have to delete the next scheduled occurrence of the action that was scheduled via planning strategy. All future scheduling of the action is deleted.

Result

If there are no conflicts, the actions in the action pattern are automatically inserted into the DBA Planning Calendar. The system also schedules background jobs to perform the actions. All jobs are scheduled for periodic repetition according to the schedule in the action pattern.

Maintaining Actions in the DBA Planning Calendar (DB2 UDB)

Prerequisites

- If you want to change or delete an action, it must be in the state *SCHED* (that is, not already executed).
- If you want to insert an action, you must choose today or a later day, and if you choose today, you must choose a time after the current time.
- If an action has already been executed, you can only display it. See Checking the Results of Actions (DB2 UDB) [Seite 1254].

Procedure

1. To start the DBA Planning Calendar, select *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar*.
2. Choose the day you want by double-clicking the day's header bar.
   The system displays actions already scheduled on the chosen day.

   If an action is one of a sequence, you can only change or delete the next scheduled occurrence of the action. If you do this, the system also deletes all future occurrences of the action in the same sequence.

   For example, you cannot change or delete an action scheduled to run in six weeks' time, if the next action of the same sequence is scheduled to run next week. Instead, you have to change or delete the occurrence for next week.

Function	**Procedure**
Add an action	See Scheduling Actions in the DBA Planning Calendar (DB2 UDB) [Seite 1251].
Change an action	1. Select the action you wish to change. 2. Choose *Change*. The system shows the basic parameters set for the action. Refer to the table in Scheduling Actions in the DBA Planning Calendar (DB2 UDB) [Seite 1251]. 3. If required, change the basic parameters for the action. The system warns you if there is a conflict with an existing action. If so, you must choose another time for the action. 4. If required, choose *Parameters* to change the parameters specific to the action. 5. Choose *Continue* to save your changes. 6. Make sure any resources required by your change are available.

Delete an action	1. Select the action you wish to delete. 2. Choose *Delete*. The system asks you to confirm the deletion. 3. Confirm the deletion. The system deletes the action including the corresponding background job. Deletion of the job also stops automatic repetition of the action, if that was scheduled.

Result

The schedule of the DBA Planning Calendar is updated with the results of your insertion, change or deletion.

See Also

Choosing a Predefined Action Pattern (DB2 UDB) [Seite 1247]

DBA Planning Calendar (DB2 UDB) [Seite 1244]

Scheduling Actions in the DBA Planning Calendar (DB2 UDB)

Use

This section describes how to schedule actions in the DBA Planning Calendar. From Release 4.5B variable action patterns are available. For information on how to schedule these patterns, see Choosing a Predefined Action Pattern (DB2 UDB) [Seite 1247].

Procedure

1. Choose the day you want by double-clicking the day's header bar.
2. Choose *Add*.

   The system displays the actions supported by the DBA Planning Calendar for your database platform.
3. Select the action you want to schedule.

   The system shows the basic parameters currently set for the action.
4. Enter the basic parameters for the action as follows:

Parameter	What to enter	For example
Start time	• The time when the action is to start, using 24-hour clock notation. • Choose *Start immediately*, if you are entering an action for today and want to start the action immediately.	`17:00` This means the job is to be executed at 5 o'clock in the afternoon.
Period	The interval for the action, in weeks. The action is repeated at the interval you enter. If you do not enter a period, the action is run once only.	`2` This means the action is to be repeated on the same day and time every two weeks.
Calendar	Select the calendar for your country or area.	`US` This means the calendar for the United States is to be used.

   The system warns you if there is a conflict with an existing action. If so, you must choose another time for the action.

   Depending on the action you are inserting, the system may prompt for further input parameters.
5. If necessary, enter further input parameters.
6. Choose *Continue* to insert the action.

Result

The schedule of the DBA Planning Calendar is updated with the results of your insertion.

Displaying the Status of a Day's Actions (DB2 UDB)

1. Click the header bar for a particular day.

   Unsuccessful or interrupted actions are shown in **red** in the DBA Planning Calendar and, if there are only unsuccessful or interrupted actions, then the header bar for that day is also highlighted in **red**.

   In the scheduling overview, you can see if any logs were written for an action.

2. To survey the action logs or background job log for a particular action, select *Action logs* or *Job logs* as appropriate. All logs written on the day you have selected are listed. Some actions do **not** write logs.

   The timestamp is used to assign logs to scheduled actions. An action log is assigned to the action which has the same type and the closest corresponding timestamp. In some cases, for example, if no background work process is available, the action is delayed and even postponed until after the next scheduling time. Unfortunately this means that the action log is then assigned to the next scheduling time and the original scheduling time log is incorrect. This is the case if the logs for the previous schedules are displayed for the next schedule of the same type.

See Also

Background Processing: Concepts and Features [Seite 74]

Checking the Results of Actions (DB2 UDB)

Use

Check regularly that the scheduled actions are running correctly. The DBA Planning Calendar allows you to display and check the status of an action. In addition, the most important actions scheduled in the DBA Planning Calendar, for example backups and log file archiving, generate logs, and also give the user details of the results of an actions. Background jobs are used to schedule actions and these background jobs generate job logs. You can view all the information using the DBA Planning Calendar. In addition, all previous backup actions can be displayed.

Displaying the Status of Scheduled Actions

1. Click the header bar for a day. The following status information is displayed:

State	Explanation
SCHED	The action has been scheduled, but the scheduled time has not yet been reached.
OK	The action has been completed successfully.
ERROR	The action has terminated abnormally.
N.A.	No log giving information about the status of the action is available. It is possible that the action is currently being executed.
TIME	The action was executed until the maximum runtime was reached.
WARN	The action ended with a warning.
CANC	The action was canceled.
ABORT	The action was aborted.

2. Unsuccessful actions are highlighted in **red** in the DBA Planning Calendar.

Displaying Action Logs and Job Logs

Use

From the DBA Planning Calendar, you can display the logs that most actions generate. These logs detail the results of an action.

You can also display the job logs that are generated by the background jobs that carry out DBA Planning Calendar actions.

Procedure

1. To see whether a database log (action log) was generated by a particular action, click the header bar of a particular day.

   A mark in the *Log* field shows that a log was written.

2. To display available action logs and job logs, put the cursor on a particular day.

   Choose *Action logs* to display a list of the database action logs generated on that day.

Choose *Job logs* to see the background processing job logs generated by the actions on that day.

Log and action timestamps are used to associate database logs and scheduled actions. An action log is assigned to the action which has the same type and the closest corresponding time stamp. In rare cases, this assignment mechanism can result in false association of a log and action.

Troubleshooting (DB2 UDB)

Use

Any scheduled action can occasionally fail, so you must at least check the more critical actions such as database backup daily.

See also:

Checking the Results of Actions (DB2 UDB) [Seite 1254]

Procedure

1. Was the background job executed correctly? Here you can consult the job log, and if no job log exists then the background job was probably not started. You can get more details using the job overview with transaction SM37 (note that the names of all jobs scheduled in the DBA Planning Calendar start with DBA). The job log will also tell you whether an external program was started.
2. Consult the action log (if one is available) if you are sure that the background job ran successfully.
3. You can sometimes display logs at operating system level. These logs contain details of the external program. There are several different errors that may occur when using a tape drive. The more common errors include:

– No tape in the tape drive.

– The tape in the tape drive is write protected.

– The tape in the tape drive has not been initialized.

– The tape drive contains the wrong tape.

– The tape is full.

– An error has occurred in the tape drive.

4. If no further log files exist for programs executed on the operating system level (BRARCHIVE), consult the program documentation.
5. Once the error has been corrected, carry out the action manually using *Start immediately*, making sure there are no clashes with other scheduled actions.

See Also

Initiating a Log Archive Backup Manually (DB2 UDB) [Seite 1267]

Update Statistics (DB2 UDB)

Use

You can use the DBA Planning Calendar to schedule an update of the database statistics. There are two different jobs that can be scheduled:

- *Upd. Statistics + Reorgcheck all tables*
- *Update Statistics + Reorgcheck (DBSTATC)*

Scheduling Upd. Statistics +Reorgcheck All Tables – DB2 UDB [Seite 1259]

Upd. Statistics + Reorgcheck all tables operates on all tables and may take several hours depending on the size of the database. You should schedule this job to run once a week.

Scheduling Update Statistics + Reorgcheck (DBSTATC) – DB2 UDB [Seite 1258]

Update Statistics + Reorgcheck (DBSTATC) operates on a subset of tables entered in the DBSTATC table. You should schedule this job to run daily.

To make work easier, you can alternatively use the following action patterns:

- *Upd. Statistics + Reorgcheck all tables (weekly)*
- *Upd.Statistics + Reorgcheck on DBSTATC (daily)*

To access the action patterns, select *Tools* → *CCMS* → *DB administration* → *DBA Planning Calendar* → *Planning* → *Action Pattern*.

For more information about using the action patterns, see:

Choosing a Predefined Action Pattern – DB2 UDB [Seite 1247]

Activities

If you decide to use one of the patterns, you only have to specify the execution time. The rest is done automatically.

Since the job *Upd. Statistics +Reorgcheck all tables* affects system performance, you should schedule it to run outside normal working hours, for example on Sundays.

Scheduling Update Statistics + Reorgcheck (DBSTATC) (DB2 UDB)

1. Select *Administration* → *CCMS* → *DB Administration* → *DBA Planning Calendar*. Alternatively, call transaction DB13.
2. Select the day on which the job is to run.
3. Select *Update Statistics + Reorgcheck (DBSTATC)*.
4. Enter the start time for the job.
5. Choose *Continue*.

Alternatively, you can use the action pattern *Check. Step, Upd.Statistics + Reorgcheck on DBSTATC (daily)* to schedule both jobs in a single step.

See Also

Scheduling Actions in the DBA Planning Calendar (DB2 UDB) [Seite 1249]

Scheduling Upd. Statistics + Reorgcheck All Tables (DB2 UDB)

Procedure

1. Choose *Tools* → *CCMS* → *DB Administration* → *DBA Planning Calendar*.
2. Select the day on which the job is to run.
3. Select *Upd. Statistics + Reorgcheck all tables*.
4. Enter the start time for the job.
5. Enter the repetition period for the job (if appropriate).

   If the job is to run once only, leave the *Period* field blank.
   To schedule the job to run weekly, enter 1.
   To schedule the job to run fortnightly, enter 2, and so on.
6. Choose *Continue*.
7. You have the option of entering a maximum runtime (in minutes) for the job.
8. Choose *Continue*.
9. You have the option of entering calculation time for tables with long fields.
10. Choose *Continue*.
11. Select the level of detail you require for the statistics.
12. Choose *Continue*.

Alternatively, you can use the action pattern *Upd. Statistics + Reorgcheck all tables (Sun.weekly)*. In this case, only the start time is selectable. The job is scheduled to run on Sundays. You do not have the option of restricting the runtime and the maximum processing duration for tables with long fields is 60 minutes. These default values are suitable for most users.

See also: Scheduling Actions in the DBA Planning Calendar (DB2 UDB) [Seite 1249]

Scheduling Strategies

- In the case of large databases, it may be advisable to restrict the execution time of the *Upd. Statistics + Reorgcheck all tables* job if the total runtime of the job exceeds the available time window.

  Assuming the job takes 10 hours to run, you could restrict the execution time of the job to 2 hours and schedule it to run on 5 days of the week, specifying an execution time of 120 minutes for each run. Since the table with the oldest statistics is processed first each time the job runs, this would ensure that all table statistics would have been updated after 5 days with 2 hours runtime per day.

- Processing tables with “long” fields

It takes much longer to process tables with “long” fields than to process other tables. To limit the total runtime of the *Upd. Statistics +Reorgcheck all tables* job, you may be advised to restrict the maximum processing time for tables with “long” fields. This maximum processing time, which you specify when you schedule the job, is the maximum processing time per table.

You reduce the value to 10 minutes. Processing of a table with “long” fields is then stopped after 10 minutes. Processing then continues with the next table.

This restriction does not affect the correctness of the database statistics. Only the calculation of the size of tables with “long” fields may become less precise.

Updating the Database Statistics for a Single Table

To update the database statistics for a single table:

- *Select Administration → System admin. → Monitor → Performance → Database → Tables/Indexes*

  or

- *Administration → CCMS → DB Administration → Cost-based optimizer → Single table statistics*

Enter the name of a table.

Select *Detailed Analysis → Tables and indexes → Run statistics*.

Database System Monitoring in CCMS (DB2 UDB)

Use

The alert monitor analyses and maintains configuration and snapshot data of the DB2 Universal Database for UNIX & Windows. It checks the contents of the admin database mirrored in the R/3 System. If these checks find critical situations, for example, if given thresholds are exceeded, alerts are raised and this enables early recognition of critical situations by the database administrator.

Integration

The monitoring functions are fully integrated into the new alert monitor and monitoring architecture.

Features

The following categories of information are currently checked:

- Disk space monitoring of the tablespaces, and file systems required for the database system
- Parameters relevant to performance
  - Access behavior of database buffers
  - Lock behavior of the application, the monitoring of deadlock situations and lock escalations
  - Recognition of potential candidates for updating table statistics
- Availability of backup and recovery mechanisms
  - Last available backup
  - Availability of the log files necessary to achieve the current state of the database from the last available backup.
  - The state of the admin database mirrored in R/3
- Configuration parameters
  - Database and database manager configuration
  - Availability data of tablespaces and containers
  - Proper clean up of the alert and performance monitoring log tables

Activities

We recommend that you check the information displayed on database system monitoring daily in the alert monitor.

See Also

Displaying the Results of Database System Monitoring (DB2 UDB) [Seite 1263]

Configuring Database System Monitoring (DB2 UDB) [Seite 1265]

Displaying the Results of Database System Monitoring (DB2 UDB)

Procedure

Text Display

To display an overview of the results of system monitoring, choose *Tools* → *CCMS* → *DB administration* → *Check* → *Monitor* or call transaction DB16. The screen *DB2 UDB System Check* - *Warnings* appears.

The results are displayed as either information messages, warnings or errors and are initially ordered by log date. You can sort in ascending or descending order. Select a column and choose the appropriate sort button.

You can limit the display to specific error levels, check categories, or partitions (only for EEE systems – Extended Enterprise Edition, the partitioned database system version). The fields *Objects* and *Attributes* enable restrictions to single attributes.

You can delete messages from any given time period. To do this:

1. Choose *Delete*.

   A dialog box appears.

2. In the *Date* field, specify the date from which you want all messages to be deleted. It is also possible to enter the component or partition as selection criteria. If you selected any other lines before by clicking the first column, only the selected lines are deleted.

   To ensure that the log table does not get too large, automatic clean-up programs run and delete entries older than seven days.

Only the most important data is displayed in the overview. You can get to the detail display by double-clicking one or more lines. Within the detail display screen, you can scroll through to other selected lines.

The detail screen has three divisions:

- Complete description of the attribute, as displayed in the alert monitor tree
- Time of message creation
- Brief description of the type of error and which value or parameter is being monitored.

It is also possible to delete a selected value in the detail display.

To go to the detail display of the configuration data of the attribute, choose *DB Alert Config* .

Display in Alert Monitor Tree

To display the results of system monitoring in the alert monitor:

1. Choose *Tools* → *CCMS* → *Control/Monitoring* → *Alert Monitor* or call transaction RZ20. The alert monitor set screen appears.

Displaying the Results of Database System Monitoring (DB2 UDB)

2. Choose the *SAP CCMS Monitor Templates,* and *Database.* The node *DB2 Universal Database for NT/UNIX* is displayed.
3. You can display information on the following categories:
   - space management
   - performance
   - backup/restore
   - health

   The checked parameters are displayed in the following colors depending on the type of error:

Error Type	Color
Information message	Green
Warning	Yellow
Error	Red

If a check resulted in a warning or error, a short text is additionally displayed next to the parameter in the *Open alerts* view.

Configuring Database System Monitoring (DB2 UDB)

DB2 UDB database system monitoring has preconfigured check categories and parameters.

Only experienced users should make changes to the system check configuration.

There are two complementary tools available for configuring database system monitoring:

- Configuration using general alert monitoring

  In this part, you can change scheduling data by entering how often the data collection program should run.

  A useful feature of DB-specific alert monitoring is the central, automated notification function that informs you of an alert by e-mail. If you want to be notified by e-mail as soon as an alert is raised, you have to define yourself as a recipient of such generated mails. To do this:

  1. Choose *Tools* → *CCMS* → *Configuration* → *Alert Monitor* or call transaction RZ21.

     The *Monitoring: Characteristics and Methods* screen appears.
  2. Choose *Method Definition* and then *Display Overview*.
  3. Scroll through the list until you find `CCMS_DB6_WATCHDOG`.
  4. Select this method and choose *List* → *Selected Entries* → *Edit*.
  5. Choose *Parameter*.
  6. Choose *Method Definitions* → *Display* ↔ *Change*.
  7. In the *SENDER* line in the *Parameter value* column, enter a valid user for your SAP System, who should be notified in the event of an alert.
  8. In the RECIPIENT line in *Parameter value* column, enter a valid user for your SAP System.
  9. Save your changes.

  Not all alerts automatically generate a mail, only those configured to do so.

- DB2 UDB-specific configuration

  1. Choose *Tools* → *CCMS* → *DB administration* → *Check* → *Configuration* or call transaction DB17.
  2. You can specify additional parameters, for example, assignment of logged values for given alerts.

Configuring Check Parameters

The initial screen of the database-specific configuration provides you with an overview of all the configuration entries. You can display details, sort entries, and make selections using the list boxes. In addition, an entry may be activated or deactivated by selection of a line in the *active* column. If you deactivate an entry, there is no further notification of corresponding alerts.

In the *Detail* display, you can configure additional parameters:

- *Threshold*

  This subscreen is divided as follows:

  – *Normal operation*

  – *Warning*: Limited operation, with reduced performance, for example

  – *Error*: Normal operation is endangered if the error is not corrected

  The following fields can be changed depending on operation status:

  – Relational operators

    You can specify how the defined comparison value should be compared with the current given value. In addition to the relational operators, you may enter whether a value should lie inside of or outside of a range of values. A full colon (:) must separate the two values. You may also specify whether or not discrete values are within a set of explicit values. Semicolons (;) must separate such values.

  – *Comparison value*

    You can specify a value, a list of values or a value range depending on the operator. This value will later be compared with the currrent measured value.

  – *Comparison value (unit of measurement)*

    You can specify the unit of measurement of the comparison value. This is important for time values, which are normally calculated internally in seconds, to be correctly converted before comparison.

  – *Automatic e-mail notification* checkbox

Values do not need to be entered for every operation status. However, you must make sure that the sum of comparison values must cover every possible value. If this is not the case, a special alert is triggered with the message:

```
There is no configuration entry for the logged value
```

The following tabs display additional information:

- *General* (transaction RZ21)

  Here you view the scheduling data from the basic alert monitor configuration. The values are displayed here for completeness. It is not possible to make changes in this transaction. You can make changes using the general maintenance function in the alert monitor.

- *Administration*

  This screen displays the user that made the last changes and whether this entry is currently active.

  After you have made your changes, save them. Changes take effect immediately.

Initiating a Log Archive Backup Manually (DB2 UDB)

If the log archives are not being backed up automatically, and you realize that the archiving directory is getting too full, back up and delete the log archives from the archiving directory manually. You can use the *immediate start* mode in the DBA Planning Calendar or the SAP-DB2admin utility for this.

SAP/DB2/400 DBA in CCMS

Purpose

This component enables you to perform some database administration tasks using the Computing Center Management System.

Integration

There are many different tasks involved in DB2/400 database administration, some of which you can perform using CCMS, as shown in the following table:

Area of administration:	**Can be performed using:**
Data backup	Operating system
Performance optimization	R/3 (CCMS), operating system
Database check (space)	R/3 (CCMS)
Reorganization	Operating system

Features

Area of administration:	**Can be performed in CCMS using:**
Performance tuning and monitoring	Database monitor [Seite 496] (Transaction ST04) and database alert monitor [Seite 812] (Transaction RZ20)
Database check	State on disk [Seite 511] (Transaction DB02)

Constraints

You have to perform many DBA functions outside the R/3 System, that is, using tools supplied by IBM.

See Also

BC R/3 Database Guide: DB2/400 [Extern]

Monitoring the Database Using the Alert Monitor (DB2/400)

Use

Using the alert monitor [Seite 812], you can monitor the following objects:

– SpaceManagement

– Performance

Procedure

5. To call the alert monitor, choose *CCMS* → *Control/Monitoring* → *Alert monitor*. Alternatively, call Transaction RZ20.
6. Expand *SAP CCMS Monitor Templates*.
7. Choose *Entire System*.
8. Open the MTE (monitor tree element) *Database*. The system displays the following monitoring objects.

Monitoring Object	Task:
SpaceManagement	Monitors the space situation in your database. State on Disk (DB2/400) [Seite 511]
Performance	Monitors database performance. Database Monitor: Detail Analysis Menu (DB2/400) [Seite 499]
Health	Monitors ASP usage. In the standard R/3 configuration, the database library is located in ASP1.The journal receivers are in ASP2. This separation avoids disk failure. When the ASP2 is filling, the system starts storing journals in ASP1. This is called ASP overflow. ASP overflow is dangerous because it does not ensure possible data recovery.

R/3 Consistency	Monitors the database tables and checks for missing unique indexes (it also takes the SAP exception table into account). Monitors consistency between ABAP Dictionary and Database. There are no thereshold values to be maintained for this monitoring object. **Green**: Unique indexes exist for all database tables. **Red**: At least one unique index for a database table missing. *Actions*: Create the missing unique indexes in the ABAP Dictionary (Transaction SE11) in your R/3 System.

To display up-to-date and detailed information about what the alerts mean and how you should react, use the online help in the R/3 System.

Result

If you use the alert monitor continually during production operation, you can find out quickly and easily whether your database has problems. The result is a more highly tuned database and reduced downtime.

SAP DB – DBA in CCMS

This section provides the database administrator with information about administering the SAP DB database system using the Computing Center Management System (CCMS). In particular, it describes how to use the DBA Planning Calendar to backup data.

The CCMS in the R/3 System offers comprehensive database administration support. CCMS allows you to perform the most important administrative tasks within the R/3 System itself without having to work at the operating system level. Logs created within the R/3 System keep you well-informed about administrative tasks you need to perform, as well as any problems within the database.

Features

You can perform most recurring database administration tasks from within the R/3 System by using the CCMS database administration tools.

These tasks include data backup, performance tuning and monitoring. If you perform these tasks regularly, you can guarantee good database performance, ensure database security with the required backups as well as recognize database problems before they become too serious.

Database Task	Carried out in:
Installation	Operating system
Data backup: Database and log backups	CCMS, operating system
Performance tuning: Analyze database performance data, update optimizer statistics	CCMS, operating system
Monitoring:Monitor database using alert monitor, check action logs	CCMS, operating system
Other tasks	Operating system

CCMS Database Administration Tools

To start the **Computing Center Management System (CCMS)**: Choose *Tools* → *CCMS* (transaction `SRZL`).

Tool	Menu path in CCMS (transaction)	Tasks
DBA Planning Calendar [Seite 1273]	*DB Administration* → *DBA Planning Calendar* (`DB13`)	Automatic scheduling and execution of backups, monitoring and analysis sessions
Displaying backup and other logs in the DBA Action Log [Seite 1289] display.	*DB Administration* → *Backup logs* (`DB12`)	Check status of data and log backups and other DBA actions

SAP DB – DBA in CCMS

Optimizer [Seite 1292] administration tools	*DB Administration → Cost-based optimizer* (DB20, DB21)	Control and monitor optimizer
Data Archiving [Seite 1386]	*DB Administration → Backup logs* (DB15)	Management of R/3 data archiving to monitor your database
Database monitor Detail analysis menu Alert monitor Database alert monitor (3.x)	*Control/Monitoring → Performance menu → Database → Activity* *Control/Monitoring → Performance menu → Database → Activity → Detail analysis menu* *Control/Monitoring → Alert monitor* *Control/Monitoring → Performance menu → Database → Database system*	SAP DB – Monitoring and Analysis [Seite 584]

DBA Planning Calendar (SAP DB)

The DBA Planning Calendar is used to automate database administration. This includes implementing, executing, and checking actions.

You can use the DBA Planning Calendar for almost all regular database administration actions.

The database actions desired are automatically executed at the time you specify. Make sure that the resources required are available before the time of execution. You can carry out periodic scheduling (in weeks). This reduces the amount of planning required for recurring actions.

Prerequisite

You can only use the DBA Planning Calendar to start actions if the R/3 System is active and available.

If you want to execute an action that requires the R/3 System to be inactive (for example, recovering the database system), use the database manager.

You are assigned the SAP DB database administrator role.

Further prerequisites [Seite 1275]

Procedure

1. In the SAP_BC_DB_ADMIN_SAPDB role choose *Database operation* → *DBA Planning Calendar* (transaction `DB13`). The *SAP DB: System Monitoring Analysis* Screen appears.
2. Prepare a suitable **plan for scheduling the following database activities**:
   Data backups, log backups, statistics updates, consistency checks

   See also: Scheduling Concept [Seite 1276]
3. Schedule **data backups.**
   For information about how to do this, see Adding an Action [Seite 1282]

   You can use a Pre-Defined Action Pattern [Seite 1278] for backing up the data. If you choose a pre-defined action pattern, the R/3 System enters the required activities into the DBA Planning Calendar. The R/3 System also schedules the background jobs that carry out the activities.
4. Activate **automatic log backup**

   For information about how to activate Automatic Log Backup [Seite 1280], see Adding an Action [Seite 1282].
5. Schedule **table statistic updates.**

   For information about how to schedule UPDATE STATISTICS sessions, see Adding an Action [Seite 1282] (see also: CCMS Optimizer [Seite 1292]).
6. Schedule a **consistency check.**

   For information about how to do this, see Adding an Action [Seite 1282].
6. Edit the schedule to make any required changes to activities or start times.

   Adding an Action [Seite 1282]

   Changing or Deleting an Action [Seite 1284]

DBA Planning Calendar (SAP DB)

7. Evaluate the DBA Planning Calendar
   Check the calendar daily to be sure that scheduled activities have been executed correctly. Here is a suggested cycle of activities for using the Calendar:

   - Start the DBA Planning Calendar and check that the actions scheduled for the previous day were executed successfully.

     Action Results [Seite 1286]

   - To go to the backup log display in the backup monitor from the DBA Planning Calendar, choose *Environment* → *Action logs*. Using the R*ecovery report* in the DBA action log display, you can determine if the existing backups are complete enough to recover the database system in case of an error.

     Display logs in the DBA Action Log [Seite 1289] display.

   - Make sure that the data media for the next scheduled backup are installed and ready to go.

     To determine which data media are required for the next backup, choose *Volumes* in the DBA Planning Calendar.

   - If any action did not run successfully, analyze the problem and repeat the action if necessary.

     Troubleshooting [Seite 1287]

Prerequisite (SAP DB)

- **R/3 System**

  In order to use the DBA Planning Calendar, users need authorization for database administration and background job scheduling.

  Enter the **S_RZL_ADMIN** and **S_BTCH_ALL** profiles for the database administrator (Profile Maintenance (authorization object S_RZL_ADM) [Seite 278]) and authorizations for background processing)

  External programs must be able to run on the database server so that actions affecting the database can be carried out from other application servers.

- **Hardware**

  The number of media required depends on your backup strategy. The medium must already be in the specified backup drive (tape drive or tape changer), so that the database administrator does not have to interrupt a scheduled backup session that is running.

  Always save the log area to the file system (media type V: version files).

- **Backups**

  If possible, use the autosave log mechanism for your log backups (see Automatic Log Backup [Seite 1280])

  The use of the database manager or **xbackup** assumes that a valid media list has been defined in the database manager

  See: *R/3 Database Manager (DBMGUI)*

Scheduling Concept (SAP DB)

Data backup

Run at least one complete data backup each week.

In addition, run an incremental data backup every day.

An incremental data backup cannot be started when a full data backup is already running.

Automatic log backup

Always activate automatic log backup (autosave log mechanism, see also Automatic Log Backup [Seite 1280]). Full and incremental data backups can be run when automatic log backup is activated.

Check regularly to verify that automatic log backup is activated. Note that the autosave log mechanism is deactivated when the database system is stopped. When the database system is restarted, you must reactivated automatic log backup. To do this, use the **Log area** *State* in the DBA Action Logs [Seite 1289] display to determine whether automatic log backup is activated.

Log backup

When automatic log backup is deactivated, you must regularly check to determine whether there is enough memory space in the log area (see also Checking Free Log Space [Seite 687]). If necessary, backup the log segments immediately. If no more log entries can be written to the log area due to insufficient free space, the database stops.

Regularly backup the version files written during the log backup to a storage medium of your choice. After backing up the version files, delete them from the server. This ensures there is always enough space in the version file directory.

You **cannot** schedule the backup of version files using the DBA Planning Calendar.

Updating statistics

After a database migration, you only need to execute *Create optimizer/space statistics* once. After that you just need to update the statistics once a week (see also: Optimizer in CCMS [Seite 1292]).

Consistency check

If possible, execute a consistency check (*Check database structure*) once a week (see also: Consistency Checks [Seite 688]).

Activities

To schedule data and log backups, statistic updates, and consistency checks, use the DBA Planning Calendar [Seite 1273]. Proceed as described in Adding an Action [Seite 1282] .

Review your scheduling concept regularly to determine whether it meets your security and backup requirements for your database system. Use this opportunity to create a recovery report (display in DBA Action Logs [Seite 1289]).

Predefined Action Patterns (SAP DB)

The DBA Planning Calendar provides easy-to-use predefined action patterns for backups.

You can change or delete activities in an action pattern.

Once you choose a particular action pattern, the activities are entered into the DBA Planning Calendar on the corresponding dates. The background jobs that perform these activities are also automatically scheduled including the appropriate repetition intervals.

Log backups are not included in the predefined action patterns. Activate Automatic Log Backup [Seite 1280] to regularly backup log segments.

Prerequisite

You have already defined the required backup media using the database manager.

You are assigned the SAP DB database administrator role

Additional prerequisites [Seite 1275]

Procedure

1. In the SAP_BC_DB_ADMIN_SAPDB role choose *Monitoring* → *Database assistant* (transaction `DB50`). The *SAP DB: System Monitoring Analysis* Screen appears.
2. Choose *Database operation* → *DBA Planning Calendar* (transaction `DB13`).
3. Choose *Calendar* → *Action pattern*
4. Choose one of the predefined action patterns from the displayed list.

   Full database backup daily; Needed medium: TAPE

   Full daily data backup (with checkpoint) on tape (default)

   SU: Full DB backup; Mo-Fr: Partial DB backup; Needed Medium: TAPE

   Sunday: Full data backup (with checkpoint) to tape, Monday to Friday: incremental data backup to tape

1. Enter the time at which the key action in the pattern is to be executed. The system suggests an appropriate time, which you can accept or change. All other scheduling is done relative to this reference time. Of course you can always change or delete any actions in an action pattern later.
2. Optionally, choose the ID of your corresponding factory calendar from the calendar list. The factory calendar affects the backup activities. It is used, for example, so that backups are not started on public holidays, when no tapes are available.
3. The activities contained in an action pattern are automatically added to the DBA Planning Calendar. The R/3 System schedules the corresponding background jobs for carrying out the activities.

   All jobs in the action pattern are scheduled to be repeated periodically (each week).

Result

After applying an action pattern, the system checks the individual activities scheduled for conflicts with other scheduled activities.

If there are conflicts between the action pattern you have chosen and activities that are already scheduled in the DBA Planning Calendar, the system displays a list of the conflicts. No plans are activated if there are conflicts. Review and eliminate the conflicts before trying to choose the action pattern again. You can eliminate the conflicts by either changing the action pattern's execution time or the scheduled time of any conflicting existing plans.

If no conflicts are found, the system displays the schedule.

To apply the changes, select *OK*; to discard the changes, select *Cancel*.

Automatic Log Backup (SAP DB)

Necessity of log backup

A complete data backup with the checkpoint (SAVEALL) is consistent by itself. This backup is not enough, however, to recover the database instance to a specific time point (for example, to the time point just before a disk error occurred).

To recover the database to that specific time point, you must import the data backup as well as the backup of the logs written after the data backup, into the database system.

A complete data backup without checkpoint (SAVEALLNCHK) is not consistent by itself. The log backup corresponding to the data backup is required as well to recreate a consistent database system. These backups are not enough, however, to recover the database instance to a specific time point (for example, to the time point just before a disk error occurred).

To recover the database to a specific time point, you must import the data backup, the corresponding log backups, and the log entries written after the data backup, into the database system.

If no more log entries can be written to the log area due to insufficient memory space (see also: Checking Free Log Space [Seite 687]), the database will stop. This is another reason you should back up the log entries regularly (see also: Scheduling Concept [Seite 1276].)

Scheduling Automatic Log Backup in the DBA Planning Calendar

You can schedule regular, automatic log backups (autosave log mechanism) using the DBA Planning Calendar [Seite 1273].

To do this, you must select the ***Enable automatic log backup*** action and specify a **version file** as a log backup medium. For information about setting up automatic log backup, see Adding an Action [Seite 1282].

Using the autosave log mechanism, log segments are automatically backed up to version files when full.

Regularly check to ensure automatic log backup is activated. To do this, you could use **Log area** *State* in the DBA Action Logs [Seite 1289] display.

If no more space is available, the autosave log mechanism stops. Check space availability regularly to ensure the version files can always be written. Back up the old version files to a different tape. Backup of version files **cannot** be scheduled using the DBA Planning Calendar.

Automatic log backup is started as soon as a log segment is full. Because it is impossible to tell when the automatic log backup will start, always ensure there is sufficient disk space available to save the log segments. Backup already created version files to a different tape so you can delete the files from the hard disk.

If log segments have not been configured (log segment size = 0), the system assumes a log segment size equal to a third of the entire log area.

Scheduling Log Backup in the DBA Planning Calendar

If automatic log backup is not activated, check regularly to determine whether there is enough memory space in the log area (Checking Free Log Space [Seite 687]) and backup the logs immediately, if necessary.

You schedule log backups using the DBA Planning Calendar.

To schedule log backups, proceed as described in Adding an Action [Seite 1282]. To do this, you must specify the *Log backup* action and select a **version file** as a log backup medium.

Before scheduling the log backups, you must define backup media for the log backups (version files) using the database manager.

Adding an Action (SAP DB)

Prerequisite

You have created any required backup media using the database manager.

You are assigned the SAP DB database administrator role.

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role choose *Monitoring* → *Database assistant* (transaction `DB50`). The *SAP DB: System Monitoring Analysis* Screen appears.

Choose *Database operation* → *DBA Planning Calendar* (transaction `DB13`).

- If you want to add an action, you must specify the current date or a date in the future. If you choose the current date, you must specify a time later than the current time.
- If an action was already executed, you can only display it.

Adding a New Database Action

1. Place the cursor on a day to which you want to add a new action. Double-click the day and the *Create a new action for* <*date*> window appears including the actions supported by the DBA Planning Calendar.

   StartTime: Start time for the action. You can trigger an action planned for the current day immediately by choosing ***Start immediately***. This allows you to start an action manually if, for example, the planned action was unsuccessful.

   Period (optional): Repeat period in weeks during which the action is automatically restarted. The system then regularly restarts the action within the interval you specify. Without a repeat period, the action is run only once.

   Calendar (optional): ID of the factory calendar to be used. The system then does not execute the action on specified holidays, for example.

   Action: Choose from the following activities:

Check database structure (all objects)	VERI	Consistency check; database and index structure check; parallel processing
Check database structure (only tables)	VERIX	Consistency check; database table structure check; parallel processing
Complete data backup with checkpoint	SAVEALL	Full data backup with checkpoint; Backup is consistent in and of itself.
Complete data backup	SAVEALLNCHK	Full data backup without checkpoint; backup is only consistent with log
Incremental data backup with checkpoint	SAVEPAGES	Incremental data backup (with checkpoint) of pages updated since the last full or incremental data backup; backup is consistent in and of itself

Incremental data backup	SAVEPNC	Incremental data backup (without checkpoint) of pages updated since the last full or incremental data backup; backup is only consistent with log
Log backup	SAVELOG	Log backup (with checkpoint)
Enable automatic log backup	AUTO-ON	Activates Automatic Log Backup [Seite 1280]
Disable automatic log backup	AUTO-OFF	Deactivates automatic log backup
Create new optimizer/space statistics	UPD_UNCOND	Unconditional update of statistics
Check optimizer statistics	UPD_CHECK	Determines which tables need updated statistics
Refresh optimizer/space statistics	UPD_COND	Conditional update of statistics; Statistics are only updated for those tables determined by UPD_CHECK.

Follow your Scheduling Concept [Seite 1276].

1. You must specify further parameters, depending on the action you selected.
2. Choose *Continue* to finish scheduling the action.

Result

The system checks for conflicts with previously-scheduled activities. If any are found, the conflict is displayed and the action is not scheduled.

Otherwise the action is added to the Planning Calendar and the required background job is scheduled.

Editing Previously Scheduled Database Actions

Changing or Deleting an Action [Seite 1284]

Checking Action

You should check, at regular intervals, that the actions scheduled are running correctly. You can display the status of each action using the DBA Planning Calendar. The DBA Planning Calendar displays unsuccessful actions in a different color.

Action Results [Seite 1286]

Changing or Deleting an Action (SAP DB)

You can edit scheduled actions from within the DBA Planning Calendar (for example, changing the scheduled start time for an action).

You can carry out periodic scheduling (in weeks). This reduces the amount of planning required for recurring actions.

If an action is one of a sequence, you can only change or delete the **next** scheduled occurrence of the action. All future scheduling of the action is deleted.

You cannot delete all actions in an action pattern (Predefined Action Patterns [Seite 1278]) at once, only one action at a time.

> An action is to be executed once a week. The execution of this action is to change or be deleted in six weeks.
>
> To change or delete this action in six weeks, you must edit the schedule for the next week accordingly.

Prerequisite

You have created any required backup media using the database manager.

You are assigned the SAP DB database administrator role.

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role choose *Monitoring* → *Database assistant* (transaction `DB50`). The *SAP DB: System Monitoring Analysis* Screen appears.

Choose *Database operation* → *DBA Planning Calendar* (transaction `DB13`).

> Be sure that the changes you plan to make do not compromise critical database maintenance activities. For example, be sure that you do not alter or delete backup actions in such a way that you are no longer able to recover the database in the event of an error.

Procedure

1. Double-click the header bar of the day for which you wish to change or delete an action.
2. Select the action and then choose *Change action* or *Delete action*.

   Delete action: The system asks you to confirm the deletion. When you do so, the action is deleted from the DBA Planning Calendar. The corresponding background job is also deleted. Deletion of the job also stops automatic periodic repetition of the action, if scheduled. The procedure is finished.

   Change action: The system displays the *Create a new action for <date>* box and includes those actions supported by the Planning Calendar. The action you chose is selected. In the *Create a new action for <date>* window, change the start time or period for the action, or choose a different action, as needed (Adding an Action [Seite 1282]).

Alternatively, select the action you want to change or delete directly from the action line and choose *Edit* → *Change action* or *Edit* → *Delete action*.

Action Results (SAP DB)

DBA Planning Calendar

The following logs are found in the DBA Planning Calendar [Seite 1273]:

- Log of results of actions planned using the DBA Planning Calendar
- Job log, which provides information about actions planned as background jobs

If an action could not be successfully executed or the status of the action could not be determined, the action is shown in color in the DBA Planning Calendar.

Procedure

Using the DBA Planning Calendar, you can display the action logs for a particular day as follows:

1. Choose an action.
2. Choose ***Action logs.*** The system displays the action log, if available
   The log timestamp is used to assign logs to scheduled actions. An action log is assigned to an action that has the same type and the closest corresponding timestamp. In some cases, for example, if no background work process is available, the action may be delayed and even postponed until after the next scheduled time. This means that the action log is then assigned to the next scheduled time and the previous scheduled time is marked as incorrect. You can recognize this in that the system displays the same logs both for the previously scheduled time as well as for the next scheduled time of the same action.
3. Choose ***Job logs.*** The system displays the background job log for the action.

DBA Action Logs

To display the DBA action logs, choose *Environment* → *Backup logs* (transaction `DB12`).

The DBA action log display contains the following information:

- Log of the last successful or unsuccessful backup
- A list of all backup logs
- Status of the log backups
- Statistics for the database optimizer
- Results of the database checks

Procedure

DBA Action Logs [Seite 1289]

Troubleshooting (SAP DB)

On rare occasions, an action scheduled using the DBA Planning Calendar may not execute correctly, for a variety of reasons. You should therefore perform daily checks of at least the critical actions, such as database backups. (Action Results [Seite 1286])

Prerequisite

You are assigned the SAP DB database administrator role.

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role choose *Monitoring* → *Database assistant* (transaction `DB50`). The *SAP DB: System Monitoring Analysis* Screen appears.

Choose Database operation → DBA Planning Calendar (transaction `DB13`).

1. Analyzing a background job

Check the *Job logs* in the DBA Planning Calendar [Seite 1273]. If no job log exists, the background job was probably not started.

To display a list of all background jobs, choose *System* → *Services* → *Jobs* → *Job overview* (transaction `SM37`). Note that the names of all jobs scheduled in the DBA Planning Calendar start with `DBA`.

The job log can tell you whether an external program started, among other information.

2. Analyzing the action log

Check the action log (if one exists) in the DBA Planning Calendar [Seite 1273].

3. Displaying logs at the operating system level

Check the database kernel messages and the logs from the database administration tools (DBA Action Logs [Seite 1289]).

Common backup device errors:

- No medium in the backup device.
- The medium in the device is write-protected.
- The medium in the backup device has not been initialized.
- The backup device contains the wrong medium.
- The medium is full.
- An error has occurred in the backup drive.

Warnings

Use the DBA action log display and the alert monitor (3.x) or alert monitor (Release 4) to analyze errors. Critical situations are indicated using corresponding warnings and alerts.

DBA Action Logs [Seite 1289]

Troubleshooting (SAP DB)

Database Alert Monitor [Seite 677]

Alert Monitor [Seite 812] [Seite 812]

DBA Action Logs (SAP DB)

The DBA action log display provides the following information:

- Logs of database actions executed
- Messages from the database kernel
- Logs from the database administration tool (database manager)

You can also check whether the backups that have been run are sufficient to recover the database system in case of an error.

Prerequisites

You have executed backups or other actions using the DBA Planning Calendar [Seite 1273].

You are assigned the SAP DB database administrator role.

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role choose *Monitoring* → *Database assistant* (transaction `DB50`). The *SAP DB: System Monitoring Analysis* Screen appears.

Choose *Database operation* → *DB backup logs: Overview* (transaction `DB12`).

The system determines and displays the following information:

- Time of the last full data backup
- Whether the data backup is currently running or has been successfully completed
- Current log fill level
- Warnings if critical database situations have occurred

Analyze any warnings displayed.

Warning	**Explanation**
Execute a complete data backup	Last successful full data backup is too old. To ensure the security of your database system, execute a full data backup as soon as possible.
The last data backup failed.	Last full data backup failed. To ensure the security of your database system, execute a full data backup as soon as possible.
Update the optimizer statistics in your system	The latest statistics are too old. To ensure the optimizer is working effectively, update the statistics.
CAUTION: The log area is almost full. Back up the log as soon as possible or your system may stop	Back up your log immediately to prevent the log space from filling up. If no more log entries can be written, the database stops.

Choose the display you wish to use.

DBA action logs: Overview

Backups	

DBA Action Logs (SAP DB)

Last unsuccessful backup	This is only displayed if the last data backup failed. This helps you determine the cause of a failure of a scheduled backup.
Last successful backup	Log for the data backup executed at the specified time.
List of logs	List of logs for all previously scheduled and executed data backups.
Log area	
Status	Log status information
Optimizer statistics	
List of logs	List of logs for all previously scheduled and executed UPDATE STATISTICS sessions
Consistency checks	
List of logs	List of logs for all previously scheduled and executed consistency checks
Recovery report	List of logs for all full and incremental data backups that ran after the last successful full data backup
Recovery with incremental backups	List of all backups (incremental data backups and log backups) that are required along with the last successful full data backup to recover the database system as quickly as possible.
Recovery without incremental backups	List of all log backups, since the last successful full data backup, that are required to recover the database system.
Extras → DB messages	Messages from the database kernel (Database Message Log [Seite 607]) *Current:* System messages (`knldiag`) *Old:* Old system messages (`knldiag.old`) *Error:* System error messages (`knldiag.err`)
Extras → DBA logs	Logs from the database manager. *Log:* Log of the command string of the DBA actions (for example, `dbm.prt`) (Control Log File [Seite 606]) *History:* Database kernel history of past save and restore requests (for example, `dbm.knl`) *Kernel commands*: Database kernel logs for the internal utility requests (save and restore requests, add devspace, update statistics, for example `dbm.utl`) *Media definitions:* Defined backup media (for example, `dbm.mmm`) *Media use:* Contents of backup media (for example, `dbm.mdf`) *Load system tables:* Log of last serverdb installation (for example, `dbm.ins`)

See also:

Database Monitor [Seite 585]

Database Alert Monitor [Seite 677]

Alert Monitor [Seite 812](Release 4)

Detail Analysis Menu [Seite 592] [Seite 812]

Checking Free Log Space (SAP DB) [Seite 687]

Optimizer in CCMS (SAP DB)

Optimizer Description

SQL statements describe what information data is to contain. However, they do not describe how the data is to be read or written. Therefore, there must be an instance responsible for translating the SQL statements into concrete access operations. This instance is called the Optimizer.

Use

Every SQL statement can be converted into a variety of access operations leading to varying length run times, depending on the amount of data to be processed. The job of the Optimizer is to process data as cost-effectively as possible.

SAP DB operates using the cost-based optimizer alone, that is, there are no background rules indicating how data can be accessed. Data can be accessed on single tables (basis component tables) or on multiple tables (joins). For joins, the optimizer is designed to determine the fastest access plan for the tables included in the join. The optimizer determines an access strategy for multiple tables using statistics.

The cost-based optimizer uses statistics created by UPDATE STATISTIC sessions. The database system provides two methods for creating these statistics:

- The database system automatically creates these statistics (implicit UPDATE STATISTICS).
- The administrator can execute statistic updates (explicit UPDATE STATISTICS).

Implicit UPDATE STATISTICS

Implicit UPDATE STATISTICS is switched on when the database is started and should remain active while the database is running.

If, when accessing a table, the system determines that the statistics for choosing the most cost–effective strategy are no longer current, the database system itself will activate an implicit UPDATE STATISTICS session for the table.

In an implicit UPDATE STATISTICS session, statistic values are not determined based on absolute values (computed values) but rather based on estimated values. The basis of the estimate is a set number of rows (20,000).

Implicit UPDATE STATISTICS should always be switched on. You can check this using the Overview [Seite 608] screen in the *Detail analysis menu* (make sure *Update statistics* is set to *ON*).
If implicit UPDATE STATISTICS is switched off, the database administrator alone is responsible for updating statistics. If this is the case, the administrator should switch on implicit UPDATE STATISTICS using the appropriate diagnostic command at the database level as soon as possible to ensure good database performance.

Explicit UPDATE STATISTICS

In general, the SAP DB database system does not require an explicit UPDATE STATISTICS. Implicit UPDATE STATISTICS, which is continually active, creates and updates the required statistics.

The UPDATE STATISTICS statement also updates other information required for the Computing Center Management System (CCMS). Therefore, an explicit UPDATE STATISTICS session should be run once a week (see Planning Concept [Seite 1276])

There are several ways of starting an explicit UPDATE STATISTICS:

- On the operating system level
- From the database manager
- From CCMS in the R/3 System

The following section describes the ways of starting and running explicit UPDATE STATISTICS from **CCMS**.

	CCMS Function:	**Procedure**
1.	Automatic recognition of database objects whose statistics need to be updated	If needed, schedule the action *Check optimizer statistics* (UPD_CHECK) into the DBA Planning Calendar (transaction `DB13`) (Adding an Action [Seite 1282]).
2.	Automatic statistics update	Plan the actions *Check optimizer statistics* (UPD_CHECK) and *Refresh optimizer/space statistics* (UPD_COND) into the DBA Planning Calendar (transaction `DB13`) (Adding an Action [Seite 1282]). The system responds with estimated values.
3.	Check UPDATE STATISTICS sessions	If necessary, check in the DBA Planning Calendar [Seite 1273] (transaction `DB13`) and in the DBA Action Log [Seite 1289] display (transaction `DB12`) whether the actions from 1 and 2 are planned regularly and whether they are running successfully. Also use the Database Alert Monitor [Seite 677] to check UPDATE STATISTICS sessions.
	In special cases, you can also use the following CCMS functions:	
	Create and update single table statistics	You can check statistics for specific single tables and if necessary, update those statistics (transaction `DB20` Single Table Statistics [Seite 1295]). You can specify how accurately the statistics are created.

	Edit the optimizer control table (DBSTATC)	You can configure statistics updates by changing the values in the *DBSTATC* control table (transaction `DB21` Maintaining DBSTATC Control Table [Seite 1296]).

Optimizer statistics must be created for all tables after an installation or R/3 System upgrade. These procedures are documented in the installation and upgrade documentation, and are not covered here.

Single Table Statistics (SAP DB)

You can use this procedure to perform the following actions on a single table in your server database:

- Check whether the existing statistics for the table deviate significantly from the current value
- Update the statistics for the table, if required

Use

Use the following procedure if statistics have to be updated for a table. This should only be required in exceptional cases because implicit UPDATE STATISTICS sessions should keep statistics current. A request to update individual statistics might come, for example, from SAP EarlyWatch for diagnostic purposes. If data in a table has changed significantly (for example, due to batch input), it may be necessary to update the statistics for that particular table.

Prerequisite

You are assigned the SAP DB database administrator role.

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role choose *Monitoring* → *Database assistant* (transaction `DB50`). The *SAP DB: System Monitoring Analysis* Screen appears.

Choose *Database performance and configuration* → *Edit database statistics* (transaction `DB20`).

1. Enter the name of the table.

   The system displays the results from the last UPDATE STATISTICS session. This information includes the date, time and accuracy, as well as the number of rows determined (*Number of rows*, *Statistical value*).

2. To check whether the table's statistics need updating, determine the current number of rows.

   Choose *Count total numbers of rows*

   Total: Current number of table rows (*statistical value*); statistical value valid up to now
   Deviation: Deviation of the *Total* value from the *Statistical value* value

3. If the deviation between the current number and the statistical value is too large, run UPDATE STATISTICS:

   To do this, choose the *Accuracy*, with which the statistical data is to be created.

   Then choose *Refresh* to update statistics.

Result

The database system now contains current statistics for this table.

Optimizer in CCMS [Seite 1292]

Maintaining DBSTATC Control Table [Seite 1296]

Maintaining DBSTATC Control Table (SAP DB)

Changes to the DBSTATC control table are usually not required in the SAP DB database system.

The following information is provided as background information for the database administrator. SAP handles all entries for the DBSTATC table and delivers them as part of the R/3 System. The DBSTATC table entries are updated using functions of the Computing Center Management System (CCMS).

Use

The DBSTATC control table contains a list of select R/3 database tables. These tables were selected because their settings for updating the cost–based optimizer statistics deviate from the default values. The DBSTATC control table is delivered with several hundred entries for R/3 Basis component and application tables.

For all other R/3 tables, UPDATE STATISTICS sessions are executed using the default settings.

If you create or change an existing entry for a table in the DBSTATC control table, all UPDATE STATISTICS sessions will use these new values (in the database manager and in CCMS (including the DBA Planning Calendar).

Prerequisite

You are assigned the SAP DB database administrator role.

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role choose *Monitoring* → *Database assistant* (transaction `DB50`). The *SAP DB: System Monitoring Analysis* Screen appears.

Choose *Database performance and configuration* → *Database optimizer control* (transaction `DB21`).

DBSTATC Control Table

Field	Value	Explanation
DB object	*<Name>*	Table name in capital letters
Use	*0* *A*	R/3 Basis component table relevant to the optimizer (**default setting**), table relevant to the application monitor. If A is specified, additional statistics are created for this table. This procedure takes a long time to run.
Activ	*A* *N* *P* *U* *R*	Table statistics are updated, if required (**default**) Statistics are not created, but deleted Statistics are created with high priority Statistics are created during every UPDATE STATISTICS session Statistics are only created for the application table monitor

TODO flag	*X*	Tells the system to create statistics for the table
TODO date	*<Date>*	The date on which the entry in *DBSTATC* was last changed

The rest of the fields in table *DBSTATC* are either not relevant to the SAP DB database for R/3 or are automatically maintained by the R/3 System. Do not enter a value for these fields.

Changing existing entry

1. Select an entry you wish to change.
2. Choose *Goto* → *Details.*
3. Enter the required values (use the standard values).
4. Save the entry.

Adding an entry

1. Choose *Edit* → *New entries.*
2. Enter the required values (use the standard values).
3. Save the entry.

Deleting an entry

Do NOT delete any entries delivered by SAP for the application monitor, if you want to use them.

1. Select the entry you wish to delete.
2. Choose *Edit* → *Delete.*
3. Save the deletion.

The system will now use the default settings for statistics updates for the table whose entries you deleted from *DBSTATC*.

Result

Statistic updates for tables that are contained in *DBSTATC* are executed according to the entries in *DBSTATC*. Statistic values are determined based on estimated values.

Changes to these estimated values for a table become effective at the next UPDATE STATISTICS session.

Optimizer in CCMS [Seite 1292]

Single Table Statistics [Seite 1295]

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 [Extern]
- BC R/3 Database Guide: DB2 common server [Extern]
- BC R/3 Database Guide:DB2/400 [Extern]
- BC R/3 Database Guide: Informix [Extern]
- BC R/3 Database Guide: Oracle [Extern]

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

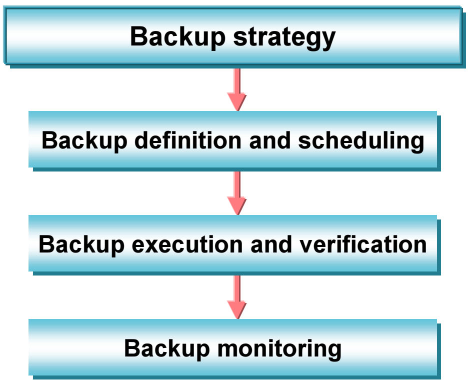
A backup can only effectively contribute to safeguarding your database, if it is performed as part of an overall backup and restore strategy. It is therefore important to first carefully work out a strategy and to test it before applying it in a productive system. It should define details such as the 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 [Seite 1302]

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

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, *msdb* database, *master* 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 [Seite 1317]

Backing Up with R/3 [Seite 1314]

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 [Seite 1319]

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 [Seite 1345]

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 [Seite 1303]

What Needs to be Backed Up? [Seite 1305]

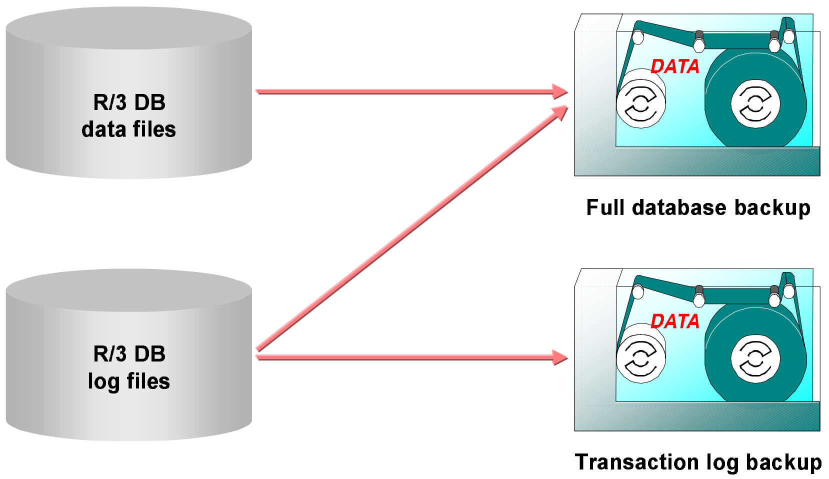
Backup Device Management [Seite 1308]

Disk Configuration [Seite 1311]

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

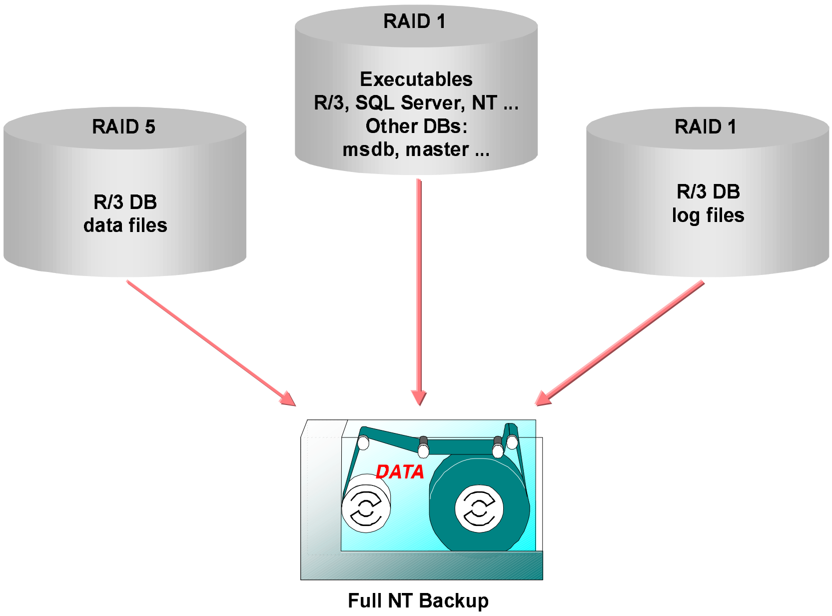
Transaction log backups are made when the database is online. As they only contain the changes since the last transaction log backup, they are much smaller than database backups and therefore have less impact on performance.

- **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 no 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 [Seite 1311] 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.

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 [Seite 1314]

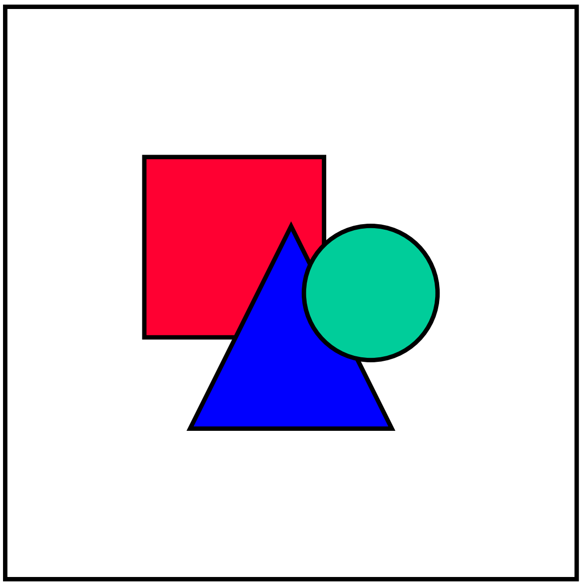
Backing Up with the Enterprise Manager [Seite 1317]

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).

- 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 [Seite 1314]

Backing Up with the Enterprise Manager [Seite 1317]

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

- 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.

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 [Seite 1314]

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 ConfigurationTable

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

Disk Configuration

C:\TEMPDB	1	*tempdb*
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 [Seite 1327]

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 [Seite 1314]

- 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 [Seite 1317]

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
Database Selection	Choose the databases you wish to back up.
Backup Device Selection	Select the backup device that the backup will be written to. Generally the default device R3DUMP0 is the correct entry here.

Details for Tapes	*Unload Tape* The tape is automatically unloaded *Initialize Tape* The existing contents of the tape are overwritten. *Verify backup* Ensures that all files that have just been written are readable. *Format tape* Formats the tape so that the existing contents and header is deleted *Expiration Period* 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.

Transaction Log

Dialog Box	Entry
Backup Device Selection	Select the device that the backup will be written to. Generally the default device R3DUMP0 is the correct entry here.
Log Backup Tape Options	Accept the expiration period of 28 days, if you are performing the backup in the framework of the overall backup strategy recommended by SAP. *Verify backup* Ensures that all files that have just been written to the backup device are readable. **Always** choose this option. If you want to write multiple log backups of one day to a single tape, set the following options: 1st backup: Select *Initialize Tape* and deselect *Unload tape* Intermediate backups: Deselect *Initialize Tape* and *Unload Tape.* Last backup: Select *Unload Tape* and deselect *Initialize Tape.*

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.

See also:

Backup Device Management [Seite 1308]

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 [Seite 1375]

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.

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 [Seite 1320]

  Checking Backups with the Enterprise Manager [Seite 1322]

- 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 [Seite 1321]

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 [Seite 1321]

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 [Seite 1341]

R/3 Planning Calendar [Seite 1373]

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 [Seite 1341]

Checking Individual Backups with R/3 [Seite 1320]

R/3 Backup Monitor [Seite 1377]

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 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 [Seite 1342]

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 [Seite 1342]

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

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 [Seite 1302]

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.

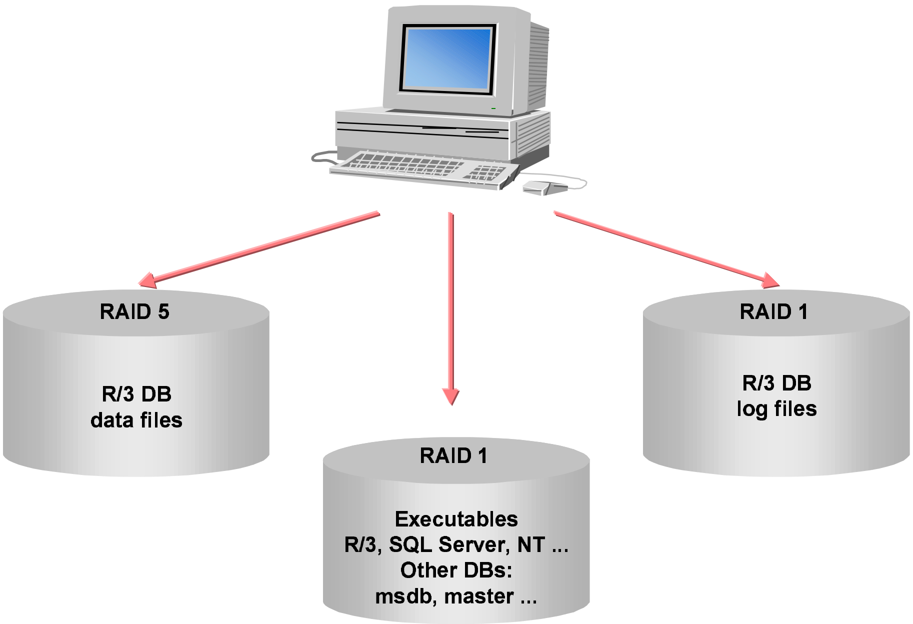
> After a power failure the database does not have to be restored. When the power is cut off, the system shuts down abruptly and the execution of transactions is interrupted. The SQL Server has an automatic recovery mechanism to deal with this situation. When the database is restarted, an automatic recovery 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 [Seite 1327]

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 [Seite 1329]

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 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 [Seite 1329].

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 [Seite 1336]

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.

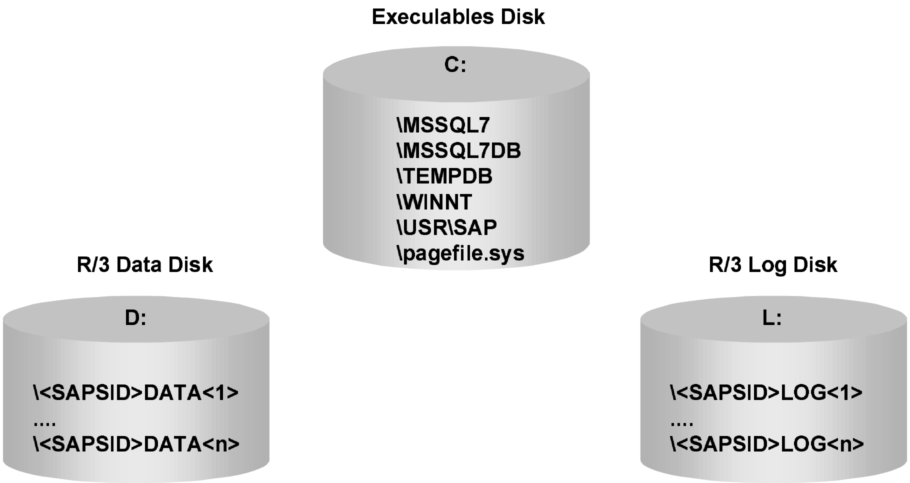
Distribution of Files to Disk

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	*msdb* and *master* databases
C:\TEMPDB	1	*tempdb*
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



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.

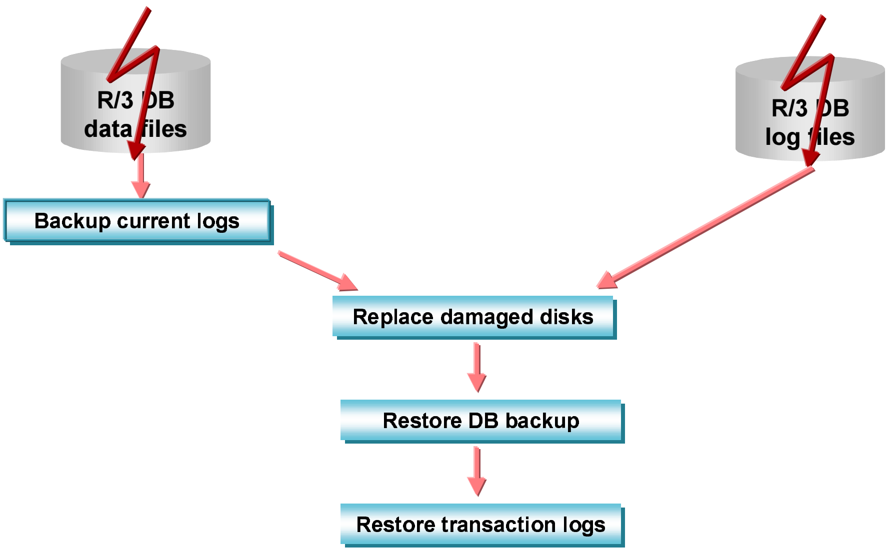
See also:

Distribution of Files to Disk [Seite 1327]

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 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 [Seite 1331]

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.

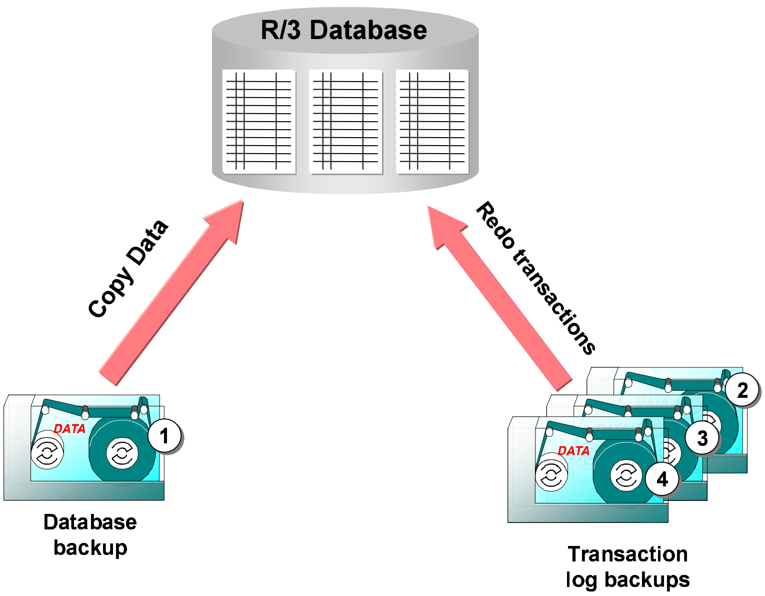
3. Database and Transaction Log Restore

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 [Seite 1332]

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

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 [Seite 1327] 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

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 [Seite 1334].

   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 and7.
   - Restore the last transaction log from device. Follow the instructions given in the section Restoring a Backup from Device [Seite 1334]. 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

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 [Seite 1332] 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*.

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.

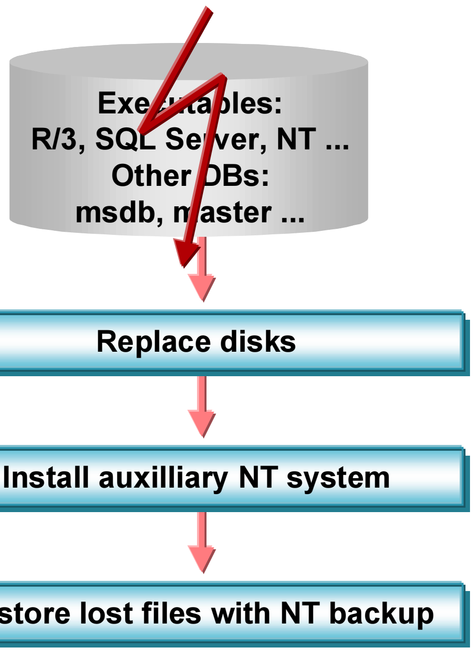
Restoring After Executables Disk Crash

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.

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 [Seite 1338]

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 [Seite 1341]

  or

- The Enterprise Manager [Seite 1342]

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 [Seite 1343]

- 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 [Seite 1343]

- 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 [Seite 1344]

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 [Seite 1345]

- The table check examines individual tables to find out whether they are physically consistent.

For more information, see Checking Consistency of Individual Tables [Seite 1347]

- 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 [Seite 1348]

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 [Seite 1342]

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 [Seite 1344].

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 databse 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

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 [Seite 1375]

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 [Seite 1341]

R/3 Static Database Monitor [Seite 1379]

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 [Seite 1379]

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

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.

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

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.

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

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 [Seite 1366].

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 [Seite 1358]

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

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:

Truncate log on checkpoint	This deletes the inactive part of the transaction log whenever a checkpoint occurs. In a production system, set this option for the `master` and `msdb` databases. No logs are required for these.
Select into/bulk copy	Set this for the `tempdb.`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*.

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.

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:

```
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

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 [Seite 1367]
- Optimize the Virtual Memory Manager [Seite 1368]
- Remove the screen saver [Seite 1369]

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

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 [Seite 1371]
- R/3 Planning Calendar [Seite 1373]
- R/3 Backup Monitor [Seite 1377]
- R/3 Static Database Monitor [Seite 1379]
- SQL Server Alert Monitor [Seite 1380]
- R/3 SQL Server Database Monitor [Seite 1385]

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 Entreprise 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:

- 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 [Seite 1299]

Database Restore [Seite 1324]

Database Server Checks [Seite 1339]

Database Server Management [Seite 1349]

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 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 [Seite 1375]

Database Management Tools [Seite 1370]

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	*Parameters*	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	*Volumes needed*	The name of the tape device and the volume that the backup will be written to are displayed.
Change parameter settings	Choose *Change* and then *Parameters* 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.

Delete scheduled action	*Delete*	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	*Action logs*	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 [Seite 1313]

Monitoring Backups [Seite 1319]

Checking Consistency of the Database [Seite 1345]

Database Management Tools [Seite 1370]

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 [Seite 1344].

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

Backup history	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 *Logs Backups* only displays the log backups.
Restoration history	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.
SQLAgent Properties	Technical data about the Agent
Backup device list	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.

Tapes needed for restore	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.
SQL Server jobs	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 *CCMS jobs*.

See also:

Database Management Tools [Seite 1370]

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
Database System	Name of the database and the time when the information on the screen were generated
Database	Space analysis for the R/3 database and transaction logs
Files	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 [Seite 1347]

Checking for Missing Indexes [Seite 1348]

Database Management Tools [Seite 1370]

R/3 SQL Server Database Monitor [Seite 1385]

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 [Seite 1381]

For instructions on how to handle the R/3 Alert Monitor see Detailed Tutorial [Seite 829]

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 [Seite 1384].

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. [Seite 1380]

To find out the threshold values which trigger alerts in the system, see Viewing and Customizing Threshold Values [Seite 1384].

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

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). [Seite 1379]

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 [Seite 750] (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 [Seite 1377] (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 [Seite 1379] (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 [Seite 1385] (transaction ST04).

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

3. Start the R/3 Alert monitor with *Administration* → *CCMS* → *Control/Monitoring* → *Alert monitor.* In the monitoring tree, choose *SAP CCMS Monitor Templates* → *Database.*
4. 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.

6. Choose the *Thrshld* tab.

   The threshold values that are currently active for the monitor are displayed.

7. To change a value, choose *Customizing* → *Display* <-> *Change.*
8. 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:

Reset	Temporarily sets the displays in the Server Engine/Elapsed and SQL Requests sections to zero.
Since Reset	Displays the values in the Server Engine/Elapsed and SQL Requests sections since you reset the display.
Refresh	Returns the main screen to the current values.
Totals per Second/Absolute Totals	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 [Seite 741]	Overview of memory configuration and performance.
Space Usage [Seite 743]	Shows disk space availability for the R/3 database.
Server Engine/Elapsed [Seite 744]	Shows CPU consumption and disk I/O.
SQL Requests [Seite 746]	Overview of SQL Server request statistics.

See also Detail Analysis Menu [Seite 747].

Data Archiving

To keep the size of a database as small as possible, you can archive application data that is no longer needed online or is only seldom accessed. Archiving objects specify from which tables data records are to be written to archive files, and whether the data records are to be deleted.

Using CCMS you can display the relationships between tables and archiving objects (see Displaying Archiving Objects and Tables [Seite 1387]) and query information about memory space and memory parameters. This information helps you to recognize the need for data archiving and to find suitable archiving objects for tables.

For more database-specific information, see:

Finding Storage Management Problems (DB2 UDB) [Seite 583]

Determining Space Information (Oracle) [Seite 1389]

Determining Space Information (SAP DB) [Seite 1390]

Determining Space Information (Informix) [Seite 1391]

Data archiving is part of the alert monitor. You can monitor and analyze the writing of archive files (write job) and the deletion of data from the database (delete jobs) during an archiving session. For more information, see Data Archiving Monitor [Seite 1020].

For a detailed description of data archiving in the R/3 System, see:

Archiving and Deleting Application Data [Extern]

Displaying Archiving Objects and Tables

Use

You can display:

a) The **Archiving objects** for a particular table

b) The **Tables, from which data was archived to an object**

Procedure

1. Choose *Tools* → *CCMS* → *DB administration* → *Data archiving*, or enter Transaction ***DB15*** in the command field. The cursor is located in the lower section of the screen (*Tables archived*).
2. To switch to *Archiving objects* (upper part of screen), choose *Table* ↔ *Object* (or choose *F5*).
3. Working in the upper half of the screen (*Archiving objects*): If you want to display the archiving objects that belong to a specific table, enter the table name in the field *Objects for table.*

   The system does not output the following archiving objects:

   - Objects that only read data and do not delete it from the table
   - Objects that archive data from the table using a view or a structure
   - Objects that archive data from the table using an archiving class for which no table information has been maintained (Transaction `ACLA`)
4. Working in the lower half of the screen (*Tables archived*): To display the tables connected to a particular archiving object, enter the name of the archiving object in the *Tables in object* field.

   The system only displays those tables from which data records are deleted when data is archived.
5. If you want to display **all** tables connected to the archiving object, select *All tables.* The system displays the names of the database tables. These names may be different from the table names in the ABAP Dictionary.

Pushbuttons are available in each half of the screen to allow you to activate the function of the inactive portion of the screen.

- If you are currently working in the upper half of the screen (*Archiving objects*), you can display the tables using the *Show tables* button.
- If you are currently working in the lower half of the screen (*Tables archived*), you can display the archiving objects using the *Show objects* button.

Result

You can use the information displayed to determine what to archive. Choose *Archive management* to go to archive management directly where you can plan and execute data archiving.

Determining Space Information (Oracle)

Use

You can determine information about space and space parameters online or use existing statistics.

Procedure

1. Choose *Tools* → *CCMS* → *DB administration* → *Data archiving*, or enter Transaction ***DB15*** in the command field.
2. Enter an archiving object and then display the corresponding tables (see Displaying Archiving Objects and Tables [Seite 1387]).

Intermediate result

Online space provides you with information obtained from the Oracle system table DBA_TABLES. This includes the number of records, the space used, and the date the last time statistics were created by Oracle. The system also displays general information about the time and the number of the last archiving session as well as its client–dependence.

Space statistics displays the corresponding information from the SAP tables DBSTATTORA and DBSTATIORA that are filled using statistic determining sessions (`sapdba -analyze`).

See CCMS Support for the Cost-Based Optimizer [Seite 1132]

If you want more details about the space and space parameters of a particular table, do the following:

3. Select a table
4. Choose *Online space* or *Space statistics*
5. Alternatively, you can select a table from the display and then choose *Space details.*

Result

You can use the information displayed to determine what to archive. Choose *Archive management* to go to archive management directly, where you can plan and execute data archiving.

Determining Space Information (SAP DB)

Use

You can determine information about space and space parameters online or use existing statistics.

Prerequisites

You are assigned the SAP DB database administrator role.

Procedure

In the SAP_BC_DB_ADMIN_SAPDB role choose *Monitoring* → *Database assistant* (transaction `DB50`). The *SAP DB: System Monitoring Analysis* Screen appears.

Choose *Database archiving* → *Data archiving: Database tables* (transaction `DB15`).

Enter an archiving object and then display the corresponding tables (Displaying Archiving Tables and Objects [Seite 1387]).

Online space: The database system provides you with information about the number of records, the amount of space used, and the date on which statistics were last created.

The system also displays general information about the time and the number of the last archiving session as well as its client–dependence.

Space statistics: The system displays relevant information from the DBSTATTADA and DBSTATIADA SAP tables. These tables are filled by UPDATE STATISTICS sessions planned in the DBA Planning Calendar (transaction `DB13`).

Additional information about a particular table: If you want detailed information about the space and space parameters of a specific table, select the table and choose *Online space* or *Space statistics.* Alternatively, you can select a table from the display and then choose *Space details.*

Result

You can use *Archive Administration* to activate or plan data archiving.

Determining Space Information (Informix)

Use

You can use this procedure to determine space and parameters of archiving objects and tables for your Informix database. For more information, see Data Archiving [Seite 1386].

Procedure

1. Choose *Tools* → *CCMS* → *DB administration* → *Data archiving*.
2. Enter an archiving object and then display the corresponding table.

   For more information on how to do this, see Displaying Archiving Objects and Tables [Seite 1387].
3. Choose *Online space* to display space information.

   The system displays information about the last archiving run for the selected object, followed by space information for each table in the object.
4. For more information about a table, select the table and choose *Space Details*.

Result

You can use the displayed information to decide what to archive. For more information if you want to start or schedule data archiving, see Archive Management [Extern].