



**PUBLIC**  
2023-03-10

# CCMS: Oracle

# Content

- 1 CCMS: Oracle. . . . . 4**
- 2 Getting Started in CCMS with Oracle DBA. . . . . 6**
- 3 The DBA Cockpit. . . . . 8**
  - 3.1 Central System Data. . . . . 10
  - 3.2 Maintenance Actions in the DBA Cockpit. . . . . 10
  - 3.3 Displaying the Audit Log. . . . . 11
  - 3.4 Configuration of Systems for Remote Monitoring. . . . . 12
    - Configuring Systems for Remote Monitoring Using Remote Database Connections. . . . . 13
    - Configuring Systems for Remote Monitoring Using the System Landscape Directory (SLD)  
. . . . . 19
  - 3.5 The DBA Cockpit on a Local System and on SAP Solution Manager. . . . . 20
    - Architecture Overview: End-to-End Monitoring and Alerting in SAP Solution Manager and DBA  
Cockpit. . . . . 23
    - Setting Up Database Monitoring and Alerting in SAP Solution Manager. . . . . 25
    - Setting Up Remote Monitoring Manually. . . . . 26
  - 3.6 Performance (Oracle). . . . . 26
    - SAP/Oracle Database Monitor. . . . . 27
    - Performance Overview. . . . . 28
    - Wait Event Analysis. . . . . 38
    - SQL Statement Analysis. . . . . 62
    - Statistical Information. . . . . 71
    - Feature Monitoring. . . . . 99
    - Additional Functions. . . . . 106
  - 3.7 Space (Oracle). . . . . 122
    - Space Overview. . . . . 123
    - Space Statistics Collector. . . . . 124
    - Database Overview. . . . . 128
    - Users Overview. . . . . 129
    - Users Detailed Analysis. . . . . 131
    - Tablespaces Overview. . . . . 133
    - Tablespaces Detailed Analysis. . . . . 138
    - Segments Overview. . . . . 141
    - Segments Detailed Analysis. . . . . 145
    - Segments Detailed Analysis (Aggregated). . . . . 149
    - Collector Logs. . . . . 152

	BW Analysis. . . . .	154
3.8	Jobs (Oracle). . . . .	156
	Central Calendar. . . . .	156
	DBA Planning Calendar (Oracle). . . . .	159
	Displaying DBA Logs (Oracle). . . . .	169
	Display of the Backup Logs (Oracle). . . . .	170
<b>4</b>	<b>Database System Check. . . . .</b>	<b>176</b>
4.1	Configuring Database System Check (Oracle). . . . .	177
4.2	Displaying Alert Messages from Database System Check. . . . .	182
<b>5</b>	<b>Update Statistics for the Cost-Based Optimizer in CCMS (Oracle). . . . .</b>	<b>185</b>
5.1	Updating Statistics in the DBA Planning Calendar (Oracle). . . . .	186
5.2	Updating Statistics for a Single Table (Oracle). . . . .	187
5.3	Updating Statistics for Groups of Tables in CCMS (Oracle). . . . .	191
5.4	Configuring Update Statistics in CCMS (Oracle). . . . .	192
<b>6</b>	<b>Maintaining Database Parameters (Oracle). . . . .</b>	<b>197</b>
<b>7</b>	<b>Monitoring the Oracle Database. . . . .</b>	<b>202</b>
7.1	Space Management Alerts. . . . .	203
7.2	Database Performance Alerts. . . . .	206
7.3	Database Backup and Restore Alerts. . . . .	208
7.4	Database Consistency Alerts. . . . .	209
7.5	Database Health Alerts. . . . .	211
7.6	Information on Oracle Database Alerts. . . . .	229

# 1 CCMS: Oracle

## Use

You can perform most regularly-recurring database administration tasks from within the SAP System by using the DBA tools of the CCMS. After installation, reorganizations and certain other tasks are the only tasks that you cannot perform from within the SAP System, as shown in the following table:

Database Task	Executed Using
Installation	SAP Software Provisioning Manager (SWPM) on operating-system level
Data backup Backup database files and redo log files	SAP (CCMS), operating system (BRARCHIVE, BRBACKUP)
Performance optimizing Monitor and optimize database performance	SAP (CCMS), operating system
Access optimizing Configure cost-based optimizer, create and update statistics	SAP (CCMS), operating system (BRCONNECT)
Monitoring Monitor database using alert monitor, check database and DBA action logs	SAP (CCMS), operating system (BRCONNECT)
Check Analyze space (number of objects, fragmentation, bottlenecks), check configuration (control file, online redo log files, INIT . ORA parameters)	SAP (CCMS), (BRCONNECT)
Reorganization Reorganize tables	Operating system (BRSPACE)
Other tasks For example, database recovery (restore and recovery)	Operating system (BRRECOVER)

## Features

The main CCMS tools for database administration are:

- [Display of the Backup Logs \[page 170\]](#) (in the DBA Cockpit): Check the status of database backups
- The [DBA Planning Calendar \[page 159\]](#) (in the DBA Cockpit): Automatically schedule and execute database operations (for example, database backups, update statistics, and database checks)
- [Displaying DBA Logs \[page 169\]](#) (in the DBA Cockpit): Access action logs created by BRSPACE, BRBACKUP, BRARCHIVE, BRCONNECT or an external program
- The [Database System Check \[page 176\]](#) (Transactions DB16 and DB17): Configure and check important database parameters to be able to recognize critical database situations in time.
- [Update Statistics \[page 185\]](#) (Transactions DB20 and DB21): Configure the cost-based optimizer, and check and update optimizer statistics
- [Online Maintenance of Database Parameters \[page 197\]](#) (Transaction DB26): Details and history of database parameters
- Maintaining Database Connection Information (Transaction DBCO): Information about additional, nonstandard database connections
- The [SAP/Oracle Database Monitor \[page 27\]](#): Detailed database performance information and histories

For more information about Oracle database administration with the SAP System, see .

## 2 Getting Started in CCMS with Oracle DBA

Before starting to use the Computing Center Management System (CCMS) for Oracle database administration (DBA) tasks you need to set up certain things.

1. Set up hardware for database backups and offline redo log backups to, for example, 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 BRBACKUP. This device changes the tapes as required by BRBACKUP and BRARCHIVE

For more information, see , 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 .

2. Set up 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 the documentation on profile maintenance and authorizations for background jobs.

### Note

External programs must be able to run on the database server so that actions affecting the database can be executed from other application servers.

3. Set up users and authorizations for the database system and operating system.

DBA Planning Calendar actions are executed under the authorizations of the host system user with which an SAP application server was started. This is usually the user `<SAPSID>adm` on UNIX, or `SAPService<SAPSID>` on Windows.

The user `<SAPSID>adm` or `SAPService<SAPSID>` must be:

- Specified as a trusted user `OPS$` in the Oracle database
- Granted the `SAPDBA` role

For more information, see the Oracle database documentation.

These prerequisites are normally met in the standard SAP System.

4. Configure BRBACKUP, BRARCHIVE, and BRCONNECT.

BRBACKUP, BRARCHIVE, and BRCONNECT 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`

#### **i** Note

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 .

# 3 The DBA Cockpit

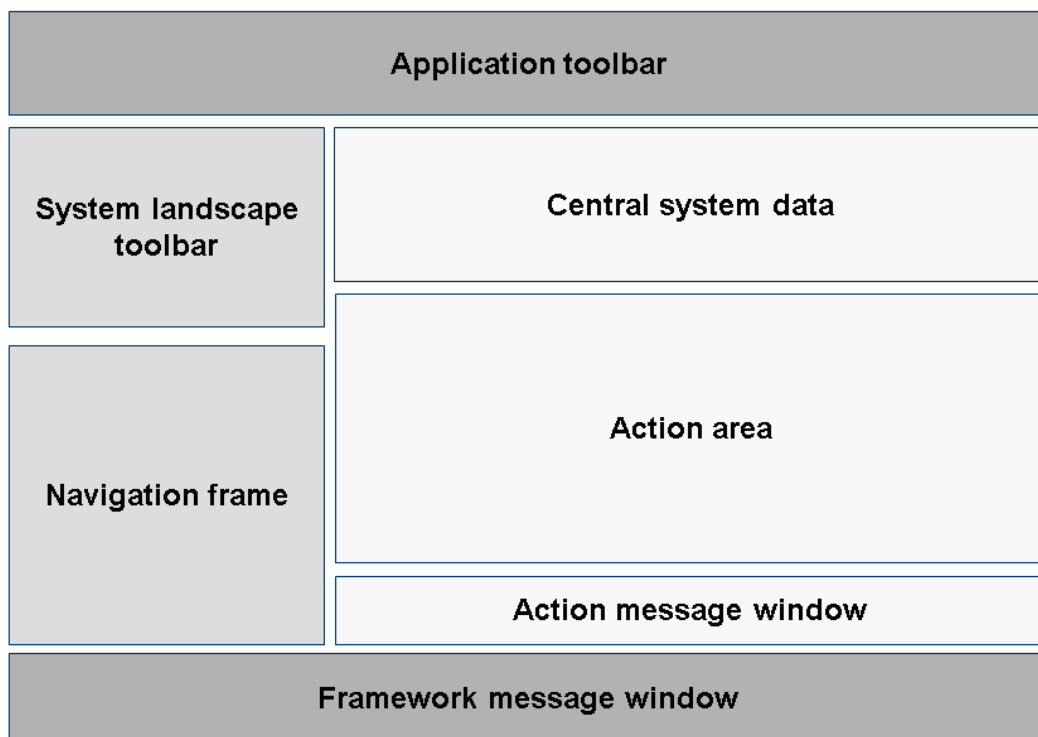
## Use

The DBA Cockpit is a platform-independent tool that you can use to monitor and administer your database. It provides a graphical user interface (GUI) for all actions and covers many aspects of handling a database system landscape.

You access the DBA Cockpit by calling transaction DBACOCKPIT. Then, the initial screen *DBA Cockpit: System Configuration Maintenance* appears.

## Features

The entry screen of the DBA Cockpit is divided into the following areas:



- **Application toolbar**  
Provides a minimum of functions, for example, to display or hide the areas on the left side.
- **System landscape toolbar**  
Provides central functions to manage the system landscape, for example:
  - Access to system configuration where you configure and set up your system landscape



- Management of database connections
- Lets you choose the system to monitor. Also, additional information about a distributed database system is displayed if available.
- Navigation frame  
Displays a tree structure divided at the top level into the main task areas of database administration. These are, for example, performance monitoring, space management, and job scheduling. Within each task area, there is a set of related action nodes.
- Central system data  
This area is common to most actions. It provides, for example, the following data:
  - Time of last refresh
  - Database startup time
  - Name of database
 It is not available for all database platforms. For more information, see the corresponding database-specific section.
- Action area  
Displays the details of the currently selected action.
- Action message window  
Only appears with certain actions and displays additional information that is related to the selected action.
- Framework message window  
Displays the message window provided by the framework. Unlike the classic SAP GUI messaging process, the framework message window contains a complete history of all messages sent during the session. In addition, you can:
  - Clean up the window by choosing *Delete*.
  - Collapse or expand the window by choosing *Minimize Message Window* or *Show Message Window*.
  - Check if a long text for a message is available by double-clicking the message or by choosing *Show Longtext*.

### **i** Note

Changes to the screen area sizes are user-specific and are restored when you next start the DBA Cockpit.

## Activities

You can use the DBA Cockpit to:

- Navigate between different actions
- Change to another action without closing the previous action and still hold all data retrieved by this action  
You can return to this action without data loss by double-clicking the previous action in the navigation frame or by choosing *Back*.
- Handle central configuration
- Monitor remote systems using remote database connections  
To use the functions offered for remote monitoring, you must configure the system you want to monitor. The local system is configured automatically when you start the DBA Cockpit for the first time. After having configured the connection and depending on the database, more actions are required to configure the database monitor and to set up database administration.

## Constraints

For systems that are monitored using a remote database connection, constraints depend on whether:

- The database release of the remote system is compatible to the database release of the local system.
- You want to monitor an ABAP-only or a Java-only SAP system.

## 3.1 Central System Data

### Definition

Most of the DBA Cockpit screens display central system data of the selected screen.

### Structure

The central system data displayed is as follows (not all fields are displayed on all screens):

Field	Description
<i>DB Name</i>	Database name
<i>DB Server</i>	Database server name
<i>DB Release</i>	Database release number
<i>Started</i>	Date and time when database was started

## 3.2 Maintenance Actions in the DBA Cockpit

The DBA Cockpit provides a set of actions to monitor and to maintain the database. To be able to perform these actions, the SAP user requires some additional authorizations.

The maintenance actions provided in the DBA Cockpit set locks to prevent parallel processing. All changes to the database are recorded in an audit log file.

### Authorization Check

When you start the DBA Cockpit or change to another system in the DBA Cockpit, an authorization check is performed.

## Granting of Database Permissions

To be able to access the database, the user used for remote monitoring must at least have sufficient authorizations.

- Local systems use the connect user for monitoring tasks. This user already has sufficient permissions. If more authorizations are required for administrative actions, a second connection using the database administration user is used.
- Systems monitored via remote database connections use the user specified for the database connections. This user must have sufficient authorizations.

## Locking of Actions

For each maintenance action that you have selected using the DBA Cockpit, a lock is set for the system being monitored. All locks are released when you exit the DBA Cockpit or when you change to another system.

## Auditing of Maintenance Actions

**Oracle only:** When you make changes that affect database objects such as database configuration parameters or tablespaces, an audit log is written. You can display this audit log in the DBA Cockpit.

### i Note

The audit log in the DBA Cockpit is not a certified tool for auditing.

## Related Information

[Displaying the Audit Log \[page 11\]](#)

## 3.3 Displaying the Audit Log

### Use

You can track changes to the database made from the DBA Cockpit and to the monitoring setup using the maintenance actions provided in the DBA Cockpit. Changes made from outside - for example, using native database commands - are not displayed here.

## Procedure

1. Call the DBA Cockpit.
2. In the navigation frame, choose *Audit*.  
The *DBA Cockpit: Audit Log* screen appears. The audit log consists of the following fields:

Field	Description
<i>Date</i>	Start date of the action
<i>Time</i>	Start time of the action
<i>System</i>	Target system on which the action was performed
<i>Action</i>	Type of action (name of the action in the DBA Cockpit)
<i>Command</i>	Type of command (for example, ADD, DELETE or EDIT)
<i>Object</i>	Name of the modified object (for example, database or tablespace name)
<i>User</i>	Name of the SAP user who performed the action
<i>From System</i>	System from where the action was performed

3. By default, the system displays all audit entries logged during the current week. If you want to display another week, double-click a day in the calendar.  
To display more than one week, you can change the field *Number of Days*.
4. To display the details of an action, select the corresponding action and choose *Details*.  
In the lower half of the screen, the SQL statements that have been executed are displayed.

## 3.4 Configuration of Systems for Remote Monitoring

To be able to use the DBA Cockpit to monitor remote systems, you have to configure those systems in the DBA Cockpit. You can configure your database system either using database information that is stored in the system landscape directory (SLD) for automatic generation and update of system entries or manually using database connections and system entries.

You access the screen *The DBA Cockpit: System Configuration Maintenance* by choosing *System Configuration* in the DBA Cockpit. A list of all monitored systems is displayed with an icon showing the configuration status of a system (activated or deactivated). You can change the configuration status of a system by clicking on the icon.

### i Note

In the event of severe errors, we recommend that you deactivate your system to prevent further problems. After you have investigated and corrected the error, you have to activate your system again.

Normally, when you start the DBA Cockpit, the local system is set as default system. To change this setting, select a system from the list and choose *Default System*.

#### **i Note**

This setting only applies to the user currently logged on to the system. It is not a system-wide setting.

You use one of the following methods to monitor a system remotely:

- Remote database connections  
This method uses additional connections. It is the main access method for monitoring and administration tasks and is mandatory. You can specify remote connections for any database and maintain the connections using the DBA Cockpit. For more information, see the section Maintenance Actions in the DBA Cockpit.
- RFC connection  
For this method you have to assign an RFC connection to your system. RFC connections are available for SAP ABAP systems only. You can use RFC connections:
  - As an optional access path for ABAP-related monitoring functions, for example, for the consistency check of the ABAP Dictionary. That is, the DBA Cockpit uses the RFC connection in parallel to the database connection for the same system.
  - As an optional access path for ABAP-related monitoring functions, for example, for the consistency check of the ABAP Dictionary. That is, the DBA Cockpit uses the RFC connection in parallel to the database connection for the same system.

#### **i Note**

You can only maintain RFC connections with transaction SM59, not with the DBA Cockpit.

See also:

## **3.4.1 Configuring Systems for Remote Monitoring Using Remote Database Connections**

### **Use**

You use this procedure to configure systems that you want to monitor using remote database connections.

#### **i Note**

Depending on the database platform of the selected system, some options might not be available. In this case you cannot enter any data in the corresponding fields.

### **Prerequisites**

- Your application server must have client software installed to access the remote database that you want to monitor.

- The system(s) you want to monitor must have a database release supported by your version of the DBA Cockpit.
- The user for the database connection must have sufficient database permissions. For more information, see [Maintenance Actions in the DBA Cockpit](#).

## Procedure

### Adding a System

1. Call the DBA Cockpit.  
The screen [DBA Cockpit: System Configuration Maintenance](#) appears. It displays a list of all systems available with a Configuration Status icon, which shows the current system status ([Activated](#) or [Deactivated](#)).

#### i Note

When you start the DBA Cockpit for the first time, the local system is automatically added to the list of all systems available. At least one system entry is displayed.

2. Choose Add.  
The screen [Configuration: System Administration - Add System Entry](#) appears.
3. Specify the connection data as follows:
  1. In the [System](#) field, enter the name of the system you want to monitor.

#### i Note

This name is a unique ID and does not need to be the SAP system ID. You can choose any name except the SAP system ID of the local system, which is reserved for the local system entry.

Except for the local system entry, [Remote Database](#) is already selected.

2. Select [Database Connection](#).  
Enter the name of the database connection. If the database connection does not yet exist, you are directed to the [DB Connections: Add Connection Entry](#) screen where you can specify all relevant data for the new connection. For more information, see
3. After you have saved your entries, you are redirected to the screen [System Administration Details](#).
4. If an additional RFC destination is used for special ABAP monitoring functions or if the connection is initially routed using an RFC connection (MS SQL Server only), select RFC Destination, too.
5. If required, enter the name of the RFC connection.
4. After the connection data has been completed, it is displayed on the [System Data](#) tab page. You can enter a description of your system on the [Administration Data](#) tab.
5. Save your changes.

### Changing the Connection Parameters of a System

1. Perform step 1 as described above under [Adding a System](#).
2. Select a system.
3. Choose [Edit](#).  
The screen [Configuration: System Administration - Change System Entry](#) appears.
4. Enter your changes in the corresponding fields.
5. Save your changes.

### Deleting a System Entry

1. Perform step 1 as described above under *Adding a System*.
2. Select a system.
3. Choose *Delete*.

### 3.4.1.1 Configuration of Database Connections

#### Purpose

This section describes how you set and maintain technical attributes for remote database connections. The DBA Cockpit uses these connections for administration and monitoring or for application programs that use secondary connections to external databases.

#### Process Flow

1. You call the DBA Cockpit and choose *DB Connections* in the system landscape toolbar. The screen *DBA Cockpit: Database Connections* appears displaying a list of all available database connection definitions grouped by database platform:

Column	Description
<i>Remote Database Connections</i>	Name of database connection
	<div style="border-left: 2px solid #0070C0; padding-left: 10px;"> <p><b>i Note</b></p> <p>This is a unique name that you can freely choose.</p> </div>
<i>DB Name</i>	Name of database
<i>DB Host</i>	Name of database host
<i>DB Schema</i>	Name of the database schema to be monitored
<i>User</i>	Name of the connection user
<i>Permanent</i>	Specifies whether the connect user must be permanently available
<i>Max. Connections</i>	Maximum allowed number of open connections
<i>Opt. Connections</i>	Optimal number of connections

By default, the database connections that are defined in the local system are displayed.

#### i Note

MS SQL Server only:

You can also display the database connections of another ABAP system via RFC by entering an RFC destination in the appropriate field.

### i Note

DB2 for IBM i only:

For more information about the privileges required for the connection user, see SAP Note [2031950](#) - IBM i: Authorities for Database Remote Connections.

2. You are able to perform one of the following tasks:
  - You add database connections.
  - You change an existing database connection.
  - You delete a database connection.
  - You test a database connection.

See also:

- 
- 
- 
- 

## 3.4.1.1.1 Adding a Database Connection

### Use

You can use this procedure to add a database connection in the DBA Cockpit.

### Procedure

1. Call the DBA Cockpit.
2. In the system landscape toolbar, choose *DB Connections*.
3. Choose *Add*.  
The screen *DB Connections - Add Connection* Entry appears.
4. In the *Connection Name* field, specify the name of the connection.

### i Note

This name is a unique ID that you can choose freely except for names that are reserved by SAP for generated connections.

For example, you cannot use administrator connections or connections that are used by systems from the system landscape directory (SLD).

5. Specify the database connection attributes as follows:
  1. In the *Database System* field, specify the name of the database platform.
  2. In the *Connection Maximum* field, enter an appropriate value.



This value limits the number of database connections that are currently held by the SAP system. The SAP system does not let you exceed this limit.

3. In the *Connection Optimum* field, enter an appropriate value.  
This value is a flexible limit that can be exceeded if required.
4. If you want the connection to be mandatory for the SAP system, choose *Permanent Connection*. This parameter defines the availability of the database connection.  
It is then handled like the local default connection. This means that, if this database connection is not available for a work process, the work process of the SAP system cannot run.

#### ⚠ Caution

Only set this parameter if this connection is absolutely required to run your SAP system.

5. In the *User Name* field, enter the name of the connect user. Make sure that you choose a user with the appropriate authorizations.  
For more information, see *Maintenance Actions in the DBA Cockpit*.
  6. In the *Password* field, enter a password for the connect user.
6. In the *Connection Parameters* table, specify the following additional database-specific attributes:

Attribute	Description
<i>TNS Name</i>	Name of Oracle Transparent Network Substrate (TNS)
<i>Character Set</i>	Name of character set
<i>National Character Set</i>	Name of national character set

#### i Note

To support upcoming new connection parameters and for support scenarios, you might have to add connection parameters in an unchecked raw format.

To enable this, choose *Guided Mode* <-> *Expert Mode* and switch to the expert mode. In expert mode, you can enter connection parameters as a string instead of using the guided mode.

However, we recommend that you do **not** use the expert mode.

7. To confirm your entries, choose *Add*.

## Result

As soon as the connection has been specified, the DBA Cockpit connects automatically to the newly added database system and displays data on the *System Data* tab page.

### 3.4.1.1.2 Changing a Database Connection

1. Call the DBA Cockpit.
2. In the system landscape toolbar, choose *Database Connections*.
3. Select a database connection entry and choose *Edit*.  
The screen *DB Connections - Change Connection Entry* appears.
4. Enter your changes in the appropriate fields as described in .
5. Save your changes.

### 3.4.1.1.3 Testing a Database Connection

#### Use

You test a database connection to make sure that, for example, you entered the correct user and password information as well as the correct technical connection data, such as host name.

#### Procedure

1. Call the DBA Cockpit.
2. In the system landscape toolbar, choose *Database Connections*.
3. Select a system and choose *Test*.  
The result is displayed in the action message window below.

### 3.4.1.1.4 Deleting a Database Connection

1. Call the DBA Cockpit.
2. In the system landscape toolbar, choose *DB Connections*.
3. Select a system and choose *Delete*.

#### Caution

If the selected database connection is still in use by a system that is registered in the DBA Cockpit, you cannot delete it.

## 3.4.2 Configuring Systems for Remote Monitoring Using the System Landscape Directory (SLD)

### Use

The system landscape directory (SLD) contains data from all database systems available in your system landscape. You can use this data to set up the system configuration in the DBA Cockpit instead of setting it up manually.

When you set up the DBA Cockpit for the first time, you use this procedure to import the appropriate data from the SLD. During production operation, you use the procedure to synchronize the data between the SLD and the DBA Cockpit periodically.

### Procedure

1. To import database connection data from the SLD, call the DBA Cockpit.
2. In the system landscape toolbar, choose *System Configuration*.  
The screen *The DBA Cockpit: System Configuration Maintenance* appears.
3. Choose *SLD System Import*.  
The *SLD System Import* screen appears. Depending on the system landscape, one or more of the following nodes are displayed:
  - *New Database Systems in the SLD*  
All database systems registered in the SLD that are so far unknown to the DBA Cockpit are displayed.
  - *Changed Systems from Earlier SLD Imports*  
All database systems for which the main data differs between the SLD and the DBA Cockpit are displayed.
  - *Systems No Longer Registered in the SLD*  
All systems that were originally imported from the SLD into the DBA Cockpit but that are no longer registered in the SLD are displayed.
  - *Systems Identical in the SLD and in the DBA Cockpit*  
All systems that are registered in the SLD and that are identical in the DBA Cockpit are displayed.
  - *Unsupported Database Systems in the SLD*  
All database systems that are registered in the SLD but not supported by the DBA Cockpit are displayed.

#### **i** Note

Each database system is described as follows:

<Name (System ID) of the Database System> on <Main Database Host>  
( <Database Platform> )

The actions allowed for each database system are displayed in the second column of the tree.

4. To import database system data, select the actions that you want to execute for the selected database systems and choose Import. By default, only the import of new database systems is selected. The selected actions are executed. A short message for each executed action is displayed in the message window below.

### i Note

Connection data that is retrieved from the SLD might not be complete for one of the following reasons:

- Depending on the data provided by a system to the SLD, some connection data can be incomplete.
- User or password data is generally not available via SLD.

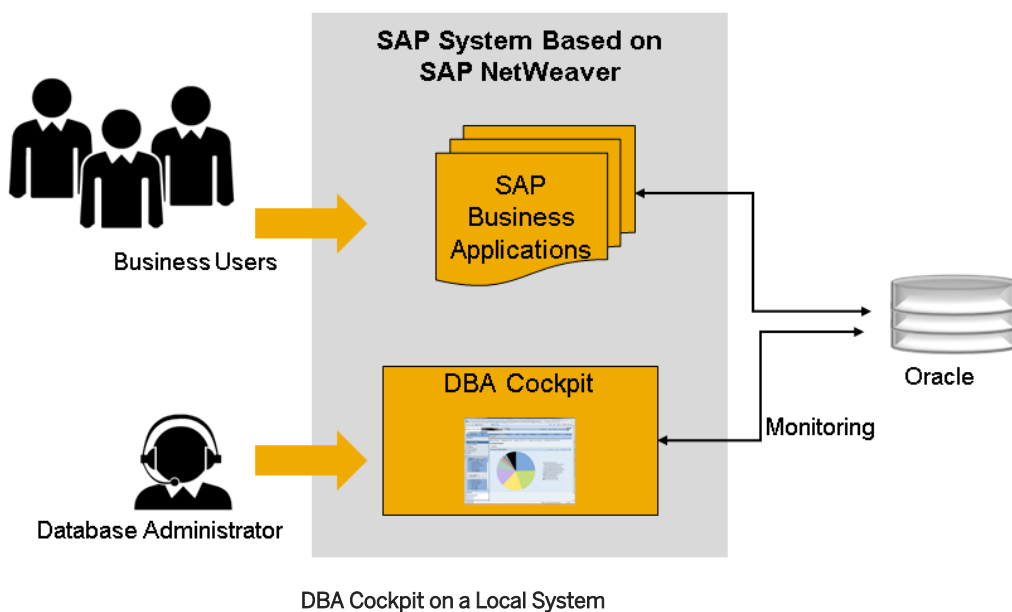
When you establish the connection to an imported system, the DBA Cockpit checks the completeness of a configured system. That is, if necessary, you are prompted for user, password and connection information.

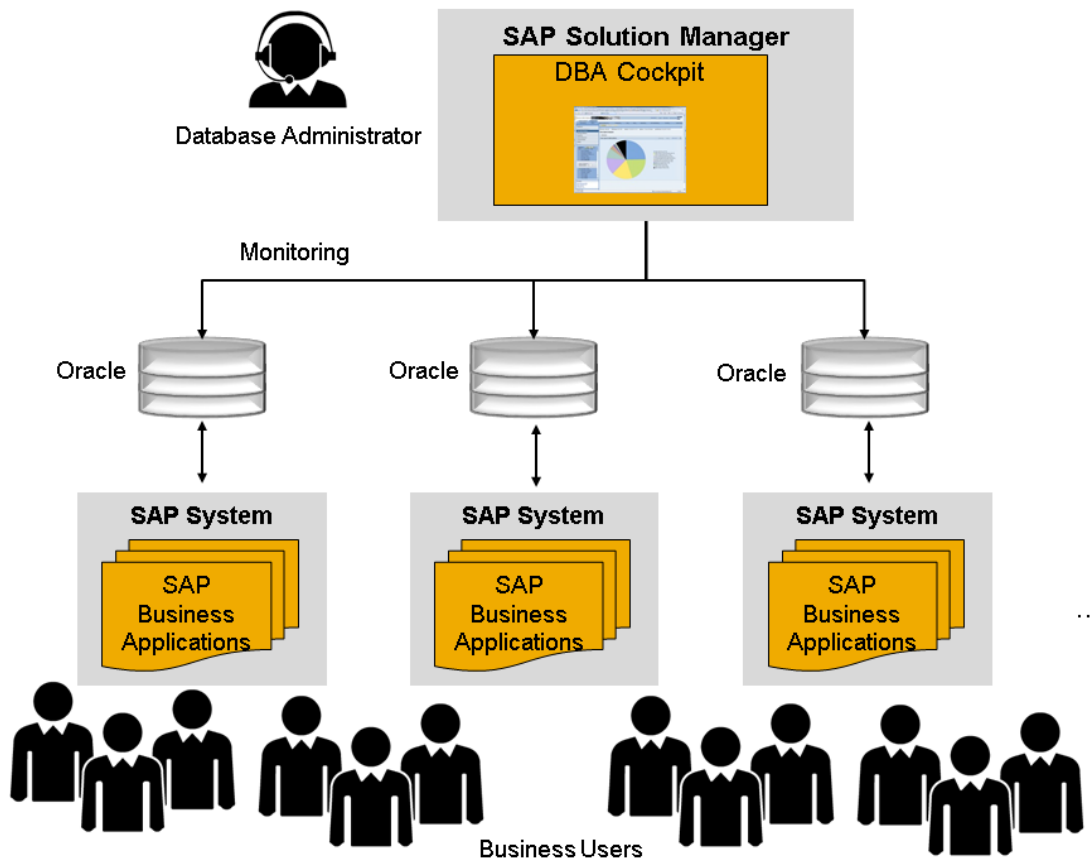
If additional connection information is required, enter the required data according to the maintenance dialog that is described in

## 3.5 The DBA Cockpit on a Local System and on SAP Solution Manager

### Use

The DBA Cockpit is part of every SAP NetWeaver-based system. You can run the DBA Cockpit locally on an SAP NetWeaver-based system by calling the DBACOCKPIT transaction. Alternatively, you can run the DBA Cockpit on your SAP Solution Manager system, where you can access all databases in your system landscape using remote connections. If you use the DBA Cockpit as part of the SAP Solution Manager system, this allows you to update and administrate all databases from a central system rather than logging on to each individual system separately.

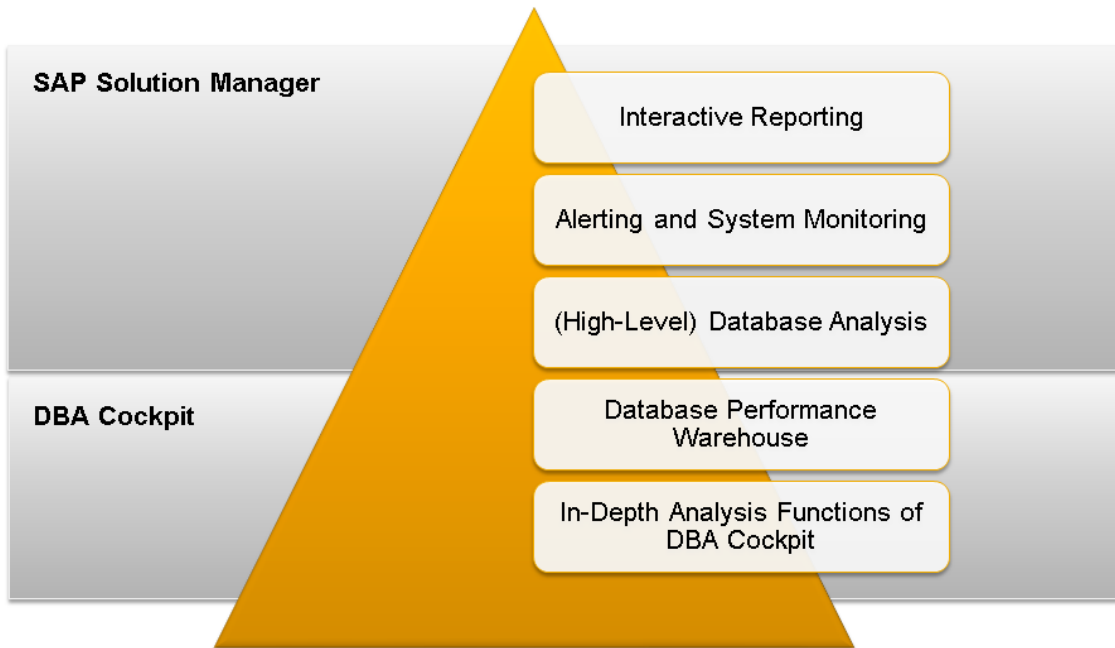




DBA Cockpit on an SAP Solution Manager System

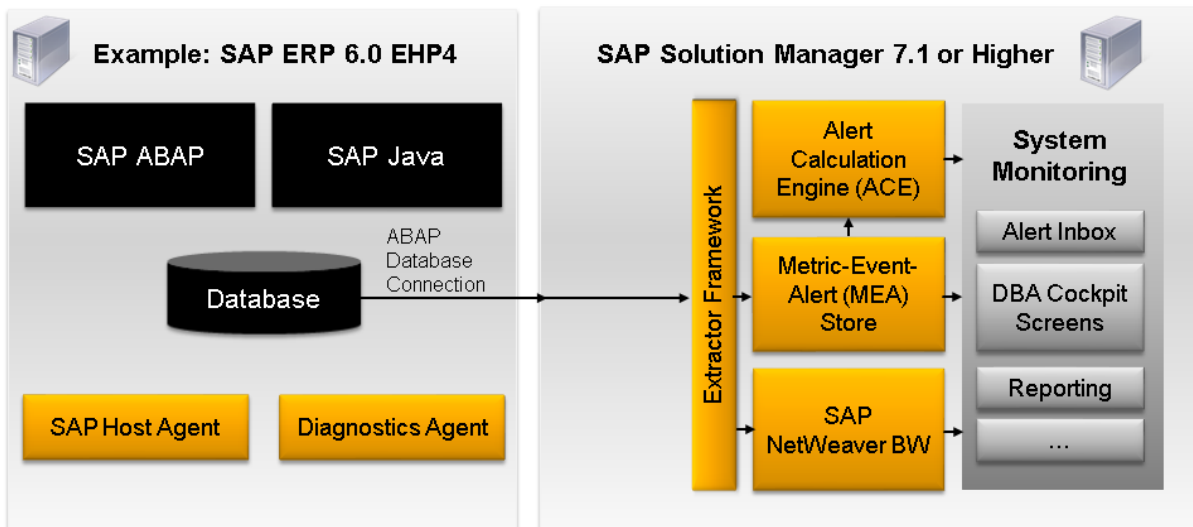
SAP Solution Manager is the free SAP toolset in your system landscape to monitor the full stack of an SAP system: from the operating system up to the business process. SAP Solution Manager is typically installed on a separate system, where it provides central access to tools, methods, and preconfigured contents that you can use during the evaluation, implementation, and operations of your systems. For database administrators, SAP Solution Manager offers a range of tools for root cause analysis, alerting, and reporting.

The tools of the DBA Cockpit complement the available SAP Solution Manager tools, which provide high-level overviews of possible database issues, including reporting and alerting functions. In addition to the high-level overviews of SAP Solution Manager, the DBA Cockpit serves as an in-depth analysis tool for database-related issues.



Available Database Monitoring in SAP Solution Manager and the DBA Cockpit

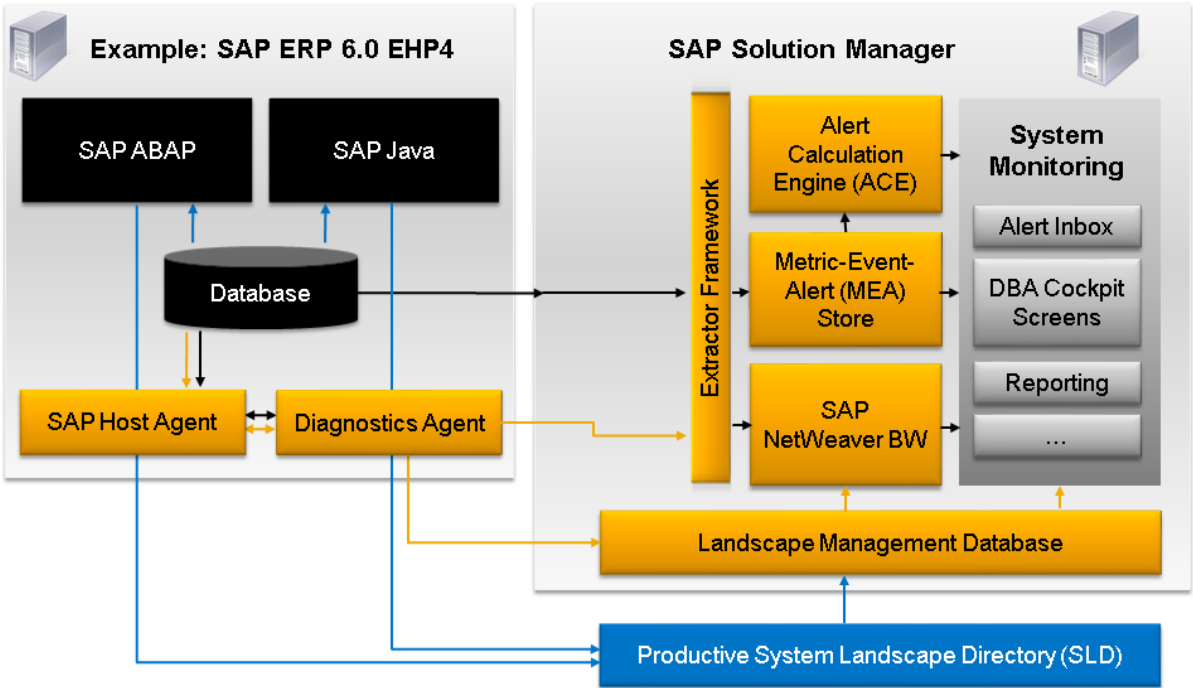
The DBA Cockpit not only provides more tools for the expert database administrator, but it also plays an important part in the SAP Solution Manager infrastructure. The remote database monitoring infrastructure of the DBA Cockpit is used by SAP Solution Manager to extract metrics from remote databases. These metrics are then passed on to the different applications in SAP Solution Manager.



### 3.5.1 Architecture Overview: End-to-End Monitoring and Alerting in SAP Solution Manager and DBA Cockpit

#### Use

The architecture of SAP Solution Manager comprises a number of tools and frameworks that collect detail data about databases and their statuses automatically. The tools and frameworks for automatic discovery of databases and database details include the landscape management database (LMDB), the Diagnostics Agent, and the extractor framework:



Data Collection and Database Monitoring and Alerting with SAP Solution Manager

#### Landscape Management Database (LMDB)

In SAP Solution Manager, all elements of a system landscape are modeled in the Landscape Management Database (LMDB). The core task of the LMDB is to provide information about the entire system landscape at a central location. The LMDB copies available data from the system landscape directory (SLD), but it enriches the data copied from SLD with additional information. For example, the SLD can provide the information which databases are available. For each database, the LMDB adds details such as the cluster topology of the database. All SAP Solution Manager applications are based on LMDB.

## Diagnostics Agent

The Solution Manager Diagnostics Agent (Diagnostics Agent), which is installed on every local system of your system landscape, gathers information from the managed systems and reports the information to the SAP Solution Manager system. This includes information about the availability of the database and its host.

The Diagnostics Agent uses SAP Host Agent to discover all databases installed on a host. The Diagnostics Agent and the SAP Host Agent transfer some basic information, such as the database host, the database type and its name, to the SLD. In addition, the Diagnostics Agent reports technical attributes like cluster topology, database release, and properties required for a remote database connection to the LMDB. This kind of information mapped in the LMDB with the data provided by the SLD. Information about databases that do not run with an SAP system are also pushed to the LMDB by the Diagnostics Agent.

## DBA Cockpit Backend, Extractor Framework, and Data Flow

The DBA Cockpit backend performs data collection via remote database connection and calculates the metrics of the monitored databases, such as performance KPIs, space, or buffer pool quality. The DBA Cockpit takes snapshots of database statuses, and thus is able to generate historical data (time series), which allows database administrators to keep track of short- and mid-term developments in the database. The data collected by the DBA Cockpit backend is also fed into the extractor framework, the central infrastructure for SAP Solution Manager for data collection and distribution.

The extracted data is reused by the following engines and stores:

- Alert Calculation Engine (ACE)
- Metric-Event-Alert (MEA) Store
- SAP BW for reporting

These engines and stores further process the raw data taken from the databases and generate alerts and metrics for BW reporting, which are then passed on to the different SAP Solution Manager applications, such as the alert inbox, reporting functions and also the BW reporting screens in the DBA Cockpit.

## Predefined Alerting Templates Shipped by SAP

During the SAP Solution Manager setup, you can find predefined monitoring and alerting templates that contain the definitions or values of metrics, events, and alerts (MEA). Depending on the defined threshold values, incidents and notifications are then automatically triggered. For IBM IBM Db2 databases, predefined metrics, events, and alerts are available for the following:

- Database availability
- Database exceptions, such as locks, missing backups, or error messages
- Database performance issues, such as bad buffer quality or problems with disk space usage

### **i** Note

The availability of the end-to-end alerting in SAP Solution Manager depends on your support license.



## 3.5.2 Setting Up Database Monitoring and Alerting in SAP Solution Manager

### Procedure

To set up the end-to-end monitoring and alerting infrastructure, you use the standard transactions for setting up SAP Solution Manager. You need to perform the following steps:

1. **Install the Diagnostics Agent on your database systems.**

The Diagnostics Agent gathers information from the managed systems and reports them to the SAP Solution Manager system. This also includes information about the availability of the database and its host. You need to install a Diagnostics Agent on each server (virtual hostname) that you want to monitor. You can use the SAP installation tool to install the Diagnostics Agent.

2. **Configure data suppliers in the system landscape directory (SLD).**

Use the transaction *System Landscape Directory: Local Administration* (RZ70) to configure SLD data suppliers in the system landscape. Here, you need to provide the port and the host of the SLD. The SLD data suppliers are programs that collect the database attributes *Database Host*, *Database Type* and *Database Name* at defined periods. Once the SLD data suppliers are up and running, they push database attributes from the database to the SLD.

3. **Connect the Diagnostics Agent to SAP Solution Manager.**

Use the transaction **SAP Solution Manager Configuration (SOLMAN\_SETUP) > System Preparation** to connect the SAP Diagnostics Agent to SAP Solution Manager.

4. **Configure the systems managed by SAP Solution Manager.**

Use the transaction **SAP Solution Manager Configuration (SOLMAN\_SETUP) > Managed System Configuration** to perform the following activities:

- Connect the SAP Diagnostics Agent to the managed databases
- Specify system parameters required to configure the managed databases  
Most of these parameters, such as hosts or ports, are automatically detected by the Diagnostics Agent, so only a password is required here.
- Trigger automatic configuration activities  
This includes the setup of database extractors, which collect data for monitoring the databases in SAP Solution Manager.

5. **Set up technical monitoring, including alerting.**

Use the transaction **SAP Solution Manager Configuration (SOLMAN\_SETUP) > Technical Monitoring** to set up the technical monitoring of databases in SAP Solution Manager. You set up the following:

- Activation or deactivation of auto-notifications (for example, e-mail notifications) about database alerts  
As a default, auto-notification is activated.
- Recipients and recipient lists of auto-notifications
- Assignment of monitoring templates to selected systems in scope for monitoring

Monitoring templates contain the definitions or values of metrics, events, and alerts (MEA) that trigger incidents and notifications. The SAP templates have predefined settings, but you can also adapt the templates to your customer-specific needs.

## More Information

- **Complete setup of SAP Solution Manager:** Documentation for SAP Solution Manager on SAP Help Portal at <http://help.sap.com/solutionmanager>
- **Additional database-specific setup steps:** SAP Note [1027146](#)
- **Overall architecture:** Architecture Overview: End-to-End Monitoring and Alerting in SAP Solution Manager and DBA Cockpit

## 3.5.3 Setting Up Remote Monitoring Manually

### Use

If you do **not** use SAP Solution Manager for monitoring, you can also set up the DBA Cockpit and databases for remote monitoring as follows:

- By manually creating database connections and system configuration entries in the DBA Cockpit  
This is relevant, for example, if you want to monitor a database using the DBA Cockpit and without configuring the database for monitoring in SAP Solution Manager. For more information, see [Configuring Systems for Remote Monitoring Manually](#).
- By using database information that is stored in the system landscape directory (SLD) for automatic generation and update of system entries in the DBA Cockpit  
This option is deprecated and only available in the SAP GUI version of the DBA Cockpit. Do **not** use this option if you use SAP Solution Manager. For more information, see [Configuring Systems for Remote Monitoring Using the System Landscape Directory](#).
- By configuring database connections on the [Database Connections](#) screen of the DBA Cockpit  
You can use the [Database Connections](#) screen to set up database connections that are used for non-monitoring components, for example, for the access to external data resources by SAP BW. For more information, see [Configuring Database Connections](#).

#### → Recommendation

We recommend that you use SAP Solution Manager for the setup and monitoring of your system landscape (see [Setting Up Database Monitoring and Alerting in SAP Solution Manager](#)).

## 3.6 Performance (Oracle)

### Use

You can use the [Performance](#) menu in the DBA Cockpit to [monitor your Oracle database \[page 27\]](#):

- [Performance overview \[page 28\]](#)
- [Wait event analysis \[page 38\]](#)

- [SQL statement analysis \[page 62\]](#)
- [Statistical information \[page 71\]](#)
- [Feature monitoring \[page 99\]](#)
- [Additional functions \[page 106\]](#)

## 3.6.1 SAP/Oracle Database Monitor

### Use

You can use the SAP/Oracle database monitor in the DBA Cockpit to monitor your database running with Oracle or Oracle Real Application Cluster (RAC). It is an **expert** tool.

#### i Note

The documentation in this section refers to the **new** SAP/Oracle Database Monitor.

For more information on the **old** SAP/Oracle Database Monitor, based on ST04N, see *SAP/Oracle Database Monitor (Old)*.

### Integration

The database monitor is part of the Computing Center Management System (CCMS). You can start it from the DBA Cockpit.

The monitor obtains information from the Oracle performance views (V\$, GV\$, and DBA-views).

### Prerequisites

You need to apply SAP Note [706927](#)  before using the database monitor.

### Features

The monitor:

- Gives a general overview of database performance
- Provides different ways of looking at the monitoring information:
  - A [main monitor \[page 28\]](#) with an overview of the database
  - Detailed analyses using sub-monitors, grouped as follows:
    - [Wait event analysis \[page 38\]](#)

- [SQL statement analysis \[page 62\]](#)
- [Statistical information \[page 71\]](#)
- [Feature monitoring \[page 99\]](#)
- [Additional functions \[page 106\]](#)
- Fully supports Oracle Real Application Cluster (RAC)
- Does **not** support Multiple Components in One Database (MCOB)

## Activities

1. Start the database monitor from the DBA Cockpit by choosing [Performance](#) [Performance Overview](#).

### i Note

The following steps are generally valid for all screens, including screens in the sub-monitors.

2. If your SAP system uses Oracle RAC, in *DB Instances* you double-click the required RAC instance or *Total* for all RAC instances.
3. To refresh the display, choose *Refresh*.

## 3.6.2 Performance Overview

### Definition

This is the main screen in the [SAP/Oracle Database Monitor \[page 27\]](#). It gives you an overview of the performance of your Oracle database.

### Use

You start the database monitor from the DBA Cockpit by choosing [Performance](#) [Performance Overview](#). You see an overview of database performance.

You can right-click a field and choose *Details* to see the values for each instance if you are running an Oracle Real Application Cluster (RAC).

If you are running RAC, you choose in *DB Instances* whether to display overview information for one RAC instance or for the total of all RAC instances.

### i Note

The appearance of a yellow or red light indicates that the difference in percent for the value of at least one instance from the average of all instances exceeds a certain limit.

## Structure

The fields are grouped as follows:

- *General information*, data source is V\$INSTANCE

Field	Description
<i>DB instance</i>	Name of the current database instance This is the SAP-SID in non-RAC environments.
<i>DB node</i>	Host name of the selected DB node This is the database server in non-RAC environments.
<i>DB release</i>	Release of the current database
<i>Day, time</i>	Current day and time
<i>Start up at</i>	Date and time when the current database instance started
<i>Time since start</i>	Time since start of the current database instance, in days-hours-minutes-seconds
<i>DB Time</i>	<Busy wait time> + <CPU time session>

- *Sizes*

Field	Description
<i>Database Buffer Size</i>	Size of the data buffer <ul style="list-style-type: none"><li>• Non-RAC or RAC detail: data buffer size of the current instance</li><li>• RAC total: total instance-related database buffers Data source: V\$SGA</li></ul>
<i>- Size default pool</i>	Size of the default buffer pool, calculated as follows: <pre>select value from v\$parameter where name = 'db_cache_size'</pre> Data source: gv\$sga_dynamic_components
<i>- Size keep pool</i>	Size of the keep buffer pool, calculated as follows: <pre>select value from v\$parameter where name = 'db_keep_cache_size'</pre> Data source: gv\$sga_dynamic_components

Field	Description
- <i>Size others</i>	<p>Sum of the sizes of all other buffer caches pools, calculated as follows:</p> <pre>select inst_id, round(sum(CURRENT_SIZE)/1024,0) othersize from gv\$sga_dynamic_components where component = 'DEFAULT 2K buffer cache' or component = 'DEFAULT 4K buffer cache' or component = 'DEFAULT 8K buffer cache' or component = 'DEFAULT 16K buffer cache' or component = 'DEFAULT 32K buffer cache' or component = 'RECYCLE buffer cache' group by inst_id</pre> <p>Can be displayed in detail as a pop-up</p> <p>Data source: gv\$sga_dynamic_components</p>
<i>In-Memory-Size</i>	Size available to in-memory data, configurable as of Oracle 12.
<i>Shared pool size</i>	<p>Size of the shared pool</p> <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: shared pool size of the current instance</li> <li>• RAC total: total shared pool size for all instances</li> </ul> <p>Data source: V\$SGA_DYNAMIC_COMPONENTS</p>
<i>Logical reads</i>	<p>Number of logical read operations</p> <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: logical reads of the current instance</li> <li>• RAC total: total logical read operations for all instances</li> </ul> <p>Data source: V\$SYSSTAT</p>
<i>Physical reads</i>	<p>Number of physical read operations</p> <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: physical reads of the current instance</li> <li>• RAC total: total physical read for all instances</li> </ul> <p>Data source: V\$SYSSTAT</p>

Field	Description
<i>Physical writes</i>	Number of physical write operations <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: physical writes of the current instance</li> <li>• RAC total: total physical write operations for all instances</li> </ul> Data source: V\$SYSSTAT
<i>Read by other sess</i>	Number of buffer busy wait situations <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: total waitstat counters of the current instance</li> <li>• RAC total: total waitstat counters for all instances</li> </ul> Data source: GV\$SYSTEM_EVENT
<i>Read by other sess time</i>	Sum of buffer busy wait times <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: sum of wait times for all wait counters of the current instance</li> <li>• RAC total: sum of wait times for all wait counters for all instances</li> </ul> Data source: V\$WAITSTAT

- *Quality*

Field	Description
<i>Data Buffer Quality</i>	Data buffer quality, calculated as follows:  $100\% - ((\text{physicalreads} - \text{physicalreads\_direct} - \text{physicalreads\_directlob}) / (\text{sess\_logicalreads} - \text{physicalreads\_direct} - \text{physicalreads\_directlob}))$ Non-RAC or RAC detail: data buffer quality of the current instance  RAC total: average of quality for all instance-related data buffer  Data source: V\$SYSSTAT

Field	Description
<i>DD-cache Quality</i>	<p>Data dictionary cache quality as percentage, calculated as follows:</p> $100\% - (\text{totalget\_misses} / \text{totalgets})$ <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: data buffer quality of the current instance</li> <li>• RAC total: average cache quality for all instances</li> </ul> <p>Data source: V\$ROWCACHE</p>
<i>SQL area getratio</i>	<p>Ratio of <b>gethits</b> to <b>gets</b> as a percentage, calculated as follows:</p> $\text{sum}(\text{gethits}) / \text{sum}(\text{gets}) \times 100$ <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: get ratio of the current instance</li> <li>• RAC total: average of all instance-related get ratios</li> </ul> <p>Data source: V\$LIBRARYCACHE</p>
<i>SQL area pinratio</i>	<p>Ratio of <b>pinhits</b> to <b>pins</b> as a percentage, calculated as follows:</p> $\text{sum}(\text{pinhits}) / \text{sum}(\text{pins}) \times 100$ <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: pin ratio of the current instance</li> <li>• RAC total: average of all instance-related pin ratios</li> </ul> <p>Data source: V\$LIBRARYCACHE</p>
<i>SQLA Reloads/pins</i>	<p>Ratio of <b>reloads</b> to <b>pins</b> as a percentage, calculated as follows:</p> $\text{sum}(\text{reloads}) / \text{sum}(\text{pins}) \times 100$ <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: reloads per pin ratio of the current instance</li> <li>• RAC total: average of all instance-related pin ratios</li> </ul> <p>Data source: V\$LIBRARYCACHE</p>

- *Log buffer*

Field	Description
<i>Size</i>	<p>Size of the redo log buffer</p> <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: redo log buffer size of the current instance</li> <li>• RAC total: total instance-related redo log buffer sizes</li> </ul> <p>Data source: V\$SGA</p>



Field	Description
<i>Entries</i>	<p>Number of redo log buffer entries</p> <ul style="list-style-type: none"> <li>Non-RAC or RAC detail: redo log buffer entries of the current instance</li> <li>RAC total: sum of all instance-related redo log buffer entries</li> </ul> <p>Data source: V\$SYSSTAT</p>
<i>Allocation retries</i>	<p>Number of redo buffer allocation retries</p> <ul style="list-style-type: none"> <li>Non-RAC or RAC detail: allocation retries of the current instance</li> <li>RAC total: total instance-related log buffer allocation retries</li> </ul> <p>Data source: V\$SYSSTAT</p>
<i>Redo log space wait</i>	<p>Redo log wait in seconds</p> <ul style="list-style-type: none"> <li>Non-RAC or RAC detail: Redo log wait time of the current instance</li> <li>RAC total: sum of instance-related redo log wait times</li> </ul> <p>Data source: V\$SYSSTAT</p>
<i>Redo log groups</i>	<p>Number of redo log groups</p>
<i>Log files [in use]</i>	<p>Number of active log files</p> <ul style="list-style-type: none"> <li>Non-RAC or RAC detail: active log files for the current instance</li> <li>RAC total: total active log files</li> </ul> <p>The figure in brackets refers to the number of active log files in use.</p> <p>Data source: V\$LOGFILE</p>

- Time Statistics*

Field	Description
<i>Busy wait time</i>	<p>Busy wait time, calculated as the sum of the time waited for all non-idle events.</p> <ul style="list-style-type: none"> <li>Non-RAC or RAC detail: busy wait time for the current instance</li> <li>RAC total: total instance-related busy wait times</li> </ul> <p>Data source: V\$SESSION_EVENT</p>

Field	Description
<i>CPU time session</i>	<p>CPU time session, calculated as sum of CPU time used by this session</p> <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: total CPU time for the current instance</li> <li>• RAC total: total CPU time for all instances</li> </ul> <p>Data source: V\$SYSSTAT</p>
<i>Time/User call</i>	<p>Time for each user call, calculated as follows: (busy wait time + CPU time) / user calls</p> <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: time for the current instance</li> <li>• RAC total: average time for all instances</li> </ul> <p>Data source: V\$SESSION_EVENT and V\$SYSSTAT</p>
<i>Sessions busy</i>	<p>Busy sessions as a percentage, calculated as follows: (busy wait time + CPU time) / total wait time</p> <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: percentage for the current instance</li> <li>• RAC total: average percentage for all instances</li> </ul> <p>Data source: V\$SESSION_EVENT and V\$SYSSTAT</p>
<i>CPU usage</i>	<p>CPU usage as a percentage, calculated as follows: CPU time / (elapsed time x CPU count)</p> <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: percentage for the current instance</li> <li>• RAC total: average percentage for all instances</li> </ul> <p>Data source: V\$SYSSTAT, V\$INSTANCE, and V\$PARAMETER</p>
<i>Number of CPUs</i>	<p>Number of CPUs</p> <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: CPUs for the current instance</li> <li>• RAC total: total CPUs for all instances</li> </ul> <p>Data source: V\$PARAMETER</p>

- *Calls*

Field	Description
<i>User calls</i>	<p>Number of user calls</p> <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: user calls for the current instance</li> <li>• RAC total: total user calls for all instances</li> </ul> <p>Data source: V\$SYSSTAT</p>

Field	Description
<i>User commits</i>	Number of user commits <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: user commits for the current instance</li> <li>• RAC total: total user commits for all instances</li> </ul> Data source: V\$SYSSTAT
<i>User rollbacks</i>	Number of user rollbacks <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: user rollbacks for the current instance</li> <li>• RAC total: total user rollbacks for all instances</li> </ul> Data source: V\$SYSSTAT
<i>Recursive calls</i>	Internal SQL statement or SQL statement in PL/SQL statement
<i>Parse count</i>	Total number of parse calls (hard and soft)
<i>User/recursive calls</i>	Ratio of user to recursive calls
<i>Log Reads/User calls</i>	Number of logical reads per user call.

- *Redo Logging*

Field	Description
<i>Redo writes</i>	Number of redo log writes <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: redo log writes for the current instance</li> <li>• RAC total: total redo log writes for all instances</li> </ul> Data source: V\$SYSSTAT
<i>OS blocks written</i>	Number of operating system redo blocks written <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: redo blocks written for the current instance</li> <li>• RAC total: sum of written redo blocks for all instances</li> </ul> Data source: V\$SYSSTAT
<i>Latching time</i>	Redo writer latching time in seconds Non-RAC or RAC detail: redo writer latching time for the current instance RAC total: total latching time for all instances Data source: V\$SYSSTAT

Field	Description
<i>Redo write time</i>	Redo write time in seconds <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: redo write time for the current instance</li> <li>• RAC total: total redo write time for all instances</li> </ul> Data source: V\$SYSSTAT
<i>Volume written</i>	Volume amount written <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: redo log data written for the current instance</li> <li>• RAC total: total redo log data written for all instances</li> </ul> Data source: V\$SYSSTAT

- *Table scans and fetches*

Field	Description
<i>Short table scans</i>	Number of short table scans <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: short table scans for the current instance</li> <li>• RAC total: total short table scans for all instances</li> </ul> Data source: V\$SYSSTAT
<i>Long table scans</i>	Number of long table scans <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: long table scans for the current instance</li> <li>• RAC total: total long table scans for all instances</li> </ul> Data source: V\$SYSSTAT
<i>Table fetch by rowid</i>	Number of table fetches by row ID <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: table fetches by row ID for the current instance</li> <li>• RAC total: total table fetches by row ID for all instances</li> </ul> Data source: V\$SYSSTAT
<i>Fetch by contin. row</i>	Number of fetches by continued row <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: table fetches by continued row for the current instance</li> <li>• RAC total: total table fetches by continued row for all instances</li> </ul> Data source: V\$SYSSTAT

- *Sorts*

Field	Description
<i>Sorts (memory)</i>	Number of sorts in memory <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: sorts in memory for the current instance</li> <li>• RAC total: total sorts in memory for all instances</li> </ul> Data source: V\$SYSSTAT
<i>Sorts (disk)</i>	Number of sorts in disk <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: sorts on disk for the current instance</li> <li>• RAC total: total sorts on disk for all instances</li> </ul> Data source: V\$SYSSTAT
<i>Sorts (rows)</i>	Number of sorted rows <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: sorted rows in memory for the current instance</li> <li>• RAC total: total rows for all instances</li> </ul> Data source: V\$SYSSTAT
<i>WA exec. optimal mode</i>	Number of work area executions in optimal mode <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: work area executions in optimal mode for the current instance</li> <li>• RAC total: total work area executions in optimal mode for all instances</li> </ul> Data source: V\$SYSSTAT
<i>WA exec. one pass mode</i>	Number of work area executions in one-pass mode <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: work area executions in one-pass mode for the current instance</li> <li>• RAC total: total work area executions in one-pass mode for all instances</li> </ul> Data source: V\$SYSSTAT
<i>WA exec. multipass mode</i>	Number of work area executions below the one-pass memory requirement <ul style="list-style-type: none"> <li>• Non-RAC or RAC detail: work area executions in multipass mode for the current instance</li> <li>• RAC total: total work area executions in multipass mode for all instances</li> </ul> Data source: V\$SYSSTAT

- [Top System Events](#)

This shows the system events in descending order of *Wait Time*.

## 3.6.3 Wait Event Analysis

### Use

These sub-monitors in the SAP/Oracle Database Monitor help you analyze SQL statements.

### 3.6.3.1 Session Monitor

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you check the Oracle session list and related resource information. In addition, you can see an execution plan and the SQL statement performed by a session. If a session performs an SQL statement from an ABAP program you can display the ABAP source.

### Use

You choose ► *Performance* ► *Wait Event Analysis* ► *Session Monitor* ► in the DBA Cockpit.

#### Structure

This screen contains the following information:

Column	Description
<i>SID</i>	Session ID
<i>ORA proc.</i>	Oracle shadow process ID
<i>SAP instance name</i>	SAP instance name
<i>Client system</i>	Client system
<i>Client proc.</i>	Client process ID
<i>Status</i>	Session status
<i>Event</i>	Event name
<i>SQL Text</i>	Text of SQL statement

Column	Description
<i>Logical reads</i>	Logical reads
<i>Phys. Reads</i>	Physical reads
<i>Block Changes</i>	Block changes
<i>Consistent Changes</i>	Consistent changes
Last wait time	Last wait time (>0 wait time,0 wait,-1 wait short,-2 unknown)
<i>Seconds in Wait</i>	Seconds in wait
<i>P1 Text</i>	Description of first parameter
<i>P1</i>	First additional parameter
<i>P2 Text</i>	Description of second parameter
<i>P2</i>	Second additional parameter
<i>P3 Text</i>	Description of third parameter
<i>P3</i>	Third additional parameter
<i>Block Gets</i>	Block gets
<i>Consistent Gets</i>	Consistent gets
<i>User</i>	Database user of the session
<i>PGA_USED_MEM</i>	PGA memory currently used
<i>PGA_ALLOC_MEM</i>	PGA memory currently allocated
<i>PGA_FREEABLE_MEM</i>	Allocated memory that can be freed
<i>PGA_MAX_MEM</i>	Maximum PGA memory ever allocated
<i>SAP_USER (Kernel 7.1)</i>	SAP logon user
<i>Client_Info (Kernel 7.1)</i>	Connection handle: Transaction: Main program

You can double-click a row to display the detail screen with the complete SQL statement. From the detail screen you can choose [Explain SQL statement](#) to display the execution plan.

You can choose the following:

- [Reset Point](#) to reset all fields in the result list from GV\$SESS\_IO to null. This means that you set a start point.
- [Since Reset](#) to display the results since the start point

## Integration

This monitor is based on the following views:

- GV\$SQL
- GV\$SQLTEXT

### 3.6.3.2 Buffer Busy Waits

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you check buffer busy waits in the Oracle database.

A buffer busy wait indicates that there are some buffers in the buffer cache that multiple processes are attempting to access concurrently. This event happens because one of the following is true:

- An Oracle block is being read into the buffer cache by another session and the session is waiting for that read to complete.
- The buffer is already in the buffer cache but in an incompatible mode (that is, some other session is changing the buffer).

#### Use

You choose ► [Performance](#) ► [Wait Event Analysis](#) ► [Buffer Busy Wait](#) ► in the DBA Cockpit.

#### Structure

##### i Note

Entries marked “RAC only” appear only for Oracle Real Application Cluster (RAC).

- [Buffer Busy Waits](#)

This tab page contains information on buffer busy waits:

Column	Description
<a href="#">Inst Id</a>	Database instance ID
RAC only	
<a href="#">Class</a>	Class of block



Column	Description
<i>Ttl waits</i>	Total number of waits due to this class of block
<i>Tm Wait (ms)</i>	Total of all wait times for all waits due to this class of blocks in milliseconds
<i>Avg Tm wait (ms)</i>	Average duration of wait due to this class of block in milliseconds
<i>%BBW/Inst</i>	Percentage of waits due to this class of block for each instance
<i>% of Time of BBW/Inst</i>	Percentage of time spent waiting due to this class of block for each instance
<i>%BBW</i>	Percentage of waits due to this class of block for all instances
<i>% of Time of BBW</i>	Percentage of time spent waiting due to this class of block for all instances

#### i Note

In single-instance - that is, non-RAC - environments, the following is true:

- *%BBW/Inst* shows the same value as *%BBW*
- *% of Time of BBW/Inst* shows the same value as *% of Time of BBW*

- RAC only: *Buffer Busy Waits with Total Lines*

This tab page shows the same information as in the table above plus Total lines for each Class of buffer busy wait. This helps you identify a buffer cache contention problem that is not caused by a specific instance.

### 3.6.3.3 System Event

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you check the following wait events and system events in the Oracle database:

- Busy waits summary
- Wait event details
- Oracle view GV\$SYSTEM\_EVENT

## Use

You choose [► Performance ► Wait Event Analysis ► System Event ▾](#) in the DBA Cockpit. Then you choose the required tab page [Busy Waits Summary](#), [Wait event details](#), or [GV\\$SYSTEM\\_EVENT](#).

You can choose the following:

- [Reset](#) to define a starting point for a measurement so that you can trace the measurement values over a time period
- [Since Reset](#) to display the measurement values for a time period between the current time and the time that you chose [Reset](#)
- [Since DB Start](#) to display all measurement values for the time period between the database start and the current time (default setting)

## Structure

### i Note

Entries marked “RAC only” appear only for Oracle Real Application Cluster (RAC)

- [Busy Waits Summary](#)  
This tab page displays a summary of busy waits:

Column	Description
<a href="#">Inst Id</a>	Database instance ID
RAC only	
<a href="#">Session type</a>	Type of session. For example, <a href="#">BACKGROUND</a> for system sessions, <a href="#">USER</a> for application sessions.
<a href="#">User Name</a>	Name of the user connected to the database. For example, SAP applications connect as user SAPR3.
<a href="#">PName</a>	Process name
<a href="#">Sessions</a>	Number of sessions connected to the database
<a href="#">Busy wait time (ms)</a>	Wait time spent busy in milliseconds
<a href="#">Total wait time (ms)</a>	Total time waiting for an event in milliseconds
<a href="#">Busy W (%)</a>	<a href="#">Busy wait time</a> as percentage of <a href="#">Total wait time</a>

- [Wait event details](#)

This tab page displays details of wait events:

Column	Description
<i>Inst ID</i>	Database instance ID
RAC only	
<i>Event</i>	Name of the event that caused the wait
<i>Wait time (ms)</i>	Time waiting for the event in milliseconds
<i>% of non-idle</i>	Percentage of non-idle waiting time caused by this event
<i>% of tot. resp.</i>	Percentage of total response time caused by this event
<i>Waits</i>	Number of waits
<i>Timeouts</i>	Number of timeouts
<i>Avg. WT (ms)</i>	Average wait time in milliseconds

- **GV\$SYSTEM\_EVENT**

This tab page displays details from the Oracle view GV\$SYSTEM\_EVENT:

Column	Description
<i>Event</i>	Name of the event that caused the wait
<i>Inst ID</i>	Database instance ID
RAC only	
<i>Wait time (ms)</i>	Time waiting for the event in milliseconds
<i>Wait% Inst/Evt.</i>	Percentage of time spent waiting for an event
<i>Waits</i>	Number of waits
<i>Timeouts</i>	Number of timeouts
<i>Avg. WT (ms)</i>	Average wait time in milliseconds

### **i** Note

This tab page shows events and wait times per instance in descending order of the event's total wait time.

In a RAC environment, you see by default the wait times for each instance and the total wait times for all instances. If required, you can restrict the display to a single instance.

## 3.6.3.4 System Event History

### Definition

This sub-monitor in the SAP/Oracle Database Monitor gives you an overview of the history of system events of the Oracle instance(s). It helps you to work out whether:

- The load has changed during the last few days
- The instances have load peaks
- One instance has more load compared to the other instances, in a RAC environment

### Use

You choose ► [Performance](#) ► [Wait Event Analysis](#) ► [System Event History](#) ► in the DBA Cockpit.

For more information about system events of a specific day, you can double-click a row in one of the two tab pages. This provides the data in hourly units for the selected day. You can double-click again to show all system events of the selected hour.

### Structure

- With idle events  
This tab page displays data for all system events including idle events.
- Without idle events  
This tab page excludes idle events.

Both tab pages display the following:

Column	Description
<i>Weekday</i>	Day of the week when the snapshot was created
<i>Date</i>	Date of the snapshot
<i>From</i>	Start of snapshot interval
<i>To</i>	End of snapshot interval
<i>Event</i>	Name of the wait event
<i>Time Waited (ms)</i>	Total amount of time waited for the event in milliseconds
<i>Total Waits in %</i>	Percentage of time waited compared to total value

Column	Description
<i>Average Wait (ms)</i>	Average amount of time waited for the event in milliseconds
<i>Total Timeouts</i>	Total number of timeouts for the event
<i>Total Waits</i>	Total number of waits for the event
<i>Time Waited (micro seconds)</i>	Total amount of time waited for the event in microseconds

### 3.6.3.5 Filesystem Requests

#### Definition

This submonitor in the SAP/Oracle Database Monitor lets you check filesystem requests in the Oracle database. It monitors the activity of filesystem requests with the Oracle GV\$FILESTAT view

#### Use

You choose ► [Performance](#) ► [Wait Event Analysis](#) ► [Filesystem requests](#) ► in the DBA Cockpit.

This monitor helps you to minimize the time needed to read or write data from or to a file, so that you can identify the frequently used data files and put them on separate disks to avoid contention, if necessary. Data file activity has an important effect on database performance.

#### Structure

##### i Note

Entries marked “RAC only” appear only for Oracle Real Application Cluster (RAC).

- [IO per File](#)

This tab page displays current statistics on physical file accesses per data file:

Column	Description
<i>File#</i>	File number
<i>Inst id</i>	Instance ID
RAC only	

Column	Description
<i>Full path</i>	Full file name including path
<i>Reads</i>	Number of reads
<i>Blk Reads</i>	Number of block reads
<i>Blk/Rd</i>	Number of blocks per read
<i>Rd Avg(ms)</i>	Read average time in milliseconds
<i>Rds/File(%)</i>	Percentage of reads per file
<i>Sgl Blk Rds</i>	Number of single block reads
<i>Sgl Blk Rds Avg(ms)</i>	Average time for single block reads in milliseconds
<i>Writes</i>	Number of writes
<i>Blk wrts</i>	Number of block writes
<i>Wrt Avg(ms)</i>	Average time for writes in milliseconds
<i>BBW</i>	Number of buffer busy waits
<i>Avg BBW(ms)</i>	Average buffer busy wait time in milliseconds
<i>BBW/File(%)</i>	Percentage of buffer busy waits per file

- *I/O per File With Total Lines* - RAC only  
This tab page displays the same information as in the table above plus *Total* lines for each *Full path*. This helps you identify a filesystem request problem that is not caused by a specific instance.
- *Total per Device*  
This tab page displays current statistics on total physical file accesses per disk device. There are also entries for each file on the device.

Column	Description
<i>File#</i>	File number
<i>Inst id</i>	Instance ID
RAC only	
<i>Name / Device</i>	Full file name including path
<i>Reads</i>	Number of reads
<i>Blk Rds</i>	Number of block reads
<i>Blk/Rd</i>	Number of blocks per read

Column	Description
<i>Rd Avg(ms)</i>	Read average time in milliseconds
<i>Rds/File(%)</i>	Percentage of reads per file
<i>% of Ttl Blk Rds</i>	Percentage of total block reads
<i>Sgl Blk Rds</i>	Number of single block reads
<i>Sgl Blk Rds Avg(ms)</i>	Average for single block reads in milliseconds
<i>Writes</i>	Number of writes
<i>Blk wrts</i>	Number of block writes
<i>Wrt Avg(ms)</i>	Average time for writes in milliseconds
<i>BBW</i>	Number of buffer busy waits
<i>Avg BBW(ms)</i>	Average buffer busy wait time in milliseconds
<i>BBW/File(%)</i>	Percentage of buffer busy waits per file

- *I/O per Path*

This tab page displays current statistics about total physical file accesses per path.

Column	Description
<i>Path</i>	File number
<i>Reads</i>	Number of reads
<i>Blk Rds</i>	Number of block reads
<i>Blk/Rd</i>	Number of blocks per read
<i>Rd Avg(ms)</i>	Read average time in milliseconds
<i>Rds/File(%)</i>	Percentage of reads per file
<i>% of Ttl Blk Rds</i>	Percentage of total block reads
<i>Sgl Blk Rds</i>	Number of single block reads
<i>Sgl Blk Rds Avg(ms)</i>	Average for single block reads in milliseconds
<i>Writes</i>	Number of writes
<i>Blk wrts</i>	Number of block writes
<i>Wrt Avg(ms)</i>	Average time for writes in milliseconds

Column	Description
<i>BBW</i>	Number of buffer busy waits
<i>Avg BBW(ms)</i>	Average buffer busy wait time in milliseconds
<i>BBW/File(%)</i>	Percentage of buffer busy waits per file

### 3.6.3.6 Enqueue Stat

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor helps you monitor enqueues and so reduce wait events.

#### Use

You choose ► [Performance](#) ► [Wait Event Analysis](#) ► [Enqueue Stat](#) ► in the DBA Cockpit.

#### i Note

Entries marked “RAC only” are only relevant for Oracle Real Application Cluster (RAC).

#### Structure

- *v\$enqueue stat*  
This tab page contains the following information:

Column	Description
<i>Instance Id</i> RAC only	Database instance ID
<i>Enqueue Type</i>	Type of enqueue requested
<i>Total Requests</i>	Total number of enqueue requests or enqueue conversions for this type of enqueue
<i>% Requests</i>	Requests for this enqueue type as a percentage of total requests



Column	Description
<i>Total Waits</i>	Total number of times an enqueue request or conversion resulted in a wait
<i>% Waits</i>	Waits for this enqueue type as a percentage of total waits
<i>Total Grants</i>	Number of times an enqueue request or conversion was granted
<i>% Grants</i>	Grants for this enqueue type as a percentage of total requests
<i>Total Fails</i>	Number of times an enqueue request or conversion failed
<i>% Fails</i>	Fails for this enqueue type as a percentage of total fails
<i>Cumulative Wait Time</i>	Cumulative (that is, total) amount of time in milliseconds spent waiting for the enqueue or enqueue conversion
<i>% Wait Time</i>	Wait time for this enqueue type as a percentage of total wait time
<i>% Wait / Uptime</i>	Cumulative wait time for this enqueue type as a percentage of the total database uptime

- Generating totals  
For numeric columns, you can select the column and choose *Total* to generate totals. For example, you can generate totals for the column *Total Requests*.
- Generating subtotals  
For the non-numeric columns you can select the column and choose *Subtotals* to generate subtotals. For example, you can generate subtotals for the column *Enqueue Type*.
- RAC only: *v\$enqueue stat with Total Lines*  
This tab page appears only for RAC systems when you are monitoring the **whole system**, that is, you have selected *Total* under *DB Instances*.  
The tab page displays the same information as in the table above plus total lines for all instances, marked with *Instance ID* set to zero.

## Integration

This monitor is based on the view GV\$ENQUEUE\_STAT.

## 3.6.3.7 Lock Monitor

### Definition

This sub-monitor in the SAP/Oracle Database Monitor helps you monitor currently active locks that are causing other requests to wait.

### Use

You choose ► [Performance](#) ► [Wait Event Analysis](#) ► [Lock Monitor](#) ► in the DBA Cockpit. You can display active locks in the following ways:

- Without DBA Objects

#### i Note

Entries marked “RAC only” are only relevant for Oracle Real Application Cluster (RAC).

### Structure

This screen contains the following information:

- [Without DBA Objects](#) - by omitting DBA objects, this has a performance advantage
- [With DBA Objects](#)
- [Blocking Graph](#) - uses an Oracle 10g technique

You can also view details on the locks displayed and also related locks:

- Detailed lock display  
You can double-click a row to view the detailed lock display, including the SQL statement.

#### i Note

In some cases, (G)V\$SESSION does not provide certain required and DBA\_OBJECTS also does not provide any object name. This means the monitor cannot evaluate the primary key values of the row involved, which means that nothing can be displayed.

- Related locks  
From the detailed lock display, you can choose [Linked Lock](#) to display the related lock holders or waiters:
  - For a lock holder, the lock holder itself and all lock waiters are displayed.
  - For a lock waiter, the lock waiter itself and the related lock holder are displayed.

## 3.6.3.8 Latch Monitor

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you view Oracle latch activity. A latch is a low-level serialization mechanism to protect shared data structures by preventing concurrent access to shared data structures in the Shared Global Area (SGA). Processes often have to wait to obtain a latch in order to access the data, which wastes CPU cycles.

### Use

You choose ► [Performance](#) ► [Wait Event Analysis](#) ► [Latch Monitor](#) ► in the DBA Cockpit.

### Structure

#### i Note

Entries marked “RAC only” are only relevant for Oracle Real Application Cluster (RAC).

- [Latch Overview](#)

You can use this tab page to identify the latches with the worst hit rates and the latches causing the most sleeps. There might be a problem if one of the library cache latches is causing the most sleeps.

This tab page displays the following information:

Column	Description
<a href="#">Name</a>	Latch name
<a href="#">Inst Id</a>	Instance ID
RAC only	
<a href="#">Wait time</a>	Elapsed time waiting for the latch in microseconds
<a href="#">% Wait time</a>	Wait time as a percentage of total wait time
<a href="#">Gets</a>	Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
<a href="#">Misses</a>	Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
<a href="#">Misses/Gets</a>	Ratio of <a href="#">Misses</a> to <a href="#">Gets</a>

Column	Description
<i>Sleeps</i>	Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
<i>% Sleeps</i>	Sleeps as a percentage of total sleeps
<i>Immediate Gets</i>	Number of times a latch was requested in no-wait mode
<i>Immediate Misses</i>	Number of times a no-wait latch request did not succeed (that is, missed)
<i>Spin Gets</i>	Willing-to-wait latch requests which missed the first try but succeeded while spinning
<i>Sleep 1</i>	Waits that slept once
<i>Sleep 2</i>	Waits that slept twice
<i>Sleep 3</i>	Waits that slept three times
<i>Sleep 4</i>	Waits that slept four times

To see the children of the selected latch, select a row and choose [Latch Children](#).

- [Latches Overview with Total Lines](#) - RAC only  
This tab page displays the same information as in the table above plus *Total* lines for each *Name*. This helps you identify a latch monitor problem that is not caused by a specific instance.
- [Latch Holder](#)  
This tab page shows details of the latch holders, based on the view GV\$LATCHHOLDER. It helps you to identify if the session holding the latch is changing and to check whether a latch is stuck on a particular session.  
This tab page displays the following information:

Column	Description
<i>Name</i>	Latch name
<i>Inst ID</i>	Instance ID
RAC only	
<i>SID</i>	ID of the session that owns the latch

- [Latch Children](#)  
This tab page shows the number of children for the latches shown

Column	Description
<i>Name</i>	Latch name

Column	Description
<i>Inst ID</i>	Instance ID
RAC only	
<i>Count</i>	Number of children

- [Latch Holders SQL Stmt](#)

This tab page shows the SQL statements that are currently being executed by the latch holders, based on the view GV\$LATCHHOLDER. Be sure to refresh the display frequently. To view the detailed SQL statement, choose [More Details](#).

Column	Description
<i>Name</i>	Latch name
<i>Inst ID</i>	Instance ID
RAC only	
<i>SID</i>	ID of the session that owns the latch

- [Cache Buffers Chains](#)

### **i** Note

Before you view, make sure that you have implemented SAP Note 159510 and use SAP\$BH instead of X\$BH.

This tab page shows cache buffer chains, based on the view GV\$LATCH\_CHILDREN. The default view is the top 200, ordered by wait time (descending) and sleeps (descending). You can use it to identify hot blocks (that is, frequently accessed blocks) in the buffer cache and also, in some cases, poorly tuned SQL statements.

Column	Description
<i>Name</i>	Latch name
<i>Inst ID</i>	Instance ID
RAC only	
<i>Address</i>	Address of the latch object
<i>Wait Tm(ms)</i>	Elapsed time waiting for the latch in microseconds
<i>% Wait Tm</i>	Wait time as a percentage of total wait time
<i>% Ttl Wait Tm</i>	Percentage of total wait time

Column	Description
<i>Gets</i>	Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
<i>% Gets</i>	Percentage of total gets
<i>Misses</i>	Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
<i>% Misses / Gets</i>	Misses as a percentage of gets
<i>Sleeps</i>	Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
<i>% Sleeps</i>	Sleeps as a percentage of total sleeps
<i>% Ttl Sleeps</i>	Percentage of total sleeps
<i>Imm Gets</i>	Number of times a latch was requested in no-wait mode
<i>Imm Misses</i>	Number of times a no-wait latch request did not succeed (that is, missed)
<i>% Imm Misses / Imm Gets</i>	Percentage of immediate misses to immediate gets
<i>Waits Holding Ltc</i>	Number of waits for the latch while the waiter was holding a different latch
<i>Spin Gets</i>	Willing-to-wait latch requests that missed the first try but succeeded while spinning
<i>% Sleeps / Gets</i>	Percentage of sleeps to gets

You can choose *All Cache Buffers Chains* to view all entries, not just the first 200.

You can choose *Hot Blocks* to view the most frequently accessed blocks in the buffer cache.

- *Latch Protected Stmts in Library Cache*

This tab page shows statements in the library cache that are protected by a latch. The library cache latch serializes access to the objects in the library cache. Every time an SQL statement, a PL/SQL block, or a stored object (that is, procedures, packages, functions, or triggers) is executed this latch is acquired.

- On tab page *SQL Stmts for Latches of Top-20 SQL Statements*, you can see the latches from the top 20 SQL statements:

Column	Description
<i>Name</i>	Latch name
<i>Inst ID</i>	Instance ID
RAC only	

Column	Description
<i>User</i>	User ID
<i>Executions</i>	Number of executions
<i>% Executions</i>	Percentage of executions
<i>Parse Calls</i>	Number of parse calls
<i>% Parse Calls</i>	Percentage of parse calls
<i>Parse Calls / Executions (%)</i>	Parse calls as a percentage of executions

- On tab page [SQL Stmts for Top Latch Protected SQL Stmts](#) you can see the latches from the top protected SQL statements:

Column	Description
<i>Child latch</i>	Latch name
<i>Inst ID</i>	Instance ID
RAC only	
<i>Address</i>	Address of the latch object
<i>Child latch</i>	Number of the child latch
<i>Hash value</i>	Hash value of the SQL statement
<i>User Name</i>	Name of the user connected to the database. For example, SAP applications connect as user SAPR3.
<i>Executions</i>	Number of executions that took place on this object since it was brought into the library cache
<i>Parse Calls</i>	Number of parse calls for this child cursor
<i>Parse calls / Exe</i>	Number of parse calls for each execution
<i>% Parse Calls</i>	Percentage of parse calls
<i>CPU Time(ms)</i>	CPU time in milliseconds
<i>CPU Time/Exe (ms)</i>	CPU time in milliseconds for each execution
<i>% CPU Time</i>	Percentage of CPU time
<i>Elapsed time</i>	Elapsed time in milliseconds
<i>Elapsed time/Exe</i>	Elapsed time in milliseconds for each execution

Column	Description
<i>% Elapsed Time</i>	Percentage of elapsed time
<i>Disk Reads</i>	Number of disk reads for this child cursor
<i>Disk reads/Exe</i>	Number of disk reads for each execution
<i>% Disk Reads</i>	Percentage of disk reads
<i>Buffer Gets</i>	Number of buffer gets for this child cursor
<i>Buffer Gets/Exe</i>	Number of buffer gets for each execution
<i>% Buffer Gets</i>	Percentage of buffer gets
<i>Rows processed</i>	Number of rows the parsed SQL statement returns
<i>Rows processed/Exe</i>	Number of rows processed for each execution
<i>Module</i>	Name of the module that was executing at the time that the SQL statement was first parsed
<i>SQL Stmt</i>	SQL statement for this child cursor

You can select a row and choose *Execution Plan of SQL Statement* to see the execution plan of the selected SQL statement.

You can select a row and choose *Call Point in ABAP Program* to see the ABAP call point of the selected SQL statement

### 3.6.3.9 Active Session History

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you view the Oracle active session history views, which provide sampled session activity in the database instances.

#### Use

You choose ► *Performance* ► *Wait Event Analysis* ► *Active Session History* ► in the DBA Cockpit.

Before the system displays the data, you have to specify the data range in the area *Preselect data pool*:

- *Data source*  
Select the view *gv\$active\_session\_history* to display current data or view *dba\_active\_session\_history* to display historical information.



- *From and To*  
Specify a time frame for the data that has to be displayed.

Use the following functions in *Preselect data pool* to complete the selection values:

- *Avail. timeframe* gives you information about available data in the two views. This function offers the option to take over this time frame in the selection fields.
- *Set to last hour* fills the time-frame selection with data from the last hour.

Finally, choose *Load data* to read the selected data.

## Structure

- ASH List  
This tab page displays the selected data in an ALV list, sorted by timestamp in descending order.

Column	Description
<i>Sample timestamp</i>	Time at which the sample was taken
<i>Session ID</i>	Session ID
<i>Sess State</i>	Session state
<i>Event Name</i>	Event name
<i>Sequence number of wait</i>	Sequence number to uniquely identify wait
<i>First additional parameter</i>	First additional parameter
<i>Parameter 2</i>	Second additional parameter
<i>Third additional parameter</i>	Third additional parameter
<i>Wait time [mics]</i>	Total wait time in microseconds for the event for which the session last waited if the session was on the CPU when sampled

Column	Description
<i>Time waited [mics]</i>	If the session state is waiting, this is the time in microseconds that the session actually spent waiting for the event. This column is set for waits that were in progress at the time the sample was taken. <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <p>If a wait event lasted for more than a second and was caught waiting in more than one session sample row, the actual time spent waiting for that wait event is populated in the last of those session sample rows. At any given time, this information is not available for the latest session sample.</p> </div>
<i>Dictionary object number of the object</i>	Object ID of the object that the session is referencing
<i>Object name</i>	Object name
<i>File number</i>	File number
<i>Block number</i>	ID of the block that the session is referencing  This information is only available if the session was waiting for cluster, concurrency, and user I/O wait events
<i>SQL_ID</i>	SQL identifier of the SQL statement that the session was executing at the time of sampling
<i>SQL String</i>	SQL string
<i>SQL Child #</i>	SQL child ID
<i>SQL statement hash</i>	SQL hash ID
<i>Session serial number</i>	Session serial number
<i>Oracle user identifier</i>	Oracle user ID
<i>OPCODE</i>	Indicates what phase of operation the SQL statement was in
<i>Service Hash</i>	Hash to identify service
<i>Sess. Type</i>	Session type
<i>Operating system Program name</i>	Name of the operating system program
<i>Name of the currently executing module</i>	Name of the currently executing module
<i>Currently executing action</i>	Name of the currently executing action

Column	Description
<i>Client ID</i>	Client ID of the session
<i>Blocking Session ID</i>	Session identifier of the blocking session  Populated only when the session was waiting for en-queues or a buffer busy wait.
<i>Bl. Sess. St.</i>	Status of the blocking session
<i>Blck Session Serial #</i>	Serial number of the blocking session
<i>QC Session ID</i>	Query coordinator session ID  This information is only available if the sampled session is a parallel query slave.
<i>QC Instance ID</i>	Query coordinator instance ID  This information is only available if the sampled session is a parallel query slave.
<i>Sample ID</i>	ID of the sample

- ASH Graphic  
This tab page displays the selected data in a time chart graphic. Here you have the option to change the graphics display interactively.
  - In the upper area there are the following features:

Feature	Description
<i>Key fig. (y axis)</i>	Key figure used as the value for the y axis.  You can choose between the number of <i>Active sessions</i> or the <i>Time waited</i> .
<i>Series type</i>	Characteristic represented by a line or series in the chart  You can choose between <i>Wait event</i> or <i>Wait event Class</i> .
<i>Select Series</i>	Filter for the wait events or wait event classes displayed in the chart
<i>TOP n</i>	Filter to restrict the display to the top n number of elements
<i>Time interval</i>	Time interval of the timestamps for summation of values in the chart  If you choose the interval size <i>millisec (sampled info)</i> the values in the y axis are the unchanged values from the ASH views. All other options are "condensed", which means that the values in the y axis are accumulated in intervals of, for example, 1 minute, 10 seconds, and so on.
<i>Refresh chart</i>	Refreshes the display immediately to reflect any changes made to the parameters above

Feature	Description
<a href="#">Save chart to frontend file</a>	Saves the chart to a front-end file You can choose a file type and a resolution in pixels.
<a href="#">Print graphic</a>	Prints the chart on your front-end printer
<a href="#">Graphic settings on / off</a>	Starts or ends a mode where you can change graphic settings For example, you can change the chart type, the dimensions of the chart, colors, and textures.
<a href="#">Load settings from frontend</a>	Defines graphic settings in front-end files
<a href="#">Save settings to frontend</a>	You can use this to transport settings between different systems.
<a href="#">Load settings in database</a>	Prompts you for an ID
<a href="#">Save settings in database</a>	The ID is used to process the current graphic settings by:
<a href="#">Delete settings in database</a>	<ul style="list-style-type: none"> <li>• Loading the settings from the database</li> <li>• Saving the settings in the database</li> <li>• Deleting the settings from the database</li> </ul>
	You can use this to share the same settings with other users.

- You have two more navigation functions that are not indicated explicitly on the interface:
  - Change the display from [Wait event classes](#) to all [Wait events](#) for a class  
Click on a wait event class in the chart's legend section (at the bottom of the screen) to display changes to all wait events belonging to the selected wait event class.
  - Detail information for a single point  
Click on a point in the chart (line or bar) to see more detailed information:
    - Tab [Wait events](#)  
Shows history data from the active session
    - Tab [File request](#)  
Shows file statistics information, which refers to the selected point. This means that the displayed files were accessed by sessions that met the wait event (or wait event class) within the time interval of the selected point.  
Note that the values of the key figures are based on the current system status, not on the selected time frame.  
Double-click a line to show accessed blocks for this file with timestamp.
    - Tab [SQL statement](#)  
Contains SQL statistics for statements referred to by the selected point. This means that the listed statements were processed by sessions that met the wait event (or wait event class) within the time interval of the selected point.  
Note that the values of the key figures are based on the current system status and not on the selected time frame.  
Double-click a line to show the timestamp of the sampled information. Choose [Explain plan](#) to display an [Execution Plan](#) for the selected SQL statement.

## 3.6.3.10 Workload Reporting

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you display very detailed reports from Oracle's Active Workload Repository.

For more information, see SAP Note [1028068](#). You can see the default values for workload reporting in SAP Note [1485598](#).

### Use

You choose **► Performance ► Wait Event Analysis ► Workload Reporting ►** in the DBA Cockpit.

Choose one of the following to get the report you require:

- *AWR report* (text)  
Workload report with text output
- *AWR report* (web)  
Workload report with web page display
- *AWR DIFF report* (text)  
Workload compare period report (difference report) with text output
- *AWR DIFF report* (web)  
Workload Compare period report (difference report) with web page display
- *SQL report* (text)  
Workload SQL report with text output
- *SQL report* (web)  
Workload SQL report with web page display
- *ASH report* (text)  
Active session history report with text output
- *ASH report* (web)  
Active session history report with web page display
- *ADDM report* (text)  
Automatic database diagnostic monitor (ADDM), only available as text output

#### i Note

For more information about the reports, see the Oracle documentation about the Active Workload Repository.

You need to enter the following parameters for the report:

- *Begin Snapshot* and *End Snapshot* for all reports
- *Begin Snapsh. 2* and *End Snapshot 2* for compare period reports only
- *SQL ID* for SQL reports
- *AvgBlockReadT*, "average block read time", in **milliseconds** for the ADDM report.

This is the average time to read a single database block. This value is used in the ADDM report for parameter `DBIO_EXPECTED`.

For more information about this parameter, see the Oracle documentation. Oracle uses a default value of 10 milliseconds. The current monitor proposes `gv$system_event-time_waited_mico / gv_system_event-total_waits` for event `db_file_sequential_read`.

## Structure

The monitor displays reports according to the format you selected:

- Text  
The display uses an editor with search and download functions.
- Web  
The HTML display uses a browser area with navigation functions and right-click functions similar to a normal web browser.

### 3.6.3.11 IO Redo Logs per Hour

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you display the number of IO redo log switches. For more information, see SAP note [1704439](#).

You choose ► *Performance* ► *Wait Event Analysis* ► *IO Redo Log Per Hour* ►

You can change the minimum and maximum time between log switches if required.

#### Structure

The number of IO redo log switches is displayed per hour by date, weekday, and time of day between the minimum and maximum time in seconds.

## 3.6.4 SQL Statement Analysis

### Use

These sub-monitors in the SAP/Oracle Database Monitor help you analyze SQL statements.

## 3.6.4.1 Shared Cursor Cache

### Definition

This function in the DBA Cockpit lets you check the shared cursor cache and related resource information. In addition, you can see an execution plan and the SQL statement. If the SQL statement is part of an ABAP program you can display the ABAP Source.

### Use

You choose ► [Performance](#) ► [SQL Statement Analysis](#) ► [Shared Cursor Cache](#) ► in the DBA Cockpit and specify the filter and sort criteria.

You can choose the following:

- [Explain](#) to display the execution plan
- [ABAP source](#) to show the source code of the ABAP program
- [Table selection](#) to restrict the display to SQL statements that use the specified table
- [Reset Point](#) to reset all fields in the result list for each SQL statement to null. This means that you set a start point
- [Since Reset](#) to display the results since the start point

You can also double-click a row to display a detail screen with the complete SQL statement. From the detail screen you can choose [Explain](#) to display the execution plan.

### Structure

This screen contains the following information:

Column	Description
<a href="#">Executions</a>	Total number of executions
<a href="#">Disk Reads</a>	Number of disk reads
<a href="#">Reads/Exec</a>	Number of disk reads per execution
<a href="#">Buffer Gets</a>	Number of buffer gets 999999999999 means overflow
<a href="#">Bgets/Exec.</a>	Number of processed rows
<a href="#">Proc. Rows</a>	Processed rows for execution

Column	Description
<i>Rproc/Exec</i>	Number of buffer gets per row
<i>Elapsed time</i>	Elapsed time in microseconds
<i>Elapsed Time/Exec</i>	Elapsed time per execution
<i>SQL Statement</i>	SQL statement
<i>CPU Time</i>	CPU time in microseconds
<i>CPU Time/Exec</i>	CPU time per execution
<i>Wait Time</i>	Wait time in microseconds
<i>Wait Time/Exec</i>	Wait time per execution
<i>SQL Sort</i>	Sort orders for statement
<i>Direct Writes</i>	Direct writes
<i>Open curs.</i>	Number of open cursor versions
<i>Users</i>	Number of users of cursor version
<i>Versions</i>	Number of loaded cursor versions
<i>Share mem</i>	Total shareable memory
<i>Pers. mem.</i>	Persistent memory
<i>Runtime mem.</i>	Total runtime memory
<i>Number of "invalidations"</i>	Number of invalidations
<i>Loads</i>	Number of loads or reloads
<i>Parse call</i>	Number of parse calls
<i>Program Name</i>	Name of calling program
<i>User</i>	User
<i>Curs.Ex</i>	Number of users executing this cursor
<i>Inst. ID</i>	Instance ID
<i>First Load time</i>	First cursor load time
<i>Last load time</i>	Last cursor load time
<i>Last active time</i>	Last active time of query plan



Column	Description
<i>Application wait time</i>	Application wait time in milliseconds
<i>Concurrency wait time</i>	Concurrency wait time in milliseconds
<i>Cluster wait time</i>	Cluster wait time in milliseconds
<i>User I/O wait time</i>	User I/O wait time in milliseconds

## 3.6.4.2 Table Access

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you view the shared cursor cache from the viewpoint of the tables accessed. This helps you to identify performance problems for a table rather than for a statement, such as a missing index on a table.

### Use

You choose ► [Performance](#) ► [SQL Statement Analysis](#) ► [Table Access](#) ► in the DBA Cockpit.

### Structure

#### i Note

Entries marked "RAC specific" are only relevant for Oracle Real Application Cluster (RAC). In a non-RAC database, the values in these columns are zero.

- [Summary on Table Level](#)

This screen displays the following information:

Column	Description
<i>Instance ID</i>	Instance ID
RAC only	
<i>Owner</i>	Table owner

Column	Description
<i>Table</i>	Table name
<i>Size (kB)</i>	Table size in kilobytes
<i>Type</i>	Type of object (v\$object dependency)
<i>Buffer pool</i>	Default buffer pool of the object
<i>Executions</i>	Total number of executions of all statements accessing this table
<i>Cache Hit Ratio</i>	Cache hits calculated with the following formula: $100 \times (1 - (\text{sum of disk reads} / \text{sum of buffer gets}))$
<i>Disk Reads</i>	Total number of disk reads executed on this table
<i>Disk Read Ratio</i>	Sum of (disk reads / physical reads)
<i>Buffer Gets</i>	Total number of buffer gets for the table
<i>Logical Read Ratio</i>	Sum of buffer gets / (sum of buffer gets + sum of disk reads)
<i>Rows Proc.</i>	Total number of rows returned or accessed by the statements on this table
<i>Rows/Exec</i>	Number of rows processed / number of executions
<i>Bufgets/Record</i>	Number of buffer gets / number of rows processed
<i>Sorts</i>	Total number of sorts done by the statements on this table
<i>CPU Time (ms)</i>	Total CPU time in milliseconds used by all statements on this table for parsing, executing, or fetching.
<i>Users Opening</i>	Total number of users executing the statement
<i>Opening Version</i>	Total number of statements related to this table having the child cursor locked
<i>Loaded Version</i>	Total number of statements related to this table having the context heap locked
<i># Childs</i>	Maximum number of the child cursor
<i>Sharable Mem.</i>	Total amount of shared memory used by the statement in bytes

<i>Persistent Mem.</i>	Sum of the fixed amount of memory used for the lifetime of this statement in bytes
<i>Runtime Mem.</i>	Sum of the fixed amount of memory required during the execution of this statement in bytes
<i># Invalidations</i>	Total number of times any cursor has been invalidated
<i>Parse Calls</i>	Total number of parse calls for all the statements
<i>ITL Waits</i>	Number of Interested Transaction List (ITL) waits for this table
<i>Buffer Busy Waits</i>	Number of buffer busy waits for this table
<i>DB Block Changes</i>	Number of database blocks changed for this table
<i>Global Cache CR Blocks Served</i> RAC specific	Number of global cache CR blocks served for this table
<i>Global Cache Current Blocks Served</i> RAC specific	Number of global cache current blocks served for this table
<i>Logical Reads</i>	Number of logical reads for this table
<i>Physical Reads</i>	Number of physical reads for this table
<i>Physical Reads Direct</i>	Number of direct physical reads for this table
<i>Physical Writes</i>	Number of physical writes for this table
<i>Physical Writes Direct</i>	Number of direct physical writes for this table
<i>Row Lock Waits</i>	Number of row lock waits for this table

- [Detail of Operations for <Table Name>](#)

You can double-click a row on the screen [Summary on Table Level](#) to see the details of operations for a table. The only differences from the previous screen are:

- The column [Operation](#) is new.
- The column [Table](#) no longer appears.

- [Details of Statements for <SQL Statement>](#)

You can double-click a row on the screen [Detail of Operations for <Table Name>](#) to display the detail screen with the complete SQL statement in the final column.

From this screen you can select a row and choose:

- [Execution Plan of SQL statement](#) to display the execution plan
- [Call Point in ABAP Program](#) to display the ABAP coding, positioned at the calling point of the parsed statement

## 3.6.4.3 Column Usage

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you check the contents of the system table SYS.COL\_USAGE\$. The table SYS.COL\_USAGE\$ lets you monitor the usage of predicates on columns (equal, like, and so on) in select statements. If `_column_tracking_level` is set to 1, smon updates the table at intervals, so it is not always up-to-date.

### Use

You choose ► *Performance* ► *SQL Statement Analysis* ► *Column Usage* ► in the DBA Cockpit.

After calling this sub-monitor, you can specify selection parameters:

- *Owner name*
- *Table name*
- *Column name*

Choose *Select* to start the query. You can use *Reset Selection* to clear the selection fields for a different query.

### Structure

The result set is a table:

Column	Description
<i>Owner</i>	Table owner
<i>Table name</i>	Table name
<i>Column name</i>	Column name
<i>Equ. Pred</i>	Number of equal predicats
<i>Equal Join</i>	Number of equal joins
<i>NonEquJoin</i>	Number of non-equal joins
<i>Range Pred</i>	Number of range predicats
<i>Like Pred</i>	Number of like predicats
<i>NULL Pred</i>	Number of null predicats

Column	Description
<i>Date</i>	Date of analysis

## Integration

Before you can use this sub-monitor, you must create the following synonyms:

- `sap_syscol` for the database object `SYS.COL$`
- `sap_sysobj` for the database object `SYS.OBJ$`
- `sap_sysusage` for the database object `SYS.COL_USAGE$`
- `sap_sysuser` for the database object `SYS.USER$`

See also [SAP Note 706927](#), which has an attached script to create all objects required to run the database monitor.

### 3.6.4.4 System Statistics for the CBO

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor provides an overview of I/O and CPU statistics for the Cost-Based Optimizer (CBO). This allows the optimizer to generate relevant costs for system-resource plans. For each plan, the CBO optimizer computes estimates for I/O and CPU costs. For more information, see [SAP Note 1159158](#).

The collected statistics are:

- Single block read time in ms
- Multi-block read time in ms
- CPU speed in MHz
- Average multi-block\_read\_count in number of blocks

The system statistics must be collected when the system has an average workload.

The statistics are gathered with the PL/SQL `DBMS_STATS` package:

- `DBMS_STATS.CREATE_STAT_TABLE` - create a user table to collect the statistics
- `DBMS_STATS.GATHER_SYSTEM_STATS` - collect statistics for a special time frame
- `DBMS_STATS.IMPORT_SYSTEM_STATS` - transfer the data from the user table to the dictionary tables
- `DBMS_STATS.DELETE_SYSTEM_STATS` - delete any existing system statistics from the dictionary

## Use

You choose [► Performance](#) [► SQL Statement Analysis](#) [► System Statistics for CBO](#) [►](#) in the DBA Cockpit.

The sub-monitor only displays information if one of the following is true:

- System statistics are collected for the system
- System statistics are activated by import into SYS.AUXSTAT\$

## Structure

- [DDIC Stats](#)

The displayed data is the result of the following SQL query:

```
select table_name, last_analyzed from dba_tables
where owner = 'SYS'
order by LAST_ANALYZED asc
```

- [Fixed object stats](#)

The monitor displays for which "fixed objects" a statistic exists. "Fixed Objects" are, for example, the performance views. The displayed data is the result of the following SQL query:

```
select table_name, last_analyzed from dba_tab_statistics
where owner = 'SYS' and table_name LIKE 'X%'
order by last_analyzed asc
```

- [System Statistics](#)

This uses the evaluation with DBMS\_STATS.GET\_SYSTEM\_STATS to display system statistics that currently exist.

- [Saved Statistics](#)

This displays data that was created using the procedure DBMS\_STATS.CREATE\_STAT\_TABLE for tables.

- [Locked/Missing Table Statistics](#)

This lists tables that have locked statistics or tables for which the IGNORE indicator is set in the table DBSTATC. In addition, you can recognize tables that have missing statistics.

For information about the SQL query used, see SAP Note [1159158](#).

## 3.6.4.5 Oracle Tuning Advisor

### Definition

For more information on how to enable the Oracle tuning advisor, see SAP Note [1136582](#). You must have purchased the Oracle Tuning Pack to use this functionality.

## 3.6.5 Statistical Information

### Use

These sub-monitors in the SAP/Oracle Database Monitor show statistical information.

### 3.6.5.1 SGA Monitor

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you monitor the System Global Area (SGA).

#### Use

You choose ► *Performance* ► *Statistical Information* ► *SGA Monitor* ► in the DBA Cockpit.

#### Structure

##### i Note

Entries marked “RAC only” are only relevant for Oracle Real Application Cluster (RAC).

- *SGA*  
This screen provides basic information about the SGA components.  
The Oracle views *GV\$SGA* and *GV\$SGA\_DYNAMIC\_FREE\_MEMORY* supplies the information displayed.  
If there are multiple instances, total rows are displayed.

Column	Description
<i>Comp .grp.</i>	SGA component group
<i>Inst Id</i>	Instance ID
RAC only	
<i>Instance Name</i>	Instance name
RAC only	
<i>Mem. size</i>	Memory size in bytes

- [SGA \(detail\)](#)

This tab page provides detailed information about the SGA components.  
The Oracle view GV\$SGASTAT supplies the information displayed.

Column	Description
<a href="#">Pool</a>	Pool in which the memory of this <i>SGA component name</i> resides
<a href="#">SGA component name</a>	SGA component name
<a href="#">Inst Id</a> RAC only	Instance ID
<a href="#">Instance Name</a> RAC only	Instance name
<a href="#">Mem. size</a>	Memory size in bytes

- [Curr. SGA resize op.](#)

This tab page provides information about current resize operations on the SGA.  
The Oracle view GV\$SGA\_CURRENT\_RESIZE\_OPS supplies the information displayed.

- [Full](#)

This view groups the information by component and shows totals for all instances.

Column	Description
<a href="#">Component name</a>	SGA component name
<a href="#">Inst ...</a> RAC only	Instance ID
<a href="#">Instance Name</a> RAC only	Instance name
<a href="#">Op. type</a>	Type of operation, grow or shrink
<a href="#">Op. mode</a>	Mode of operation, manual or automatic
<a href="#">Parameter for the resize op.</a>	Parameter used in the resize operation
<a href="#">Parameter value at the start</a>	Parameter value at the start of the operation
<a href="#">Desired param. value</a>	Desired parameter value after the resize
<a href="#">Curr. value</a>	Current value of the parameter
<a href="#">Start time of the operation</a>	Start time of the operation



Column	Description
<i>Start date</i>	Start date of the operation
<i>Upd. time</i>	Last time progress was made for the operation
<i>Upd. date</i>	Last date progress was made for the operation

- *Sort by Component*  
This tab page sorts the information by component. There are no instance totals.  
This tab page displays the same information as the table above.
- *Dyn. SGA Components*  
This tab page displays information about dynamic SGA components. It summarizes information from all completed SGA resize operations since instance startup. All sizes are in bytes.  
The Oracle view GV\$SGA\_DYNAMIC\_COMPONENTS supplies the information displayed.

- *Full*  
This view groups the information by component and shows totals for all instances.

Column	Description
<i>Component</i>	SGA component name
<i>Inst ...</i>	Instance ID
RAC only	
<i>Instance Name</i>	Instance name
RAC only	
<i>Curr. value</i>	Current size of the component
<i>Min. size</i>	Minimum size of the component since instance startup
<i>Max. size</i>	Maximum size of the component since instance startup
<i>Operations</i>	Number of operations since instance startup
<i>Last op.</i>	Last completed operation for the component
<i>Last mode</i>	Mode of last completed operation for the component
<i>Start time</i>	Start time of the last completed operation for the component
<i>Op. Date</i>	Start date of the last completed operation for the component
<i>Granul.</i>	Granularity of the last completed operation for the component

- *Sort by Component*  
This tab page sorts the information by component. There are no instance totals.  
This tab page displays the same information as the table above.
- *Comp. SGA Resize op.*  
This tab page displays information about the last 100 completed SGA resize operations (excluding in-progress operations). All sizes are in bytes.  
The Oracle view GV\$SGA\_RESIZE\_OPS supplies the information displayed.

- *Full*  
This view groups the information by component and shows totals for all instances.

Column	Description
<i>Component name</i>	SGA component name
<i>Inst ...</i> RAC only	Instance ID
<i>Instance Name</i> RAC only	Instance name
<i>Op. type</i>	Type of operation, grow or shrink
<i>Op. mode</i>	Mode of operation, manual or automatic
<i>Parameter for the resize op.</i>	Parameter used in the resize operation
<i>Parameter value at the start</i>	Parameter value at the start of the operation
<i>Desired param. value</i>	Desired parameter value after the resize
<i>Real. value after resize</i>	Real parameter value after the resize
<i>Status</i>	Completion status of the operation: normal, cancel, or error
<i>Start Time</i>	Start time of the operation
<i>End Time</i>	End time of the operation

- *Sort by Component*  
This tab page sorts the information by component. There are no instance totals.  
This tab page displays the same information as the table above.
- *Cache Advisory stat.*  
This tab page displays information to predict the number of physical reads for the cache size corresponding to each row.  
The Oracle view GV\$DB\_CACHE\_ADVICE supplies the information displayed.

- *Full*

This tab page groups the information by component and shows totals for all instances. It only shows size factor one.

Column	Description
<i>Pool ID</i>	Buffer pool identifier
<i>Pool name</i>	Buffer pool name
<i>Inst ...</i>	Instance ID
RAC only	
<i>Instance Name</i>	Instance name
RAC only	
<i>Block Size</i>	Block size in bytes for buffers in this pool
<i>Status</i>	Status of the advisory: <ul style="list-style-type: none"> <li>• ON - currently running</li> <li>• OFF - disabled</li> </ul>
<i>Cache (MB)</i>	Cache size for prediction in MB
<i>SizeFactor</i>	Physical read factor for this cache size This is the ratio of the number of estimated physical reads to the number of reads in the real cache.
<i>CacheSize</i>	Cache size for prediction in buffers
<i>Phys.R. F.</i>	Physical read factor. This is the ratio of the number of estimated physical reads to the number of reads in the real cache.
<i>PhysReads</i>	Estimated number of physical reads for this cache size

- *Size for estimation*  
This tab page displays the same information as the table above for all size factors.
- *Shared pool advice*  
This tab page displays information to predict the number of physical reads for the cache size corresponding to each row.  
The Oracle view GV\$SHARED\_POOL\_ADVICE supplies the information displayed.
- *Full*  
This tab page groups the information by component and shows totals for all instances. It only shows size factor one.

Column	Description
<i>InstID...</i>	Instance ID
RAC only	
<i>Instance Name</i>	Instance name
RAC only	
<i>Size (MB)</i>	Shared pool size for estimate
<i>Sizefactor</i>	Size factor with respect to the current shared pool size
<i>lib. cache</i>	Estimated memory in use by the library cache
<i>Mem. Obj.</i>	Number of library cache memory objects
<i>Parse time</i>	Estimated elapsed parse time saved (in seconds) because library cache memory objects are found in a shared pool of the specified size
<i>ParseFact</i>	Estimated parse time saved factor with respect to the current shared pool size
<i>CacheHits</i>	Estimated number of times a library cache memory object was found in a shared pool of the specified size

- *Size for estimation*  
This tab page displays the same information as the table above for all size factors.
- *Buffer pool statistic*  
The tab page displays information about all buffer pools available for the instance. The “sets” are the LRU latch sets.  
The Oracle view GV\$BUFFER\_POOL\_STATISTICS supplies the information displayed.

Column	Description
<i>Inst ...</i>	Instance ID
RAC only	
<i>Instance Name</i>	Instance name
RAC only	
<i>Pool ID</i>	Buffer pool identifier
<i>Name</i>	Buffer pool name
<i>Block Size</i>	Block size in bytes
<i>Set MSi...</i>	Buffer Pool Maximum Set Size

Column	Description
<i>Repl.Lst</i>	Number of buffers on replacement list
<i>WriteList</i>	Number of buffers on write list
<i>B.In Set</i>	Number of buffers in set
<i>Got By Set</i>	Number of buffers gotten by the set
<i>Written</i>	Number of buffers written by the set
<i>Scanned</i>	Number of buffers scanned in the set
<i>Free wait</i>	Free buffer wait statistic
<i>WriteCompl</i>	Write complete wait statistic
<i>BusyWait</i>	Buffer busy wait statistic
<i>FreeInsp.</i>	Free buffer inspected statistic
<i>DirtyBuff.</i>	Dirty buffers inspected statistic
<i>BlkChange</i>	Database blocks changed statistic
<i>BlkGets</i>	Database blocks gotten statistic
<i>Cons.Gets</i>	Consistent gets statistic
<i>PhysReads</i>	Physical reads statistic
<i>PhysWrites</i>	Physical writes statistic

## 3.6.5.2 PGA Monitor

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you monitor the Program Global Area (PGA).

### Use

You choose ► [Performance](#) ► [Statistical Information](#) ► [PGA Monitor](#) ► in the DBA Cockpit.

## Structure

### i Note

Entries marked RAC only are only relevant for Oracle Real Application Cluster (RAC).

- *Status*

- *PGA Status*

This tab page shows the following information about the PGA configuration based on the view GV\$PGASTAT:

Column	Description
<i>Name</i>	Name of the statistic
<i>Inst Id</i> RAC only	Instance ID
<i>Inst Name</i> RAC only	Instance name
<i>Statistic value</i>	Statistic value
<i>Unit</i>	Statistic unit, such as <i>bytes</i> .

- *SQL Workarea*

- *View SQL WORKAREA*

This tab page shows the following information about the PGA configuration based on the view GV\$SQL\_WORKAREA:

Column	Description
<i>Inst Id</i> RAC only	Instance ID
<i>Inst Name</i> RAC only	Instance name
<i>Workarea Address</i>	Address of the parent cursor handle
<i>Parent address</i>	Address of the work area handle. This is the "primary key" for the view.
<i>Hash Value</i>	Hash value of the parent statement in the library cache

Column	Description
<i>Chil...</i> - <i>Number of this child cursor</i>	Number of the child cursor that uses this work area
<i>Operation Type</i>	Type of operation using the work area ( <b> SORT , HASH JOIN, GROUP BY, BUFFERING, BITMAP MERGE, or BITMAP CREATE</b> )
<i>Oper. ...</i> - <i>Operation ID</i>	Unique number used to identify the operation in the execution plan
<i>Sizi</i> - <i>Sizing</i>	Sizing policy for this work area ( <b> MANUAL or AUTO</b> )
<i>EstOptSiz</i> - <i>Estimated Optimal Size</i>	Estimated size (in KB) required by this work area to execute the operation completely in memory (optimal execution).
<i>Est IpSiz</i> - <i>Estimated Onepass Size</i>	Estimated size in KB required by this work area to execute the operation in a single pass
<i>Last Mem</i> - <i>Memory Used for Last Execution</i>	Memory in KB used by this work area during the last execution of the cursor
<i>Last Exec</i> - <i>Last Execution</i>	Indicates whether this work area runs using <b> OPTIMAL, ONE PASS, or ONE PASS </b> memory requirement (or <b> MULTI - PASS</b> ), during the last execution of the cursor
<i>Last De...</i> - <i>Degree of parallelism of last exec</i>	Degree of parallelism used during the last execution of this operation
<i>Tot. Execs</i> - <i>Total Executions</i>	Number of times this work area was active
<i>Opt. Execs</i> - <i>Optimal Executions</i>	Number of times this work area ran in optimal mode
<i>Onepass</i> - <i>Onepass Executions</i>	Number of times this work area ran in one-pass mode
<i>Multipa...</i> - <i>Multipasses Executions</i>	Number of times this work area ran below the one-pass memory requirement

Column	Description
<i>Act. Time</i> - <i>Average Active Time</i>	Average time this work area is active in hundredths of a second
<i>Max Tseg...</i> - <i>Maximum Temporary Segment Size</i>	Maximum temporary segment size in bytes created by an instantiation of this work area. This column is null if this work area has never spilled to disk.
<i>Last Tseg</i> - <i>Last Temporary Segment Size</i>	Temporary segment size in bytes created in the last instantiation of this work area. This column is null if the last instantiation of this work area did not spill to disk.

- *Top 10 mem. cache con*

This tab page shows the top 10 consumers of memory cache, based on the view GV\$SQL\_WORKAREA. The information shown is the same as in the table above.

- *One-multipass workarea*

This tab page shows the work areas, the SQL text, the number of executions in the different modes, and the percentage of the total number of executions. The information shown is based on the views GV\$SQL and GV\$SQL\_WORKAREA:

Column	Description
<i>SQL Text</i>	First thousand characters of the SQL text for the current cursor
<i>Optimal ...</i> - <i>Optimal Executions</i>	Number of times this work area ran in optimal mode
<i>Optimal Pe</i> - <i>Optimal Percentage</i>	<i>Optimal Executions</i> as a percentage of <i>Total Executions</i>
<i>Onepass ...</i> - <i>Onepass Executions</i>	Number of times this work area ran in one-pass mode
<i>Onepass Pe</i> - <i>Onepass Percent</i>	<i>Onepass Executions</i> as a percentage of <i>Total Executions</i>
<i>Multipass ...</i> - <i>Multipasses Executions</i>	Number of times this work area ran below the one-pass memory requirement
<i>Multipass ...</i> - <i>Multipasses Percent</i>	<i>Multipasses Executions</i> as a percentage of <i>Total Executions</i>



Column	Description
<i>Total Exec</i>	Number of times this work area was active
- <i>Total Executions</i>	

- *SQL Workarea Histogram*

- *Histogram*

This tab page shows how many work areas were executed in optimal, one-pass, or multi-pass mode. The information shown is based on the view GV\$SQL\_WORKAREA\_HISTOGRAM:

Column	Description
<i>Inst Id</i>	Instance ID
RAC only	
<i>Inst Name</i>	Instance name
RAC only	
<i>Low bound</i>	Lower bound for the optimal memory requirement of work areas included in this row (bytes)
<i>High Bound</i>	Higher bound for the optimal memory requirement of work areas included in this row (bytes)
<i>Opt. Execs</i>	Number of times this work area ran in optimal mode
- <i>Optimal Executions</i>	
<i>Onepass</i>	Number of times this work area ran in one-pass mode
- <i>Onepass Executions</i>	
<i>Multipa - Multipass Executions</i>	Number of times this work area ran below the one-pass memory requirement
<i>Tot. Execs</i>	Number of times this work area was active
- <i>Total Executions</i>	

- *Percent optimal*

This tab page shows how many work areas were executed in optimal, one-pass, or multi-pass mode and the percentage. The information shown is based on the view GV\$SQL\_WORKAREA\_HISTOGRAM:

Column	Description
<i>Inst Id</i> RAC only	Instance ID
<i>Inst Name</i> RAC only	Instance name
<i>Optimal</i> - <i>Optimal Executions</i>	Number of work areas with an optimal memory requirement between <b>LOW_OPTIMAL_SIZE</b> and <b>HIGH_OPTIMAL_SIZE</b> that have been executed in optimal mode since instance startup
<i>Optimal Pe</i> - <i>Optimal Percent</i>	<i>Optimal Executions</i> as a percentage of <i>Total Executions</i>
<i>Onepass ...</i> - <i>Onepass Executions</i>	Number of work areas with an optimal memory requirement between <b>LOW_OPTIMAL_SIZE</b> and <b>HIGH_OPTIMAL_SIZE</b> that have been executed in one-pass mode since instance startup
<i>Onepass Pe</i> - <i>Onepass Percent</i>	<i>Onepass Executions</i> as a percentage of <i>Total Executions</i>
<i>Multipas ...</i> - <i>Multipasses Executions</i>	Number of work areas with an optimal memory requirement between <b>LOW_OPTIMAL_SIZE</b> and <b>HIGH_OPTIMAL_SIZE</b> that have been executed in multi-pass mode since instance startup
<i>Multipas ...</i> - <i>Multipasses Percent</i>	<i>Multipasses Executions</i> as a percentage of <i>Total Executions</i>
<i>Total Exec</i> - <i>Total Executions</i>	Number of times this work area was active

- *Workarea Executions*

This tab page shows how often work areas were executed in different modes. The information shown is based on the view GV\$SYSSTAT.

Column	Description
<i>Inst Id</i>	Instance ID
RAC only	
<i>Inst Name</i>	Instance name
RAC only	
<i>Name</i>	Statistic name
<i>Value</i>	Statistic value
<i>%</i>	Percentage of executions for each statistic name

- *Snapshot*

- *Current Operations*

This tab page shows currently active operations. The information shown is based on the view GV\$SQL\_WORKAREA\_ACTIVE.

Column	Description
<i>Inst Id</i>	Instance ID
RAC only	
<i>Inst Name</i>	Instance name
RAC only	
<i>SID</i>	Session identifier
<i>Oper. Type</i>	Type of operation using the work area (sort, hash join, group by, buffering, bitmap merge, or bitmap create)
<i>Exp. Size</i>	Expected size in KB for the work area
- <i>Expected workarea size</i>	
<i>Act. Used</i>	Amount of PGA memory in KB currently allocated for this work area
- <i>PGA Memory Currently</i>	
<i>Max Mem</i>	Maximum amount of memory used by this work area
- <i>Maximum memory used</i>	
<i>Passes</i>	Number of passes for this work area
- <i>Number of Passes</i>	

Column	Description
<i>TmpSeg ...</i> - Temporary Segment Size	Size in bytes of the temporary segment used for this work area.
<i>SQL Text</i>	Text of SQL statement

- *PGA Memory Usage*

This tab page shows currently active operations. The information shown is based on the view GV\$PROCESS.

Column	Description
<i>OS Program Name</i>	Operating system program name
<i>Inst Id</i> RAC only	Instance ID
<i>Inst Name</i> RAC only	Instance name
<i>PGA Memory</i>	PGA memory currently used by the process
<i>PGA Memory</i>	PGA memory currently allocated by the process
<i>Max PGA Memory</i>	Maximum PGA memory allocated by the process
<i>Process St.</i>	Process status
<i>SQL Text</i>	Text of SQL statement

- *PGA Advice*

- *Target Advice Size*

This tab page predicts how the cache hit percentage and over allocation count statistics displayed by the V\$PGASTAT performance view would be impacted if the value of the PGA\_AGGREGATE\_TARGET parameter is changed. The prediction is performed for various values of the PGA\_AGGREGATE\_TARGET parameter, selected around its current value. The advice statistic is generated by simulating the past workload run by the instance.

The information shown is based on the view V\$PGA\_TARGET\_ADVICE.

Column	Description
<i>Inst Id</i> RAC only	Instance ID

Column	Description
<i>Inst Name</i> RAC only	Instance name
<i>TARGET (MB)</i>	Operating system program name
<i>Val</i> - <i>Estimated value of the cache hit percent</i>	Value of <code>PGA_AGGREGATE_TARGET</code> for this prediction (in bytes)
<i>Overall.Cn.</i> - <i>Overalloc. count</i>	Estimated number of PGA memory over-allocations if the value of <code>PGA_AGGREGATE_TARGET</code> is set to <code>PGA_TARGET_FOR_ESTIMATE</code> .

- *Advice Histogram*

Column	Description
<i>Inst Id</i> RAC only	Instance ID
<i>Inst Name</i> RAC only	Instance name
<i>PGA_TARGET</i>	<code>PGA_TARGET_FACTOR</code> , equal to <code>PGA_TARGET_FOR_ESTIMATE</code> divided by current value of <code>PGA_AGGREGATE_TARGET</code> .
<i>LOW KB</i>	Lower boundary for the optimal memory requirement of work areas included in this row, in bytes
<i>HIGH_KB</i>	Upper boundary for the optimal memory requirement of work areas included in this row, in bytes
<i>Optimal Ex</i> - <i>Optimal Executions</i>	Number of work areas with an optimal memory requirement between <i>LOW KB</i> and <i>HIGH_KB</i> , which are predicted to run optimally when <code>PGA_AGGREGATE_TARGET = PGA_TARGET_FOR_ESTIMATE</code> .
<i>Onepass Ex</i> - <i>Onepass Executions</i>	Number of work areas with an optimal memory requirement between <i>LOWKB</i> and <i>HIGH_KB</i> , which are predicted to run one-pass when <code>PGA_AGGREGATE_TARGET = PGA_TARGET_FOR_ESTIMATE</code> .

Column	Description
<i>Multipasse</i> - <i>Multipasses Executions</i>	Number of work areas with an optimal memory requirement between <i>LOWKB</i> and <i>HIGH_KB</i> , which are predicted to run multi-pass when <code>PGA_AGGREGATE_TARGET = PGA_TARGET_FOR_ESTIMATE</code> .

### 3.6.5.3 Undo Statistics

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you check the undo statistics provided by the Oracle view GV\$UNDOSTAT. You can see:

- Daily summaries
- Undo statistics: daily and average values
- Maximum space consumption for undo tablespaces

#### Use

You choose ► *Performance* ► *Statistical Information* ► *Undo Statistics* ► and the required tab page *Daily Summaries*, *Undo statistics*, or *Max space consumption*.

#### Structure

##### i Note

Entries marked “RAC only” appear only for Oracle Real Application Cluster (RAC)

- *Daily Summaries*  
This tab page displays daily summaries of undo statistics:

Column	Description
<i>Instance Id</i> RAC only	Database instance ID

Column	Description
<i>begin date</i>	Begin date for the analysis
<i>begin time</i>	Begin time for the analysis
<i>end date</i>	End date for the analysis
<i>end time</i>	End time for the analysis
<i>last active undo tablespace</i>	<p>Last active undo tablespace.</p> <p>If more than one undo tablespace was active in the analysis period, the active undo tablespace that was active at the <b>end</b> of the period is reported.</p>
<i>total number of undo blocks</i>	<p>Total number of undo blocks</p> <p>You can use this column to obtain the consumption rate of undo blocks, and thereby estimate the size of the undo tablespace needed to handle the workload on your system.</p>
<i>total number of transactions</i>	Total number of transactions
<i>length of the longest query (sec)</i>	<p>Length of the longest query in seconds</p> <p>You can use this statistic to estimate the proper setting of the UNDO_RETENTION initialization parameter.</p>
<i>highest Nbr. of TAs executed Concurrently</i>	Highest number of transactions executed concurrently
<i>attempts to obtain undo space</i>	Number of attempts to obtain undo space by stealing unexpired extents from other transactions
<i>unexpired undo blocks removed</i>	Number of unexpired undo blocks removed so they can be used by other transactions
<i>unexpired undo blocks reused</i>	Number of unexpired undo blocks reused by other transactions
<i>attempts to steal expired undo blocks</i>	Number of attempts to steal expired undo blocks from other undo segments
<i>expired undo blocks stolen from segments</i>	Number of expired undo blocks stolen from other undo segments
<i>expired undo blocks reused</i>	Number of expired undo blocks reused within the same undo segments

Column	Description
<i>number of occurrences of ORA-01555</i>	Number of occurrences of ORA-01555  You can use this statistic to decide whether or not the <code>UNDO_RETENTION</code> initialization parameter is set properly in view of the size of the undo tablespace. You can reduce the occurrence of this error by increasing the value of <code>UNDO_RETENTION</code> .
<i>nbr of times space was requested</i>	Number of times space was requested in the undo tablespace and there was no free space available. That is, all of the space in the undo tablespace was in use by active transactions. You can correct this by adding more space to the undo tablespace.
<i>number of transactions per second</i>	Number of transactions per second

#### **i** Note

Each row in the table shows information for a 10-minute period, as shown by the difference between *Begin Time* and *End Time*. The information is derived from the view `GV$UNDOSTAT`, which also holds information in this way.

You can see information from the previous 7 days, since there are 1008 rows in the view `GV$UNDOSTAT`.

- *Undo statistics*

This tab page displays the same information as above, with individual tab pages for each day. There is also a *Daily statistics* tab page showing the statistics - maximum, minimum, average, and total - for each day.

#### **i** Note

RAC only

The tab page *Undo statistics* does **not** appear when you choose ► *DB instances* ► *Total* ▾ to show the total of all RAC instances.

- *Max Space Consumption*

This tab page displays maximum space consumption for undo tablespaces:

Column	Description
<i>Instance Id</i>	Database instance ID
RAC only	
<i>undo tablespace name</i>	Name of the undo tablespace for which the maximum space consumption has been computed



Column	Description
<i>total undo tablespace size in MB</i>	Total size of the undo tablespace in MB
<i>max. used undospace in MB</i>	Maximum used undo space in MB
<i>max. used in %</i>	Used undo space as a percentage of the maximum

## 3.6.5.4 Performance Database

### Definition

This sub-monitor in the SAP/Oracle Database Monitor gives you an overview of the load and performance of the Oracle instance.

### Use

You choose ► [Performance](#) ► [Statistical Information](#) ► [Performance Database](#) ► in the DBA Cockpit.

You can use this sub-monitor to see if:

- The database load has changed recently
- The database instance has load peaks
- One database instance is more heavily loaded than other instances - useful with Oracle Real Application Cluster (RAC)

### Structure

The following structure applies to the tabs [Overview](#), [Intervals](#), [Peak 10-12](#), [Peak 14-16](#):

Column	Description
<i>Weekday</i>	Day of the week for the snapshot
<i>Date</i>	Date of the snapshot
<i>Time</i>	Time of the snapshot
<i>User Calls</i>	Number of user calls such as login, parse, fetch, or execute

Column	Description
<i>Recursive Calls</i>	Internal SQL statement or SQL statement in PL/SQL statement
<i>User / Recursive Calls</i>	Ratio of user to recursive Calls
<i>User Commits</i>	Number of commits and rollbacks
<i>Parses</i>	Total number of parse calls (hard and soft). A soft parse is a check on an object already in the shared pool, to verify that the permissions on the underlying object have not changed.
<i>Reads / User Call</i>	Amount of logical reads per user call. Should be less than 20.
<i>Logical Reads</i>	Sum of "db block gets" and "consistent gets"
<i>Physical Reads</i>	Number of physical reads
<i>Physical Reads Direct</i>	Number of reads directly from disk, bypassing the buffer cache
<i>Physical Reads Direct (LOB)</i>	Number of LOB reads directly from disk, bypassing the buffer cache
<i>Buffer Quality</i>	Percentage of how many db blocks are found in the db cache and haven't to read from disk.
<i>Physical Writes</i>	Number of physical reads
<i>Table Fetch by RowID</i>	Number of rows accessed by RowID, including all rows accessed using indexes
<i>Table Fetch Continued Row</i>	Number of times when second row piece of chained rows is fetched. A high number indicates that rows are chained.
<i>Table Scans Rows Gotten</i>	Number of rows accessed by all full table scans. This is not the same as the number of rows returned because only qualifying rows are returned.
<i>Table Scans Blocks Gotten</i>	Number of blocks accessed by all full table scans
<i>Redo Blocks Written</i>	Total number of redo blocks written
<i>Redo Write Time (ms)</i>	Total redo write time since database start in milliseconds
<i>Avg. Redo Write time (msec)</i>	Average time the LGWR needs to write the redo log information from buffer to disk in milliseconds

Column	Description
<a href="#">Buffer Busy Waits</a>	Number of times block access failed because another process held the block in incompatible mode. If this statistic is over 10% of logical reads then use V\$WAITSTAT to check contention.
<a href="#">Buffer Busy Waits Time (sec)</a>	Total buffer busy wait time since database start in seconds
<a href="#">Avg. Buffer Busy Waits Time (msec)</a>	Average time a session has to wait for the event buffer busy waits in milliseconds
<a href="#">Full Table Scans</a>	Sum of full table scans for long and short tables: <ul style="list-style-type: none"> <li>• Short table scans are against tables with 4 or less database blocks.</li> <li>• Long table scans are against tables with 5 or more database blocks.</li> </ul>

Note the following about the tabs:

- All tabs include load and performance data for the days where snapshot information is available.
- The [Overview](#) tab displays the data accumulated since database start on a daily basis.
- The [Intervals](#) tab displays the data for each day.

#### **i** Note

To see the load and performance data for every snapshot on a certain day, double-click the desired day in the [Overview](#) or [Intervals](#) tab.

- The tabs [Peak 10 - 12](#) and [Peak 14 - 16](#) show the load and performance data at the peak times of 10:00 to 12:00 and 14:00 to 16:00. You cannot double-click here.

## 3.6.5.5 System Metrics

### Definition

This monitor in the SAP/Oracle Database Monitor gives you metrics on the Oracle database system with data derived from the GV\$SYSMETRIC\_HISTORY or DBA\_HIST\_SYSMETRIC\_HISTORY view.

For more information about the licensing of the diagnostic pack, see SAP Notes [1028068](#) and [1250596](#).

### Use

You choose ► [Performance](#) ► [Statistical Information](#) ► [System Metrics](#) in the DBA Cockpit.

## i Note

The default display when you start the monitor is from `GV$SYSMETRIC_HISTORY` with a timeframe of the last hour.

You can set the following fields in the *Selection box*:

- *Metrics datasource*  
Use this to display data from the *GV\$-VIEW* or the *DBA View*.
- *Avail. Timeframe*  
Use this to set the timeframe for the displayed data, either with *use GV\$-VIEW* or *use DBA view*. In each case, the monitor uses the maximum available timeframe, as shown in the *From* and *TO* fields.
- *Set to last Hour*  
Use this to set the timeframe for the displayed data to the last hour, as shown in the *From* and *TO* fields.

You choose *Load Data* to refresh the data in the table.

## Structure

- *System Metrics*

Column	Description
<i>Snapshot ID</i>	Unique snapshot ID
<i>Database ID</i>	Database ID for snapshot
<i>Instance number</i>	Instance number for snapshot
<i>Begin Time</i>	Begin of snapshot time interval
<i>End Time</i>	End of snapshot time interval
<i>Int Size</i>	Time interval in hundredths of a second
<i>Group ID</i>	Metric group ID
<i>Metric ID</i>	Metric ID
<i>Metric Name</i>	Metric name
<i>Value</i>	Metric value
<i>Metric Unit</i>	Metric unit description

- *System Metrics Graph*  
This is a graph of the information from the *System Metrics* tab. No graph is initially displayed. Select the required parameters in *Metric Group* and *Metric Name*. The selection here includes all possible metric values, even if your system has been set up not to gather certain metrics.

Choose [Reload Graph](#) to display the graph:

- The vertical Y axis of the graph is dynamic, based on the maximum value of the selected metric in the chosen timeframe. The unit of measure for the selected metric in the vertical axis is displayed in the [METRIC UNIT](#) field.
- The horizontal X axis shows time information for the selected data.
- You can click on a colored node in the graph to see a popup showing all timestamps and values for that particular node.

## 3.6.5.6 System Summary Metrics

### Definition

This monitor in the SAP/Oracle Database Monitor gives you system summary metrics on the Oracle database system with data derived from the `GV$SYSMETRIC_SUMMARY` or `DBA_HIST_SYSMETRIC_SUMMARY` view.

For more information about the licensing of the diagnostic pack, see SAP Notes [1028068](#) and [1250596](#).

### Use

You choose [Performance](#) [Statistical Information](#) [System Summary Metrics](#) in the DBA Cockpit.

#### i Note

The default display when you start the monitor is from `GV$SYSMETRIC_SUMMARY` with a timeframe of the last hour.

You can set the following fields in the [Selection box](#):

- [Metrics datasource](#)  
Use this to display data from the [GV\\$-VIEW](#) or the [DBA View](#).
- [Avail. Timeframe](#)  
Use this to set the timeframe for the displayed data, either with [use GV\\$-VIEW](#) or [use DBA view](#). In each case, the monitor uses the maximum available timeframe, as shown in the [From](#) and [TO](#) fields.
- [Set to last Hour](#)  
Use this to set the timeframe for the displayed data to the last hour, as shown in the [From](#) and [TO](#) fields.

You choose [Load Data](#) to refresh the data in the table.

### Structure

- [System Summary](#)

Column	Description
<i>Snapshot ID</i>	Unique snapshot ID
<i>DB ID</i>	Database ID for snapshot
<i>Inst_ID_CH</i>	Instance number for snapshot
<i>Snapshot begin time</i>	Begin of snapshot time interval
<i>Snapshot end time</i>	End of snapshot time interval
<i>Int size</i>	Time interval in hundredths of a second
<i>Group ID</i>	Metric group ID
<i>Metric ID</i>	Metric ID
<i>Metric Name</i>	Metric name
<i>Metric Unit</i>	Metric unit description
<i>Num Interv</i>	Number of intervals observed
<i>Minval</i>	Minimum value observed
<i>Maxval</i>	Maximum value observed
<i>Average</i>	Average over the period
<i>StdDeviat</i>	One standard deviation

- **System Summary Graph**

This is a graph of the information from the *System Summary* tab. No graph is initially displayed. Select the required parameters in *Metric Group* and *Metric Name*. The selection here includes all possible metric values, even if your system has been set up not to gather certain metrics.

Choose *Reload Graph* to display the graph:

- The graph differs according to the selected view:
  - For the GV\$ view there is only one timestamp stored, you only see three single points in the graph for maximum, minimum, and average values. There is no line in the graph.
  - For the DBA view, you see three lines for the selected metric, one for the minimum, one for the maximum, and one for the average values.
- Numeric values for the overall *Minimum Value*, *Maximum Value*, and *Average Value* are shown above the graph.
- The vertical Y axis of the graph is dynamic, based on the maximum value of the selected metric in the chosen timeframe. The unit of measure for the selected metric in the vertical axis is displayed in the *METRIC UNIT* field.
- The horizontal X axis shows time information for the selected data.
- You can click on a colored node in the graph to see a popup showing all timestamps and values for that particular node.

## 3.6.5.7 File Metrics

### Definition

This monitor in the SAP/Oracle Database Monitor gives you file metrics with data derived from the `GV$FILEMETRIC_HISTORY` or `DBA_HIST_FILEMETRIC_HISTORY` view.

For more information about the licensing of the diagnostic pack, see SAP Notes [1028068](#) and [1250596](#).

### Use

You choose **Performance** > **Statistical Information** > **File Metrics** in the DBA Cockpit.

#### Note

The default display when you start the monitor is from `GV$FILEMETRIC_HISTORY` with a timeframe of the last hour.

You can set the following fields in the *Selection box*:

- *Metrics datasource*  
Use this to display data from the *GV\$-VIEW* or the *DBA View*.
- *Avail. Timeframe*  
Use this to set the timeframe for the displayed data, either with *use GV\$-VIEW* or *use DBA view*. In each case, the monitor uses the maximum available timeframe, as shown in the *From* and *TO* fields.
- *Set to last Hour*  
Use this to set the timeframe for the displayed data to the last hour, as shown in the *From* and *TO* fields.

You choose *Load Data* to refresh the data in the table.

### Structure

- *File Metrics*

Column	Description
<i>Snapshot ID</i>	Unique snapshot ID
<i>DB ID</i>	Database ID for snapshot
<i>Inst_ID_CH</i>	Instance number for snapshot
<i>Fst file#</i>	File number

Column	Description
<i>File name</i>	File name
<i>Tablespace</i>	Tablespace name
<i>CreatTime</i>	Timestamp of file creation
<i>Snapshot begin time</i>	Begin of snapshot time interval
<i>Snapshot end time</i>	End of snapshot time interval
<i>Int size</i>	Time interval in hundredths of a second
<i>Group ID</i>	Metric group ID
<i>AvgRdTime</i>	Average file read time
<i>AvgWrtTime</i>	Average file write time
<i>Phys rds</i>	Number of physical reads
<i>Phys wrt</i>	Number of physical writes
<i>Blk rd</i>	Number of physical block reads
<i>Blk wr</i>	Number of physical block writes

- *File Metrics Graph*

This is a graph of the information from the *File Metrics* tab. No graph is initially displayed.

Select what you want to display in *FILE/TABLESPACE*:

- *SINGLE DATAFILE* - this is the default  
In *Selection Box for Datafile or Tablespace* double-click to select the required datafile.
- *FOR TABLESPACE* - displays all datafiles of a tablespace  
In *Selection Box for Datafile or Tablespace* double-click to select any datafile from the required tablespace.
- *ALL DATAFILES* - displays all datafiles on the system  
This disregards any selection in *Selection Box for Datafiles or Tablespaces*.  
We only recommend this for small systems because for larger systems the display becomes too cluttered.

In *IO Selection* you can select the required input/output metric for the graph. The default is *Avg READ and WRITE TIMES*. When you change this selection, to refresh the graph display you have to repeat the selection described above in *Selection Box for Datafile or Tablespace* (or you can also repeat the selection of *ALL DATAFILES* if you are viewing all datafiles).

The graph has the following features:

- The vertical Y axis of the graph is dynamic, based on the maximum value of the metric selected in *IO Selection* for the chosen timeframe.
- The horizontal X axis shows time information for the selected data.
- You can click on a colored node in the graph to see a popup showing all timestamps and values for that particular node.



- At the bottom of the graph there is a legend showing the meaning of each colored line. If there is not enough room, the legend cannot be displayed, which can happen if you select all datafiles or a large tablespace with many datafiles.

## 3.6.5.8 Wait Class Metrics

### Definition

This monitor in the SAP/Oracle Database Monitor gives you wait class metrics on the Oracle database system. Data is derived from the GV\$WAITCLASSMETRIC\_HISTORY or DBA\_HIST\_WAITCLASSMET\_HISTORY view.

For more information about the licensing of the diagnostic pack, see SAP Notes [1028068](#) and [1250596](#).

### Use

You choose ► [Performance](#) ► [Statistical Information](#) ► [Wait Class Metrics](#) in the DBA Cockpit.

#### i Note

The default display when you start the monitor is from GV\$WAITCLASSMETRIC\_HISTORY with a timeframe of the last hour.

You can set the following fields in the [Selection box](#):

- [Metrics datasource](#)  
Use this to display data from the [GV\\$-VIEW](#) or the [DBA View](#).
- [Avail. Timeframe](#)  
Use this to set the timeframe for the displayed data, either with [use GV\\$-VIEW](#) or [use DBA view](#). In each case, the monitor uses the maximum available timeframe, as shown in the [From](#) and [TO](#) fields.
- [Set to last Hour](#)  
Use this to set the timeframe for the displayed data to the last hour, as shown in the [From](#) and [TO](#) fields.

You choose [Load Data](#) to refresh the data in the table.

### Structure

- [Wait Class](#)

Column	Description
<a href="#">Snapshot ID</a>	Unique snapshot ID

Column	Description
<i>DB ID</i>	Database ID for snapshot
<i>Inst_ID_CH</i>	Instance number for snapshot
<i>W Class ID</i>	Identifier for the class of the wait event
<i>W Class NA</i>	Name for the class of the wait event
<i>Snapshot begin time</i>	Begin of snapshot time interval
<i>Snapshot end time</i>	End of snapshot time interval
<i>Intsize</i>	Time interval in hundredths of a second
<i>Group ID</i>	Metric group ID
<i>AvgWtCnt</i>	Average wait count
<i>DBTimeWait</i>	Database wait time
<i>Time waited</i>	Elapsed wait time waited
<i>Wait Cnt</i>	Number of times waited

- [Wait Class Graph](#)

This is a graph of the information from the [Wait Class](#) tab. No graph is initially displayed.

Select the required parameters in [Waitclass](#) and [Wait Statistics](#), then choose [Reload Graph](#) to display the graph:

- The vertical Y axis of the graph corresponds to the selected [Wait Statistic](#) and is dynamic, based on the maximum value of the selected metric in the chosen timeframe.
- The horizontal X axis corresponds to the selected [Waitclass](#).
- You can click on a colored node in the graph to see a popup showing all timestamps and values for that particular node.

## 3.6.5.9 DNFS Monitor

### Definition

This monitor in the SAP/Oracle Database Monitor supports the Direct Network File System (DNFS) in Oracle version 11.2.

#### i Note

Since the DNFS feature is only available as of Oracle 11.2, the monitor is hidden for systems running with older versions of the Oracle database. See SAP Note [1366285](#).

## Use

You choose ► [Performance](#) ► [Statistical Information](#) ► [DNFS Monitor](#) ► in the DBA Cockpit.

## Structure

- [Channel Sums](#)

This displays the result of the following query:

```
select path, sum(sends), sum(recvs)
from v$dnfs_channels group by path
```

- [Channels](#)

This displays the result of the following query:

```
select pnum, svrname, path, ch_id, svr_id, sends, recvs, pings
from v$dnfs_channels
```

- [Servers](#)

This displays the result of the following query:

```
select id, svrname, dirname, mntport, nfSPORT, wtmax, rtmax
from v$dnfs_servers
```

## 3.6.6 Feature Monitoring

### Use

These sub-monitors in the SAP/Oracle Database Monitor show features.

### 3.6.6.1 Automatic Segment Space Management

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you check the automatic segment space management (ASSM) of the database. You can see:

- Tablespaces with ASSM
- All tablespaces
- Tables with ASSM
- Tables without ASSM

ASSM simplifies and blocks the storage of tables and indexes by replacing linked-list freelists with bitmap freelists, which are faster and more efficient. ASSM reduces [buffer busy waits \[page 40\]](#).

## Use

You choose ► [Performance](#) ► [Feature Monitoring](#) ► [Automatic Segment Space Management](#) ► in the DBA Cockpit and then the required tab page.

In [Tables with ASSM](#) and [Tables without ASSM](#), output is limited to the first 50 tables. Choose [Select Table](#) to display information from a table of your choice.

## Structure

### i Note

Entries marked “RAC only” appear only for Oracle Real Application Cluster (RAC)

- [Tablespaces with ASSM](#)

This tab page displays information on tablespaces with ASSM:

Column	Description
<a href="#">Name</a>	Name of tablespace
<a href="#">Block Size</a>	Tablespace blocksize
<a href="#">Status</a>	Tablespace status. For example, <a href="#">ONLINE</a> , <a href="#">OFFLINE</a> , <a href="#">INVALID</a> .
<a href="#">Contents</a>	Type of tablespace: <a href="#">TEMPORARY</a> for dedicated temporary tablespaces or <a href="#">PERMANENT</a> for tablespaces that can store both temporary sort segments and permanent objects.
<a href="#">Extent Management</a>	Extent management, <a href="#">LOCAL</a> or <a href="#">DICTIONARY</a>
<a href="#">Allocation Type</a>	Allocation type, <a href="#">USER</a> , <a href="#">SYSTEM</a> or <a href="#">UNIFORM</a>
<a href="#">Segment Space Mngt</a>	Segment space management, <a href="#">AUTO</a>

- [All Tablespaces](#)

This tab page shows the same information as in the table above, but includes tablespaces with and without ASSM.

- [Tables with ASSM](#)

This tab page displays information on tablespaces with ASSM:

Column	Description
<i>Table Name</i>	Name of table
<i>Tablespace Name</i>	Name of tablespace
<i>Used Space (Bytes)</i>	Used space in the table in bytes
<i>Unused Space (Bytes)</i>	Unused space in the table in bytes
<i>Meta Data Blocks</i>	Total blocks reported by DBA_TABLES minus sum of values reported by PL/SQL routine SPACE_USAGE

#### **i** Note

Choose [Select Table](#) to display information on a selected single table or a group of tables.

- [Tables without ASSM](#)

This tab page shows the same information as in the table above, but only for tables without ASSM.

## 3.6.6.2 Online Redefinition

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you check the online redefinition of tables in the database. You can see:

- Tables in redefinition mode
- Operations overview

Online redefinition lets you redefine tables - add, rename, or drop columns - while keeping the table fully online and available.

This monitor makes it easier for you to get detailed information about which tables have been redefined online, when, and which Data Manipulation Language (DML) operations took place.

### Use

You choose ► [Performance](#) ► [Feature Monitoring](#) ► [Online Redefinition](#) ► in the DBA Cockpit and then the required tab page.

## Structure

### i Note

Entries marked “RAC only” appear only for Oracle Real Application Cluster (RAC).

- *Tables in redefinition mode*

This tab page displays the tables that are currently in online redefinition mode:

Column	Description
<i>Table Name</i>	Name of table
<i>Created</i>	Date when the table was created
<i>DML Operation</i>	DML operation
<i>Occurrence</i>	Number of times for this DML operation on the table

- *Operations Overview*

This tab page displays the time of each DML operation on the redefined tables:

Column	Description
<i>Table Name</i>	Name of table
<i>Operation</i>	DML operation
<i>Date</i>	Date when the table was created
<i>Hour</i>	Hour at which the redefinition occurred
<i>Occurrence</i>	Number of times for this DML operation on the table

### 3.6.6.3 Resumable Space Allocation

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you check the resumable space allocation. If a statement is suspended for space allocation reasons, the resumable space allocation feature enables the statement to be resumed, so that the work done so far is saved.

## Use

You choose ► [Performance](#) ► [Feature Monitoring](#) ► [Resumable Space Allocation](#) ►.

## Structure

### Note

Entries marked “RAC only” appear only for Oracle Real Application Cluster (RAC)

This screen displays the following information:

Column	Description
<i>User ID</i>	User ID of the resumable statement owner
<i>Username</i>	User name of the resumable statement owner
<i>Session ID</i>	Session identifier
<i>Inst ID</i>	Instance ID of resumable statement
<i>Coord Inst_ID</i>	Inst ID on which the Parallel Coordinator is running
<i>Coord Sess ID</i>	Session ID of the Parallel Coordinator
<i>Status</i>	Statement status.  Possible values: <i>RUNNING</i> , <i>SUSPENDED</i> , <i>ABORTED</i> , <i>ABORTING</i> , <i>TIMEOUT</i>
<i>Timeout</i>	Timeout of the resumable statement
<i>Start Time</i>	Local start time of the resumable statement
<i>Suspend Time</i>	Local last time when the resumable statement was suspended
<i>Resume Time</i>	Local last time when the resumable statement was resumed
<i>Name</i>	The name given in the resumable clause of this resumable statement.
<i>SQL Text</i>	SQL text of the resumable statement
<i>Error Number</i>	The error code of the last correctable error
<i>Error Parameter 1</i>	Parameter for error message 1

Column	Description
<a href="#">Error Parameter 2</a>	Parameter for error message 2
<a href="#">Error Parameter 3</a>	Parameter for error message 3
<a href="#">Error Parameter 4</a>	Parameter for error message 4
<a href="#">Error Parameter 5</a>	Parameter for error message 5
<a href="#">Error Message</a>	The error message corresponding to <a href="#">Error Number</a> .

## 3.6.6.4 Parallel Query

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you check parallel queries. Instead of using a single process for one SQL statement, in parallel queries the work is spread across multiple processes. This is useful where there is a lot of data in operations like full table scans of large tables, creation of large indexes, or bulk inserts, updates, and deletes.

For more information, see SAP Note [1731514](#).

### Use

You choose [► Performance](#) [► Feature Monitoring](#) [► Parallel Query](#) in the DBA Cockpit.

If the RFC connection is maintained, additional information such as DDIC information and the position of the ABAP source call is available for the select query. If an RFC connection is not maintained, only the explain plan can be viewed for the query.

### Structure

This screen displays the following information:

Column	Description
<a href="#">SID</a>	Session identifier
<a href="#">Instance</a>	Oracle instance



Column	Description
<i>Username</i>	Oracle user name
<i>SQL ID</i>	SQL identifier
<i>Role</i>	Role of the process during parallel execution
<i>Action</i>	Name of action currently being executed
<i>SQL Text</i>	SQL statement text
<i>Degree</i>	Degree of parallelism being used
<i>Req Degree</i>	Degree of parallelism requested when the statement was started

## 3.6.6.5 Segments with Activated Parallelism

### Definition

This sub-monitor in the SAP/Oracle Database Monitor shows the segments that have parallel degree activity based on the Oracle SQL script `Segments_ParallelDegreeActive.txt` given in SAP note [1438410](#). This monitor displays the segments in the instance having parallel degree active. The data shown is based on the Oracle Automatic Workload Repository (AWR). For more information, see SAP note [1704439](#).

You choose ► [Performance](#) ► [Feature Monitoring](#) ► [Segments with Activated Parallelism](#) ►

### Structure

Column	Description
<i>Owner</i>	Segment owner
<i>Segment name</i>	Segment name
<i>Command</i>	Alter command for the segment
<i>Segment type</i>	Segment type
<i>Degree</i>	Degree of parallelism
<i>Instances</i>	Number of instances

## 3.6.7 Additional Functions

### Use

These sub-monitors in the SAP/Oracle Database Monitor show additional functions.

### 3.6.7.1 SQL Command Editor

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you display the results of native Oracle select statements, which you enter in an editor.

If your statement selects only content of tables with owner SYS or PUBLIC, the monitor displays the result of the select statement. Otherwise the monitor executes statements and displays miscellaneous runtime information.

### Use

You choose ► *Performance* ► *Additional Functions* ► *SQL Command Editor* ► in the DBA Cockpit.


You can use the editor in the following ways:


- Interactive  
The sub-monitor consists of an editor screen where you enter the SQL statement and a result screen that displays the result of the SQL statement.
- Batch  
The result is written to a spool list.
- Database table  
The result is written to a database table. This lets you store and read it later, or evaluate it with an SQL script. To do this, make an entry of the form **\$\$\$<RESULT\_ID>\$\$\$** before the SQL query.

#### ❁ Example

```
$$$ATEST$$$SELECT * FROM V$INSTANCE
```

The system stores the result with the ID ATEST in table ORA\_SQLC\_DATA. The table ORA\_SQLC\_HEAD stores the result structure.

For more information, see SAP Note [1114566](#) .

- Explain plan  
You have the option to display an SQL execution plan (explain). For more information, see SAP Note [1297012](#) .

## Structure

You can choose:

- [SQL Command > Save](#) or [Load](#) or [Delete](#)  
These functions ask you for an ID so that the statement can be saved to, loaded to, or deleted from the SAP database.  
The advantage of storing your statements is that you can share the same statement with other users.
- [SQL Command > Parse](#)  
This function starts a simple parser to check the syntax. This parser only makes sure that the monitor is able to generate a display structure and display the result of the statement. It also checks the owner of the tables and views that have to be read.  
It does **not** check the complete Oracle syntax. Therefore, it does not guarantee that the statement can be executed.
- [SQL Command > Execute](#)  
This function starts the parser, executes the statement, and displays the result of the statement.
- [SQL Command > Execute Background](#)  
This function creates a background job for the execution of the current SQL statement. It is best to execute in the background when you expect a long runtime for the statement.  
The system asks for some parameters as follows:
  1. **Background job parameters**
    - [Statement ID](#)  
This is the ID for saving the current SQL statement in the SAP database. The current statement is stored under this ID for later execution. If you use the default value \$TMP\_JOB, this overwrites an already existing statement with this ID. Use this default only if you are sure that there is no other job that uses this ID awaiting execution.
    - [Print result of statement](#)  
Select this box if you want to get the resulting data.
    - [Print runtime data for statement](#)  
Select this to receive the runtime information for the statement.
    - [Start execution immediately](#)  
Select this to start execution immediately.
    - [Scheduled start at](#)  
Select this to start execution at the date and time you enter.
    - [No start after](#)  
You can prevent the system from executing the statement after this time. For example, this is useful to prevent execution when you expect the system load to be high.
    - [Print time](#)  
Here you can select whether the results are sent to the SAP spooler only or if they are printed immediately.
  2. **Background print parameters**  
Here you can specify the printer and other print parameters.
- [Goto > Job overview](#)  
After starting a statement in background execution you can find your job here.
- [Goto > Spool requests](#)  
After starting a statement in background execution you can find your spool lists here.

- [Save as local file](#) or [Load local file](#)

You can save your SQL statement to a local file or load an SQL statement from a local file into the editor.

## Syntax

- A statement must have the following syntax:

```
SELECT [ hint ] [ { DISTINCT | UNIQUE } | ALL ] select_list
FROM table_reference [, table_reference]...
[ WHERE condition ]
[ hierarchical_query_clause ]
[ group_by_clause ]
[ HAVING condition ]
[ { UNION | UNION ALL | INTERSECT | MINUS } ( subquery ) ]
[ order_by_clause ]
```

You can put comments between `"/*+"` and `"*/"`

- A select list must have the following syntax:

```
{ * |
{[table_alias.]dbfieldname | expression} alias [, [table_alias.]dbfieldname |
expression} alias] ... }
```

An expression within this select list can use a calculation operator such as `+`, `-`, `*`, `/`, `||`. Unary functions (`LN`, `MIN`, `AVG` . . . ), null, or numbers, are also allowed.

- A table reference must have the following syntax:

```
{(select statement) [table_alias] | table [table_alias]}
```

- If you enter one of the following commands, you receive the same table information as if you had entered the command with the same name on Oracle with SQLPLUS:

- `DESC <table name>`
- `DESCR <table name>`
- `DESCRIBE <table name>`

Otherwise the syntax follows the SQL standard.

## Conventions and Restrictions

- Each column specified in the select list becomes a column in the output list.
- If a select list element is specified with a column alias, this alias is used as header text in the output list. Otherwise the program uses the field name of the select list element as header text. If a select list element is an expression (that is, without a database field), the alias is obligatory.

### Caution

For every expression that is not a database field, use a column alias.

### Example

The following statement cannot be processed because the only select list element is an expression.

```
select to_char(startup_time,'YYYY-MM-DD-HH24.MI.SS') from v$instance ...
```

Since expressions are not always related to exactly one database field, the monitor requires an alias name:

```
select to_char(startup_time,'YYYY-MM-DD-HH24.MI.SS') as startup from
v$instance
```

- Every column alias that is specified in the select list of a sub-query can be used in the same way as a database field name in the select statement.
- If more than one table is specified in the from clause, the columns are matched to one table for reasons of uniqueness. If a column name occurs in more than one table, uniqueness cannot be guaranteed. In this case you have to specify a table alias before the column name (that is, database field name).

### ⚠ Caution

When more than one table is specified and column names that have to be outputted occur in more than one table, use a table alias.

### 🔗 Example

If both `table_a` and `table_b` have a field with identical name, `field_1`, you have to specify the table alias for uniqueness:

```
select a.field_1 as tc_a, b.field_1 as tc_b from table_a a, table_b b
```

- The SAP List Viewer (ALV) does **not** display leading blanks that the Oracle LPAD editing function has inserted into the SQL statement. To display leading blanks, you need to switch to list output by choosing [View](#) in the ALV grid.

## Example

Here are some examples to show features of this sub-monitor:

- This statement shows an interesting rate statistics for read, write, and hit using a sub-select:
 

```
select LogicalIO, PhysicalReads, PhysicalReadsDirect,
PhysicalReadsDirectLob, PhysicalWrites,
round((1-(PhysicalReads-PhysicalReadsDirect- PhysicalReadsDirectLob)/
(LogicalIO-PhysicalReadsDirect- PhysicalReadsDirectLob))*100,2)
HitRatio
from (select SUM(DECODE(Name, 'session logical reads',Value,0)) LogicalIO,
SUM(DECODE(Name,'physical reads',Value,0)) PhysicalReads,
SUM(DECODE(Name,'physical writes',Value,0)) PhysicalWrites,
SUM(DECODE(Name,'physical reads direct',Value,0)) PhysicalReadsDirect,
SUM(DECODE(Name,'physical reads direct (lob)', Value,0))
PhysicalReadsDirectLob
from v$sysstat
where name in ('session logical reads',
'physical reads','physical reads direct',
'physical reads direct (lob)', 'physical writes'))
```
- This statement shows runtime statistics for reading SAP table `t100` and a `rowid` statistic:
 

```
Select rowid, a.* from t100 a
```
- This statement shows how to use a table cast:
 

```
select * from table(dbms_xplan.display())
```

## 3.6.7.2 Display GV\$-Views

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you see list the views existing in an Oracle database and display their contents. The list of views is taken from `V$fixed_view_definition`.

### Use

You choose ► [Performance](#) ► [Additional Functions](#) ► [Display GV\\$ Views](#) ► in the DBA Cockpit.

### Structure

The GV\$ views are arranged in a two-column list in alphabetic order of the views' names.

To see more detail, double-click an entry in the table.

The columns displayed in detail depend on the view. For more information, see the Oracle documentation.

### Integration

Some features or services in Oracle have their own views. When such features are not active, their related views do not provide any result.

One such feature is the Oracle Log Miner. Its related views (`logminer_*`) cannot be queried unless the service is up and running.

## 3.6.7.3 Display DBA Tables

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you display the contents of Oracle database DBA tables, which contain information for database administrators.

### Use

You choose ► [Performance](#) ► [Additional Functions](#) ► [Display DBA Tables](#) ► in the DBA Cockpit.

## Structure

You see a list of the DBA tables in the database.

### 3.6.7.4 Display USER Views

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you see list the USER views existing in an Oracle database and display their contents.

You choose ► [Performance](#) ► [Additional Functions](#) ► [Display User Views](#) ► in the DBA Cockpit.

#### Structure

The USER views are arranged in a two-column list in alphabetic order of the views' names.

To see more detail, double-click an entry in the table.

The columns displayed in detail depend on the view. For more information, see the Oracle documentation.

### 3.6.7.5 Database Parameters

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you view the active Oracle database parameters and the contents of the `init<SID>.ora` file. You can also see the history of changes to the parameters. You can use this sub-monitor to view parameters on different instances of an Oracle Real Application Cluster (RAC).

Data is retrieved at run time from the database through a query to the views `V$PARAMETER` and `V$SPPARAMETER`. The view `V$SPPARAMETER` shows the current values of the parameters in the Oracle `spfile` but not the current values used by the instance. This view returns NULL values if a server parameter file (`spfile`) is not being used by the instance. You can also check this by looking at the value of the parameter `SPFILE` in the view `V$PARAMETER`.

The view `V$PARAMETER` shows the current values for the parameters used (not the `spfile` values). If a parameter in the database is changed, it is logged in the alert file. This lets us retrieve the history of changes to each parameter.

## Use

- For each instance, you need to create a table called `sap_alert_<Inst_ID>` in order to access the corresponding alert log file data. For this you need to perform the following commands to create this table to access the external alert log file

1. Create the path of the alert log file :

```
[ create directory DIR_1 as 'ALERT_LOG_PATH' ; ]
```

ALERT\_LOG\_PATH contains the path of the alert log file of the required instance.

2. Create the database table corresponding to the above alert log file by issuing the following SQL command:

```
[ CREATE TABLE sap_alert_<INST_ID>
(entry VARCHAR2(2000) ) ORGANIZATION EXTERNAL
(TYPE oracle_loader DEFAULT DIRECTORY DIR_1 ACCESS PARAMETERS
(RECORDS DELIMITED BY NEWLINE
NOBADFILE
NOLOGFILE
NODISCARDFILE
FIELDS TERMINATED BY ' '
MISSING FIELD VALUES ARE NULL
(entry )
) location('ALERT_LOG_FILE_NAME') ); ]
```

Notice the directory DIR\_1 near the top of the above command. Make sure that you provide the file name in the ALERT\_LOG\_FILE\_NAME.

Make sure that the tables match the alert log file path whenever its directory or its name changed.

- To start the sub-monitor, you choose ► [Performance](#) ► [Additional Functions](#) ► [Database Parameters](#) ► in the DBA Cockpit.

## Structure

- [Active Parameters](#)

This tab page displays the parameters that are currently active in the database. It displays the following information:

Column	Description
<i>Instance ID</i>	Instance ID
<i>SID</i>	Name of the RAC instance
<i>Parameter</i>	Name of the active parameter
<i>Parameter value</i>	Value of the parameter
<i>Value in SPFILE</i>	Value in SPFILE (if present)

- [Parameters History](#)



This tab page uses the alert log file to display all changes in database parameters. Choose [Show parameters history](#) to display the following information:

Column	Description
<i>Instance ID</i>	Instance ID
<i>SID</i>	Name of the RAC instance
<i>Parameter</i>	Name of the parameter
<i>Value</i>	Value of the parameter
<i>Timestamp</i>	Timestamp for this value of the parameter
<i>Scope</i>	Indicates whether the parameter change is only temporary, or persistent and in memory
<i>Target instance</i>	RAC instance for which the change applies

- [SPFILE](#)

This tab page displays the contents of the `init<SID>.ora` file. It displays the following information:

Column	Description
<i>SID</i>	Name of the instance
<i>Parameter</i>	Name of the parameter
<i>Value</i>	Value of the parameter
<i>ISpecified</i>	Whether the parameter was specified in the server parameter file
<i>Ordinal</i>	Position - that is, ordinal number - of the parameter value (0 if a server parameter file was not used to start the instance)  Useful only for parameters for which values are lists of strings.

- [Compare Parameter Config.](#)

This tab page only appears for RAC.

## 3.6.7.6 Alert Log

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you check the database message log (also known as the alert log).

### Use

You choose ► *Performance* ► *Additional Functions* ► *Alert Log* ►.

To generate the required display of the message log, you specify the parameters:

- *Select content*
  - *Read all messages*  
All messages are displayed.
  - *Read all msg. w/o logswitch and checkpoint*  
All messages are displayed except log-switch and checkpoint messages.
  - *Read only alerts*  
Only alerts are displayed.
- *Select time*
  - *Entries starting from*  
Enter the date and time from which messages are to be displayed.
  - *All available*  
All available messages are displayed.
- *Max lines to be displayed*  
The display is restricted to the maximum number of lines.

### Structure

The message log is displayed in chronological order.

#### i Note

On Oracle Real Application Cluster (RAC), when you select *Total* under *DB Instances*, a merged message log for all instances is displayed. The sub-monitor adds the column *Instance ID* to the list.

You can use the scrolling functions, such as *Day*, to quickly find the required part of the message log.

Trace files in the message log display are highlighted. You can display the content of a trace file by selecting the relevant row and choosing *Show trace file*.

## Integration

To use this sub-monitor, your system must meet at least one of the following requirements:

- Message log

Access Method to Message Log	Requirement
BRTOOLS using file type <code>alert_log</code>	BRTOOLS must be installed on the database instances. All administrative files, such as <code>servenames</code> , must be maintained correctly.
BRTOOLS using file type <code>alert_log</code> with specified <code>sysid</code>	BRTOOLS must be installed on the database instances. This method is more tolerant of missing administrative file entries than the previous method.
BRTOOLS using file type <code>file</code>	BRTOOLS have to be installed on the database instances. This method is more tolerant of missing administrative file entries than the previous methods.
RFC call and <code>READ</code> dataset	The instance server must also host a SAP application server. The data file must be accessible by the application server.
Message log is read from an external table	The external table must exist, as described below.

- Trace file  
The sub-monitor tries to read the trace file using BRTOOLS. Therefore, you must make sure that BRTOOLS is installed on the database instance and that all administrative files are maintained correctly. For more information, see SAP Note [446172](#).  
Make sure that the trace file has not been deleted if you want to view it.
- The alert log is read from an "external table," which is normally set up by the installation. If the external table has not already been created, you can create it with the script attached to SAP Note [1546456](#).

### 3.6.7.7 Checkpoints

#### Definition

This sub-monitor in the SAP/Oracle Database Monitor displays all checkpoints found in the current `alert_<SID>.log`.

## Use

1. You set the `init.ora` parameter `LOG_CHECKPOINTS_TO_ALERT` to `TRUE` so that the information required by the monitor is written to the alert file.
2. You choose [► Performance ► Additional Functions ► Checkpoints ►](#) in the DBA Cockpit.
3. You make the following entries:
  - *Select time*  
Enter a start date and time for the displayed checkpoints in *Entries starting from* or select *All available* to display all entries.
  - *Max. lines to be displayed*  
Enter the maximum number of lines for the display.

## Structure

This screen contains the following information:

Column	Description
<i>Checkpoint Number</i>	Number of the checkpoint
<i>Start Date</i>	Time the checkpoint ended
<i>Start Time</i>	Time the checkpoint started
<i>End Date</i>	Date the checkpoint ended
<i>End Time</i>	Time the checkpoint ended
<i>Duration (sec)</i>	Duration of the checkpoint in seconds
<i>In Parallel</i>	For parallel checkpoints, the number of checkpoints

### i Note

Parallel checkpoints are indicated by:

- A number in the *in Parallel* column indicating how many parallel checkpoints
- A yellow background color

## 3.6.7.8 Oracle Net

### Definition

This sub-monitor in the SAP/Oracle Database Monitor gives you an overview of the Oracle Net configuration of the Oracle Database

### Use

You choose ► [Performance](#) ► [Additional Functions](#) ► [Oracle Net](#) ► in the DBA Cockpit.

When you start the monitor or when you first switch to a different tab page, a pop-up prompts you for the path name to the `network/admin` directory, for example, `/sapmnt/db_sw/oracle/101_64/software/network/admin/`. As long as you remain inside this area of the monitor, the pop-up no longer appears when you have already entered the correct path name for a particular tab page.

### Structure

Oracle Net consists of the following tab pages:

- [tnsnames.ora](#)
- [sqlnet.ora](#)
- [listener.ora](#)

### Integration

The files listed above on the tab pages are read from an "external table," which is normally set up by the installation. If the external table has not already been created, you can create it with the script attached to SAP Note [1546456](#) 📄.

## 3.6.7.9 User End to End Trace

### Prerequisites

The trace files are written to the directory specified by the Oracle parameter `USER_DUMP_DEST`, normally `$ORACLE_HOME/saptrace/usertrace`.

## ⚠ Caution

When the trace is active, make sure that there is enough space in the file system containing the trace directory.

There are the following restrictions for the trace:

- It is only available as of SAP Kernel 7.1 and Oracle 10.2.0.2.
- The SAP schema users of the database that is, `SAP<SID>`, `SAPR3`, or `SAPSR3` must have the EXECUTE permissions to the package `DBMS_MONITOR`.  
This means that user `SYS` must have executed `GRANT EXECUTE ON DBMS_MONITOR TO SAP<SID>`.

## Context

This function in the SAP/Oracle DBA Cockpit lets you set the trace for individual users on the Oracle database. The Oracle database server writes a trace for all actions that the user executes.

Do not confuse this trace with SAP transaction `ST05`. Only use the trace when SAP/Oracle Development Support requests you to do so for problem diagnosis.

## Procedure

1. Choose **► Performance ► Additional Functions ► User End to End Trace ►** in the DBA Cockpit.
2. Enter a user name in *SAP user*, paying attention to uppercase or lowercase.
3. If you want to include wait information in the trace, select *Tracing Waits*.
4. If you want to include bind information in the trace, select *Tracing Binds*.
5. Choose *Enable Trace* to start the trace.

You can see a list of all users with an active trace.

### i Note

The trace remains active even after a stop and restart of the database. Therefore, you have to explicitly turn it off.

6. To turn off the trace, select it from the list of active traces and choose *Disable Trace*. If required, you can turn off multiple or all traces.

SAP/Oracle Development Support can evaluate the trace files using Oracle command line tools (such as `trcsess`).

## 3.6.7.10 User End to End Stats Trace

### Prerequisites

Trace information is written to the table `v$client_stats` and is lost if the database is stopped.

There are the following restrictions for the trace:

- It is only available as of SAP Kernel 7.1 and Oracle 10.2.0.2.
- The SAP schema users of the database that is, `SAP<SID>`, `SAPR3`, or `SAPSR3` must have the EXECUTE permissions to the package `DBMS_MONITOR`.  
This means that user `SYS` must have executed `GRANT EXECUTE ON DBMS_MONITOR TO SAP<SID>`.

### Context

This function in the SAP/Oracle DBA Cockpit lets you set the trace for the statistics of individual users on the Oracle database.

### Procedure

1. Choose **► Performance ► Additional Functions ► User End to End Stats Trace ►** in the DBA Cockpit.
2. Enter a user name in *SAP user*, paying attention to uppercase or lowercase.
3. Choose *Enable Trace* to start the trace.

You can see a list of all users with an active trace.

#### **i** Note

The trace remains active even after a stop and restart of the database. Therefore, you have to explicitly turn it off.

4. To view the results of the trace, select the appropriate user in the column *Display stats*.
5. To turn off the trace, select it from the list of active traces and choose *Disable Stats*. If required, you can turn off multiple or all traces.

## 3.6.7.11 Program End to End Trace

### Prerequisites

The trace files are written to the directory specified by the Oracle parameter `USER_DUMP_DEST`, normally `$ORACLE_HOME/saptrace/usertrace`.

#### ⚠ Caution

When the trace is active, make sure that there is enough space in the file system containing the trace directory.

There are the following restrictions for the trace:

- It is only available as of SAP Kernel 7.1 and Oracle 10.2.0.2.
- The SAP schema users of the database that is, `SAP<SID>`, `SAPR3`, or `SAPSR3` must have the `EXECUTE` permissions to the package `DBMS_MONITOR`.  
This means that user `SYS` must have executed `GRANT EXECUTE ON DBMS_MONITOR TO SAP<SID>`.

### Context

This function in the SAP/Oracle DBA Cockpit lets you set the SQL trace for individual programs on the Oracle database server.

Do not confuse this trace with SAP transaction `ST05`. Only use the trace when SAP/Oracle Development Support requests you to do so for problem diagnosis.

### Procedure

1. Choose **► Performance ► Additional Functions ► Program End to End Trace ►** in the DBA Cockpit.
2. Enter a program name in *Program*, paying attention to uppercase or lowercase.
3. If you want to include wait information in the trace, select *Tracing Waits*.
4. If you want to include bind information in the trace, select *Tracing Binds*.
5. Choose *Enable Trace* to start the trace.

You can see a list of all programs with an active trace.

#### i Note

The trace remains active even after a stop and restart of the database. Therefore, you have to explicitly turn it off.



6. To turn off the trace, select it from the list of active traces and choose *Disable Trace*. If required, you can turn off multiple or all traces.

SAP/Oracle Development Support can evaluate the trace files using Oracle command line tools (such as `trcsess`).

## 3.6.7.12 Program End to End Stats Trace

### Prerequisites

Trace information is written to the table `v$client_stats` and is lost if the database is stopped.

There are the following restrictions for the trace:

- It is only available as of SAP Kernel 7.1 and Oracle 10.2.0.2.
- The SAP schema users of the database - that is, `SAP<SID>`, `SAPR3`, or `SAPSR3` - must have the EXECUTE permissions to the package `DBMS_MONITOR`.

This means that user `SYS` must have executed `GRANT EXECUTE ON DBMS_MONITOR TO SAP<SID>`.

### Context

This function in the SAP/Oracle DBA Cockpit lets you set the trace for the statistics of individual programs on the Oracle database.

### Procedure

1. Choose **► Performance ► Additional Functions ► Program End to End Stats Trace ►** in the DBA Cockpit.
2. Enter a program name in *Program*, paying attention to uppercase or lowercase.
3. Choose *Enable Trace* to start the trace.

You can see a list of all programs with an active trace.

#### **i** Note

The trace remains active even after a stop and restart of the database. Therefore, you have to explicitly turn it off.

4. To view the results of the trace, select the appropriate program in the column *Display stats*.
5. To turn off the trace, select it from the list of active traces and choose *Disable Stats*. If required, you can turn off multiple or all traces.

## 3.6.7.13 Bundle Patches

### Definition

This sub-monitor in the SAP/Oracle Database Monitor shows the installed bundle patches (if any) on the system. For more information, see SAP note [1727180](#).

You choose ► [Performance](#) ► [Additional Functions](#) ► [Bundle Patches](#) ▾.

### Structure

Column	Description
<a href="#">Action Time</a>	Time of patch action
<a href="#">Action</a>	Patch action
<a href="#">Namespace</a>	Namespace
<a href="#">Version</a>	Patch version
<a href="#">Bundle series</a>	Type of bundle patch
<a href="#">Comments</a>	Further information

## 3.7 Space (Oracle)

### Use

You can use the [User](#) menu in the DBA Cockpit to monitor the space in your Oracle database:

- Space overview
- Database overview
- Users overview
- Users detailed analysis
- Tablespaces overview
- Tablespaces detailed analysis
- Segments overview

- Segments detailed analysis
- Segments detailed analysis (aggregated)
- Collector logs
- BW analysis

## 3.7.1 Space Overview

### Definition

This screen in the DBA Cockpit gives you an overview of the space in your Oracle database. It uses a collector job to periodically retrieve data from the cluster table MONI.

You choose ► [Space](#) ► [Space Overview](#) ► in the DBA Cockpit to call up the space overview.

### Structure

The fields are as follows:

Field	Description
<i>Database</i>	
<i>Name</i>	Database name <i>DEFAULT</i> means the local database.
<i>DB system</i>	<i>ORA</i> for Oracle
<i>Size</i>	Size of the database in GB
<i>Free size</i>	Free space in the database in GB
<i>Used</i>	Percentage used space in database Equal to $100 * \text{Free size} / \text{Size}$ .
<i>Users</i>	
<i>Total number</i>	Number of database users
<i>Maximum size</i>	Maximum size used by a user The user name is in brackets.
<i>Maximum # segments</i>	Maximum number of segments used by a user The user name is in brackets.

Field	Description
<i>Tablespaces</i>	
<i>Total number</i>	Total number of tablespaces
<i>Maximum size</i>	Size of the biggest tablespace The tablespace name is in brackets.
<i>Minimum free space</i>	Minimum free space not used by a tablespace The tablespace name is in brackets.
<i>Segments</i>	
<i>Number</i>	Number of the objects <i>Tables, Indexes, Others</i>
<i>Size</i>	Total size of the objects in MB
<i>More 100 ext.</i>	Number of objects with more than 100 extents
<i>Compressed</i>	Number of compressed objects
<i>No logging</i>	Number of objects with "no logging"
<i>Critical</i>	Number of critical objects

## 3.7.2 Space Statistics Collector

### Use

The Space Statistic Collector collects data for the space monitor in the DBA Cockpit.

### Integration

A standard background job, `SAP_COLLECTOR_FOR_PERFMONITOR`, starts the collector. The data is physically stored in cluster table `MONI`.

### Prerequisites

You must make sure that the collector is set up as described in [SAP Note 868063](#).

## Features

The collector is based on program RSORACOL. This report must be included in table TCOLL as one of the components of standard SAP job SAP\_COLLECTOR\_FOR\_PERMONITOR with the following parameters:

- RNAME: RSORACOL
- RENUM: 1
- SYTYP: S (or C if 620)
- ALTSY
- DAYOW: XXXXXXX
- TIMOD: XXXXXXXXXXXXXXXXXXXXXXX

The setup process as described in "Prerequisites" above creates at least one entry in table ORA\_MON\_DBCON and many entries in table DB02\_COLL\_PLAN:

- The table ORA\_MON\_DBCON provides a list with the connection names of the databases to be monitored. The databases can be local or remote, running SAP or not. The only requirement is that the connection names must match those stored in table DBCON. The local database is always called *DEFAULT*.
- The table DB02\_COLL\_PLAN provides a list of collector modules - ABAP function modules that collect data - with their specific settings such as module name, module\_id, schedule, MONI key where data is stored, dependencies, and so on.  
All these function modules currently belong to SAP function group SDBORA3. There is normally one collector module for each submonitor in the DBA Cockpit.

### Example

The collector module US-GE-MD (SMON\_ID=200) collects the data displayed in sub-monitor [Space Statistics > User > Overview](#).

A normal collection job runs as follows:

1. The standard SAP job SAP\_COLLECTOR\_FOR\_PERMONITOR runs hourly and starts the program RSORACOL.
2. The program RSORACOL reads the table ORA\_MON\_DBCON to:
  1. Get the connection names of the databases to be monitored.
  2. Check whether the database connection is active or inactive
3. For each database, RSORACOL executes a set of collector modules according to their schedules and according to settings stored in table DB02\_COLL\_PLAN.  
If required, RSORACOL establishes a native connection to remote databases.
4. As collector modules finish, they store collected data in cluster table MONI under a specific monikey. Program RSORACOL then immediately writes a log entry to table DB02\_COLL\_LOG.
5. The previous two steps are repeated for each database to be monitored.

There is a logging mechanism in the collector program. Every time a collection module completes a collection and exports it to MONI a log entry is written to table DB02\_COLL\_LOG. The log entry contains a time stamp, submonitor ID, return code, and other information.

## Activities

### Configuration

You can configure the space statistics collector as follows:

- You add databases to the monitoring list or delete them as described in [SAP Note 868063](#). The local database is always called *DEFAULT* and does not appear in the F4 help.
- You can change database settings in the table `ORA_MON_DBCON` except `CON_NAME`:

Column	Description
<code>CON_NAME</code>	Database connection name, as stored in table <code>DBCON</code>
<code>STATUS</code>	Status of the database monitoring
<code>MAX_RUNTIME</code>	Maximum runtime in seconds desired for collector modules Default: 1800 seconds
<code>MAX_UPLOAD</code>	Maximum size in KB desired to be written to <code>MONI</code> per collector module Default: 1,000,000 KB
<code>LOG_LIFE</code>	Time in days desired to retain logs Default: 100 days
<code>DESCRIPTION</code>	Short text

- We recommend that you do **not** normally modify table `DBO2_COLL_PLAN`, which contains the collector module settings. If required, you can modify certain fields like `MONIFILL`, `SCH_DATE`, `SCH_TIME`, or `PARAM` but we recommend you to raise an SAP support message before doing so. The table `DBO2_COLL_PLAN` looks as follows:

Column	Description
<code>CON_NAME</code>	Database connection name, as stored in table <code>DBCON</code>
<code>SMON_ID</code>	Sub-monitor ID for this collector module
<code>SMON_NAME</code>	Sub-monitor name for this collector module
<code>FUNCNAME</code>	Function module name
<code>MONIFILL</code>	<code>MONI</code> filler, part of the <code>MONI</code> key

Column	Description
RANK	Rank  Relevant for sub-monitors that are organized hierarchically.
STATUS	Collector module status
SCH_DATE	Schedule date
SCH_TIME	Schedule time
PROCESS_TYPE	Process type  This differs according to the data source. For example, data from Oracle views is process type snapshot, S.
SOURCE_SMON	Source sub-monitor id  Some collector modules gather data from other collector modules, which means there can be dependencies between monitors.
PARAM	Some collector modules accept or require parameters.  For example, the collector module for top table sizes sets the length of the ranking.

### Remote Monitoring

So that you can use the space monitor to monitor Oracle databases remotely, the collector program can also collect data from remote databases using a native connection.

Make sure that you meet the following prerequisites to enable data collection from remote databases:

1. The `tnsnames.ora` files attached to local Oracle clients contain a specific entry for the remote database. If your local system does not run Oracle, you need to at least install the Oracle client software and then adapt the `tnsnames.ora` files.
2. The native connection to the remote database exists in table `DBCON` and it works without problems.
3. The database connection exists in table `ORA_MON_DBCON`.  
Do **not** make this entry manually. For more information about setting up collectors for new databases, see "Prerequisites" above.

### Troubleshooting

If the space monitor in the DBA Cockpit is not up-to-date or contains no data, see [SAP Note 1002840](#). For further support, you can create a message in component `BC-CCM-MON-ORA`.

## 3.7.3 Database Overview

### Definition

This monitor in the SAP/Oracle Database Monitor lets you display the space usage on the database, including history information. It uses a collector job to periodically retrieve data from the cluster table MONI.

### Use

You choose ► [Space](#) ► [Database](#) ► [Overview](#) ► in the DBA Cockpit.

You can choose [Without Duplicates](#) on the [History](#) tab to remove any duplicates in the data.

### Structure

The fields on the tab [Main data](#) are self-explanatory.

The tab [History](#) contains the following information for the sub-tab [Days](#) display (the columns are analogous for the sub-tabs [Weeks](#) and [Months](#)):

Column	Description
<a href="#">Date</a>	Date of last analysis
<a href="#">Name</a>	Database name
<a href="#">Size (Gb)</a>	Database size in GB
<a href="#">Chg. Size/day</a>	Change of database size per day in GB
<a href="#">Free (Gb)</a>	Database free space in GB
<a href="#">Chg. Free/day</a>	Change in database free space per day in GB
<a href="#">Used (%)</a>	Percentage used space in the database
<a href="#">Chg. Used/day</a>	Change in database used space per day
<a href="#">Users</a>	Number of database users
<a href="#">Chg. Users/day</a>	Change in number of users per day
<a href="#">Tablespaces</a>	Number of tablespaces
<a href="#">Chg. Tablespaces/day</a>	Change in number of tablespaces per day



Column	Description
<i>Segments</i>	Number of segments
<i>Chg. Segments/day</i>	Change in number of segments per day
<i>Data (%)</i>	Percentage of space containing data
<i>Chg. Data/day</i>	Change in data space per day
<i>Temp (%)</i>	Percentage of space containing temporary data
<i>Chg. Temp/day</i>	Change in temporary space per day
<i>Tables (%)</i>	Percentage of space containing tables
<i>Chg. Tables/day</i>	Change in number of tables per day
<i>Indexes (%)</i>	Percentage of space containing indexes
<i>Chg. Indexes/day</i>	Change in number of indexes per day
<i>Others (%)</i>	Percentage of space occupied by other objects (that is, objects other than tables and indexes, such as lobs, clusters, or rollbacks)
<i>Chg. Others/day</i>	Change in number of other objects per day

Each sub-tab on the *History* tab contains an average display separately at the top followed by detailed displays for each time period.

## 3.7.4 Users Overview

### Definition

This monitor in the SAP/Oracle Database Monitor lets you display the database users, including history information. It uses a collector job to periodically retrieve data from the cluster table MONI.

### Use

You choose ► [Space](#) ► [Users](#) ► [Overview](#) in the DBA Cockpit.

## Structure

This screen contains the following information:

- The tab *Main Data*:

Column	Description
<i>User name</i>	User name
<i>User id</i>	User ID
<i>Size (MB)</i>	Size of data assigned to user in MB
<i># Segments</i>	Number of segments owned by the user
<i># Tables</i>	Number of tables owned by the user
<i># Indexes</i>	Number of indexes owned by the user
<i># Extents</i>	Number of extents owned by the user
<i>Def. Tablespace</i>	Name of default tablespace for user
<i>Temp. tablespace</i>	Name of temporary tablespace for user
<i>Account status</i>	Status of user account
<i>Profile</i>	Name of user profile
<i>Creation Date</i>	Date user account was created
<i>Creation Time</i>	Time user account was created

- The tab *Summary History* contains the following information for the sub-tab *Days* (the columns are analogous for the sub-tabs *Weeks* and *Months*):

Column	Description
<i>Begin analysis date</i>	Date when the analysis begins
<i>End analysis date</i>	Date when the analysis ends
<i>User name</i>	User name
<i>User id</i>	User ID
<i>Size (MB)</i>	Size in MB
<i>Chg. Size/day</i>	Average daily change in size in MB

Column	Description
<i># Segments</i>	Number of segments
<i>Chg. Segm/day</i>	Average daily change in number of segments
<i># Tables</i>	Number of tables
<i>Chg. Tables/day</i>	Average daily change in number of tables
<i># Indexes</i>	Number of indexes
<i>Chg. Indexes/day</i>	Average daily change in number of indexes
<i># Extents</i>	Number of extents
<i>Chg. Extents/day</i>	Average daily change in number of extents

## 3.7.5 Users Detailed Analysis

### Definition

This monitor in the SAP/Oracle Database Monitor lets you display detailed information about the database users, including history information. It uses a collector job to periodically retrieve data from the cluster table MONI.

### Use

You choose ► *Space* ► *Users* ► *Detailed Analysis* ► in the DBA Cockpit.

The system prompts you to enter selection criteria for the users.

You enter selection criteria to restrict the display to the required users.

When the main grid is populated, you can double-click any row to display it in a more concise form in the *Main data* tab.

## Structure

This screen contains the following information:

- The tab *Main data*:

Column	Description
<i>User name</i>	User name
<i>User id</i>	User ID
<i>Password</i>	Password
<i>Account status</i>	Status of the user account
<i>Def. tablespace</i>	Default tablespace name for user
<i>Temp. tablespace</i>	Temporary tablespace name for user
<i>Profile</i>	Profile name of user
<i>Resource consumer gr.</i>	Resource consumption group of user
<i>External name</i>	External name of user
<i>Creation date</i>	User creation date
<i>Creation time</i>	User creation time
<i>Acc. lock date</i>	Date user account was locked
<i>Acc. expiry date</i>	Date user account expires

- The tab *History* contains the following information for the sub-tab *Days* (the columns are analogous for the sub-tabs *Weeks* and *Months*):

### **i** Note

Each sub-tab on the *History* tab contains an average display separately at the top followed by detailed displays for each time period. The table below shows the average display. The detailed display is very similar.

Column	Description
<i>Date</i>	Analysis date
<i>User name</i>	User name
<i>User id</i>	User ID

Column	Description
<i>Size (MB)</i>	Size in MB
<i>Chg. Size/day</i>	Average daily change in size in MB
<i># Segments</i>	Number of segments
<i>Chg. Segm/day</i>	Average daily change in number of segments
<i># Tables</i>	Number of tables
<i>Chg. Tables/day</i>	Average daily change in number of tables
<i># Indexes</i>	Number of indexes
<i>Chg. Indexes/day</i>	Average daily change in number of indexes
<i># Extents</i>	Number of extents
<i>Chg. Extents/day</i>	Average daily change in number of extents

## 3.7.6 Tablespaces Overview

### Definition

This monitor in the SAP/Oracle Database Monitor lets you display the tablespaces in the database, including history information. It uses a collector job to periodically retrieve data from the cluster table MONI.

### Use

You choose [Space](#) > [Tablespaces](#) > [Overview](#) in the DBA Cockpit.

## Structure

This screen contains the following information:

- Tab *Main Data*:

Column	Description
<i>Tablespace name</i>	Name of tablespace
<i>Size (MB)</i>	Size of tablespace in MB
<i>Free (MB)</i>	Size of free space in the tablespace in MB
<i>Used (%)</i>	Percentage of available space used in the tablespace
<i>Autoextend</i>	Autoextend option
<i>Total size (MB)</i>	Total size of tablespace in MB
<i>Sum of all free fragments</i>	Total size of all free fragments in MB
<i>Total used (%)</i>	Percentage of the total space used in tablespace
<i># Files</i>	Number of files in tablespace
<i># Segments</i>	Number of segments in tablespace
<i># Extents</i>	Number of extents in tablespace
<i>Status</i>	Status of tablespace
<i>Contents</i>	Type of tablespace data
<i>Compress for</i>	Type of compression (as of Oracle 11.2)
<i>Encrypted</i>	Tablespace encryption (as of Oracle 11.2)
<i>Encrypt algorithm</i>	Encryption algorithm (as of Oracle 11.2)

- Tab *Storage Management*:

Column	Description
<i>Tablespace name</i>	Tablespace name
<i>Extent management</i>	Extent management tracking, <i>DICTIONARY</i> or <i>LOCAL</i>
<i>Allocation type</i>	Type of extent allocation for tablespace
<i>SSM</i>	Segment space management tracking, <i>AUTO</i> or <i>MANUAL</i>

Column	Description
<i>Block size (KB)</i>	Tablespace block size in KB
<i>Init. extent (MB)</i>	Default initial extent size of tablespace in MB
<i>Next extent (MB)</i>	Next extent size of tablespace in MB
<i>Min. extents</i>	Minimum number of extents for tablespace
<i>Max. extents</i>	Maximum number of extents for tablespace
<i>PCT increase (%)</i>	Percentage increase of extents for tablespace

- Tab *Data/Temp. Files* (both sub-tabs have the same columns):

Column	Description
<i>File name</i>	File name
<i>File id</i>	ID of the database data file
<i>Tablespace name</i>	Tablespace name to which file belongs
<i>Size (MB)</i>	Size of file in MB
<i># Blocks</i>	Number of Oracle blocks in the file
<i>Status</i>	Status of file, <i>INVALID</i> or <i>AVAILABLE</i>
<i>Rel. file number</i>	File number in the tablespace
<i>Autoextensible</i>	Autoextensible indicator for tablespace
<i>Maxsize (MB)</i>	Maximum size of file in MB
<i>Maxblocks</i>	Maximum size of file in Oracle blocks
<i>Increment by</i>	Default increment for autoextension
<i>User size (MB)</i>	Size of useful portion of file in MB
<i>User blocks</i>	Size of useful portion of file in Oracle blocks

- Tab *Freespace statistics*:
  - Sub-tab *Overview*:

Column	Description
<i>Tablespace name</i>	Tablespace name

Column	Description
<i>Max free (MB)</i>	Largest free fragment in tablespace in MB
<i>Total free (MB)</i>	Total size of all free fragments in tablespace in MB
<i># Fragments</i>	Number of fragments in tablespace
<i>Max. next extent</i>	Maximum number of next extents in tablespace
<i>Critical objects 1</i>	Number of segments in tablespace that cannot allocate more than P extents
<i>Critical objects 2</i>	Number of segments in tablespace that cannot allocate more than P' but that can allocate more than P extents

### i Note

For the last two rows in the above table the following applies:

- P is the customizable parameter attached to SMON\_ID=509 and SMON\_ID=510 in table DB02\_COLL\_PLAN. Its default value is 2.
- P' is defined by the formula  $P' = 3 * P$  and is calculated during runtime. Therefore, its default value is 6.  
If a segment can allocate more than P' extents, *Critical objects 1* and *Critical objects 2* are not filled.

- Sub-tab *Free fragments*:

Column	Description
<i>Tablespace name</i>	Tablespace name
<i>File Id</i>	ID number of file containing the fragment
<i>Block Id</i>	Starting block number of file containing the fragment
<i>Size (MB)</i>	Size of fragment in MB
<i># Blocks</i>	Size of fragment in Oracle blocks
<i>Rel. file number</i>	Relative number of the file containing the extent

- Tab *Summary history*:

Column	Description
<i>Begin date</i>	Start date for analysis



Column	Description
<i>End date</i>	End date for analysis
<i>Tablespace name</i>	Tablespace name
<i>Size (MB)</i>	Size of tablespace in MB
<i>Chg. size/day</i>	Change in used size of tablespace per day
<i>Free (MB)</i>	Free size in MB
<i>Chg. free/day</i>	Change in free space in tablespace per day
<i>Used (%)</i>	Percentage used space in tablespace
<i>Chg. Used/day</i>	Change in used space in tablespace per day
<i>Autoextend</i>	Autoextend flag, <i>YES</i> or <i>NO</i>
<i>Total size (MB)</i>	Total size of tablespace in MB
<i>Change of total size per day (MB/day)</i>	Change in total size of tablespace per day
<i>Sum of all free fragments (MB)</i>	Total of all free fragments in tablespace in MB
<i>Chg. free/day (MB/day)</i>	Change in free size in tablespace per day
<i>Total used (%)</i>	Percentage of total space used in tablespace
<i>Chg. Total used /day (%/day)</i>	Change in percentage of total space used in tablespace per day
<i># Files</i>	Number of files in tablespace
<i>Chg. #files/day</i>	Change in number of files in tablespace per day
<i># Segments</i>	Number of segments in tablespace
<i>Chg. #segments/day</i>	Change in number of segments in tablespace per day
<i># Extents</i>	Number of extents
<i>Chg. #extents/day</i>	Change in number of extents in tablespace per day

## 3.7.7 Tablespaces Detailed Analysis

### Definition

This monitor in the SAP/Oracle Database Monitor lets you display detailed information about the tablespaces in the database, including history information. It uses a collector job to periodically retrieve data from the cluster table MONI.

### Use

You choose ► [Space](#) ► [Tablespaces](#) ► [Detailed Analysis](#) ▾ in the DBA Cockpit.

The system prompts you to enter selection criteria for the tablespaces. You enter selection criteria to restrict the display to the required tablespaces. When the main grid is populated, you can double-click any row to display it in a more concise form in the [Main data](#) tab.

### Structure

This screen contains the following information:

- The tab [Main data](#):

Column	Description
<a href="#">Tablespace name</a>	Tablespace name
<a href="#">Status</a>	Tablespace status, <a href="#">ONLINE</a> , <a href="#">OFFLINE</a> , or <a href="#">READ</a> .
<a href="#">Contents</a>	Tablespace contents, <a href="#">PERMANENT</a> or <a href="#">TEMPORARY</a>
<a href="#">Logging</a>	Default logging attribute for tablespace
<a href="#">Forced logging</a>	Forced logging mode for tablespace
<a href="#">Extent management</a>	Extent management tracking, <a href="#">DICTIONARY</a> or <a href="#">LOCAL</a>
<a href="#">Allocation type</a>	Type of extent allocation for tablespace
<a href="#">Plugged in</a>	Specifies whether a tablespace is transported to an instance
<a href="#">Segment space mgment</a>	Segment space management tracking, <a href="#">AUTO</a> or <a href="#">MANUAL</a>
<a href="#">Def. table compression</a>	Default table compression mode, <a href="#">DISABLED</a> or <a href="#">ENABLED</a>

Column	Description
<i>Size (MB)</i>	Size of tablespace in MB
<i>Free (MB)</i>	Size of free space in tablespace in MB
<i>Block size (KB)</i>	Tablespace in KB
<i>Initial extent (MB)</i>	Default initial extent size of tablespace in MB
<i>Next extent (MB)</i>	Default next extent size of tablespace in MB
<i>Min. extents</i>	Default minimum number of extents for tablespace
<i>Max. extents</i>	Default maximum number of extents for tablespace
<i>PCT increase (%)</i>	Default percent increase of extent size for tablespace
<i>Min. Extent length (MB)</i>	Minimum extent size for tablespace in MB
<i>Compress for</i>	Type of compression (as of Oracle 11.2)
<i>Encrypted</i>	Tablespace encryption (as of Oracle 11.2)
<i>Encrypt algorithm</i>	Encryption algorithm (as of Oracle 11.2)

- The tab *Files*:

Column	Description
<i>File name</i>	File name
<i>File id</i>	ID of the database data file
<i>Tablespace name</i>	Tablespace name to which file belongs
<i>Size (MB)</i>	Size of file in MB
<i>#Blocks</i>	Number of blocks in file
<i>Status</i>	Status of file, <i>INVALID</i> or <i>AVAILABLE</i>
<i>Relative file number</i>	File number in the tablespace
<i>Autoextend</i>	Autoextensible indicator for tablespace
<i>Maxsize (MB)</i>	Maximum size of file in MB
<i>Maxblocks</i>	Maximum size of file in Oracle blocks
<i>Increment by</i>	Default increment for autoextension

Column	Description
<i>User size (MB)</i>	Size of useful portion of file in MB
<i>User blocks</i>	Size of useful portion of file in Oracle blocks

- The tab *Freespace*:

Column	Description
<i>Tablespace name</i>	Tablespace name to which segment belongs
<i>File id</i>	ID of database data file containing the fragment
<i>Block id</i>	Starting block number of file containing the fragment
<i>Free (MB)</i>	Size of fragment in MB
<i># Blocks</i>	Size of fragment in Oracle blocks
<i>Relative file number</i>	File number in tablespace containing the extent

- The tab *History* contains the following information for the sub-tab *Days* (the columns are analogous for the sub-tabs *Weeks* and *Months*):

#### **i** Note

Each sub-tab on the *History* tab contains an average display separately at the top followed by detailed displays for each time period. The table below shows the average display. The detailed display is very similar.

Column	Description
<i>Date</i>	Date of analysis
<i>Tablespace name</i>	Name of tablespace
<i>Size (MB)</i>	Tablespace size in MB
<i>Maxsize (MB)</i>	Maximum size of tablespace in MB
<i>Free (MB)</i>	Size of free space in tablespace in MB
<i>Maxblocks</i>	Maximum size of tablespace in Oracle blocks
<i>Increment by</i>	Default increment for autoextension
<i>User size (MB)</i>	Total space available for data in the tablespace or datafile in MB

Column	Description
<i>Autoextend</i>	Autoextensible indicator for tablespace
<i>User blocks</i>	Number of blocks available for data in the tablespace
<i>Block id</i>	Starting block number of file containing tablespace
<i>Total free (MB)</i>	Total size of all free fragments in tablespace in MB
<i>Chg. free/day (MB/day)</i>	Change in free size in tablespace per day
<i>Total used (%)</i>	Percentage of total space used in tablespace
<i>Chg. Total used/day (%/day)</i>	Change in percentage of total space used in tablespace per day
<i># Files</i>	Number of files in tablespace
<i>Chg. #files/day</i>	Change in number of files in tablespace per day
<i># Segments</i>	Number of segments in tablespace
<i>Chg. #segments/day</i>	Change in number of segments in tablespace per day
<i># Extents</i>	Number of extents
<i>Chg. #extents/day</i>	Change in number of extents in tablespace per day

## 3.7.8 Segments Overview

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you display the segments in the database, including history information. It uses a collector job to periodically retrieve data from the cluster table MONI.

### Use

You choose ► [Space](#) ► [Segments](#) ► [Overview](#) ► in the DBA Cockpit.

## Structure

This screen contains the following information:

- [Main Data](#) shows all segments

Column	Description
<a href="#">Type</a>	Segment type
<a href="#">Size (MB)</a>	Segment size in MB
<a href="#">Avg. Size (MB)</a>	Average segment size in MB
<a href="#"># Segments</a>	Number of segments
<a href="#"># Extents</a>	Number of extents
<a href="#">More 100 Extents</a>	Number of segments of the specified segment type with more than 100 extents
<a href="#">Compressed</a>	Number of segments of the specified segment type that are compressed to reduce disk use
<a href="#">No Logging</a>	Number of segments of the specified segment type that do not log creation or modification in the redo log file
<a href="#">Not created</a>	Number of objects that have not yet been created on the database (as of Oracle 11.2)

- [Top Sizes](#) and [Top Extents](#) (the columns are identical)  
[Top Sizes](#) shows the segments with the largest sizes together with their space allocation fields.  
[Top Extents](#) shows the segments with the highest number of extents

Column	Description
<a href="#">Owner</a>	Segment owner
<a href="#">Name</a>	Segment name
<a href="#">Partition</a>	Partition name
<a href="#">Type</a>	Segment type
<a href="#">Tablespace</a>	Tablespace name of segment
<a href="#">Size (MB)</a>	Segment size in MB
<a href="#"># Extents</a>	Number of extents
<a href="#"># Blocks</a>	Number of blocks

Column	Description
<i>Next Extent (MB)</i>	Size of next extent in MB

- *Critical Objects* shows objects where the space situation is critical:
  - *Space Critical Objects*

Column	Description
<i>Owner</i>	Segment owner
<i>Name</i>	Segment name
<i>Partition</i>	Partition name
<i>Type</i>	Segment type
<i>Tablespace</i>	Tablespace name of segment
<i>Size (MB)</i>	Segment size in MB
<i>Chg. Size/month</i>	Change in size per month
<i># Extents</i>	Number of extents
<i># Blocks</i>	Number of blocks
<i>Next Extent (MB)</i>	Size of next extent in MB

- *Extents Critical Objects*

Column	Description
<i>Owner</i>	Segment owner
<i>Name</i>	Segment name
<i>Partition</i>	Partition name
<i>Type</i>	Segment type
<i>Tablespace</i>	Tablespace name of segment
<i>Size (MB)</i>	Segment size in MB
<i># Extents</i>	Number of extents
<i>Max Extents</i>	Maximum number of extents
<i>Alert Type</i>	Type of alert

- *Summary History* shows the history of changes to segments - the sub-tab *Days* is shown below (the columns are analogous for *Weeks* and *Months*)

<b>Column</b>	<b>Description</b>
<i>Begin date</i>	Start date for analysis
<i>End date</i>	End date for analysis
<i>Type</i>	Segment type
<i>Size (MB)</i>	Segment size in MB
<i>Chg. Size/day</i>	Change in segment size per day
<i>Avg. Size (MB)</i>	Average segment size in MB
<i>Chg. Avg. Size/Day</i>	Change in average segment size per day
<i># Segments</i>	Number of segments
<i>Chg. Segments/Day</i>	Change in number of segments per day
<i>#Extents</i>	Number of extents
<i>Chg. Extents/day</i>	Change in number of extents per day
<i>More 100 Extents</i>	Number of segments of the specified segment type that have more than 100 extents
<i>Chg. More 100 Extents/day</i>	Change in number of segments of the specified segment type that have more than 100 extents per day
<i>Compressed</i>	Number of segments of the specified segment type that are compressed to reduce disk use
<i>Chg. Compressed/Day</i>	Change in number of segments of the specified segment type that are compressed to reduce disk use per day
<i>Nologging</i>	Number of segments of the specified segment type that do not log creation or modification in the redo log file
<i>Chg. Nologging/day</i>	Change in number of segments of the specified segment type that do not log creation or modification in the redo log file per day



## 3.7.9 Segments Detailed Analysis

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you display selected segments in the database, including history information.

For the *History* tab, it uses a collector job to periodically retrieve data from the cluster table MONI. For the other tabs, it gathers data with an SQL statement in real time.

### Use

You choose ► *Space* ► *Segments* ► *Detailed Analysis* ► in the DBA Cockpit.

The system prompts you to enter selection criteria for the segments. You enter selection criteria to restrict the display to the required segments. When the main grid is populated, you can double-click any row to display it in a more concise form in the *Main data* tab.

### Structure

This screen contains the following information:

- *Main data*:

Column	Description
<i>Owner</i>	Owner name
<i>Segment Name</i>	Segment name
<i>Partition Name</i>	Partition name
<i>Type</i>	Segment type
<i>Tablespace</i>	Tablespace for segment
<i>Header File</i>	Number of header files
<i>Header Block</i>	Number of header blocks
<i>Freelists</i>	Number of process freelists allocated in this segment
<i>Freelist Group</i>	Number of freelist groups allocated in this segment
<i>Relative FNO</i>	Relative number of the file containing the segment header

Column	Description
<i>Buffer Pool</i>	Buffer pool
<i>Size [MB]</i>	Size in MB
<i>Extents</i>	Number of extents
<i>Blocks</i>	Number of blocks
<i>Initial Extents [MB]</i>	Size in MB of the initial extent of the segment
<i>Next Extents [MB]</i>	Size in MB of the next extent to be allocated to the segment
<i>Min Extents</i>	Number of minimum extents allowed in the segment
<i>Max Extents</i>	Number of maximum extents allowed in the segment
<i>PCT increase</i>	Percent increase in the size of the next extent to be allocated
<i>Compressed</i>	Number of segments of the specified segment type that are compressed to reduce disk use
<i>NoLogging</i>	Number of segments of the specified segment type that do not log creation or modification in the redo log file
<i>Compress for</i>	Type of compression (as of Oracle 11.2)

- *Tables + Indexes* and *Partitions* (the columns are identical):

Column	Description
<i>Owner</i>	Owner name
<i>Segment</i>	Segment name
<i>Partition</i>	Partition name
<i>Type</i>	Segment type
<i>Tablespace</i>	Tablespace for segment
<i>Size (MB)</i>	Segment size in MB
<i># Extents</i>	Number of extents
<i># Blocks</i>	Number of blocks
<i>Initial Extents [MB]</i>	Size in MB of the initial extent of the segment

Column	Description
<i>Next Extents [MB]</i>	Size in MB of the next extent to be allocated to the segment
<i>Min. Extents</i>	Minimum number of extents allowed in the segment
<i>Max. Extents</i>	Maximum number of extents allowed in the segment
<i>Pct. Increase</i>	Percent increase in the size of the next extent to be allocated
<i>Buffer Pool</i>	The default buffer pool to be used for segments blocks

- The tab *Extents*:

Column	Description
<i>Owner</i>	Owner name
<i>Segment</i>	Segment name
<i>Partition</i>	Partition name
<i>Type</i>	Segment type
<i>Tablespace</i>	Tablespace for segment
<i>Extent Id</i>	Extent ID
<i>File Id</i>	File ID
<i>Block Id</i>	Block ID
<i>Bytes</i>	Number of bytes
<i>Blocks</i>	Number of blocks
<i>Relative FNO</i>	Relative number of the file containing the segment header

- The tab *Storage*:  
The *Main Data* sub-tab shows the following information:

Column	Description
<i>Space</i>	
<i>Allocated space (Kb)</i>	Allocated space in KB
<i>Allocated blocks</i>	Number of allocated blocks

Column	Description
<i>Allocated extents</i>	Number of allocated extents
<i>Used blocks</i>	Number of blocks used
<i>Never used blocks</i>	Number of blocks never used
<i>Free in used blocks (%)</i>	Free space in used blocks as percentage
<i>Total free space (%)</i>	Total free space as percentage
<i>Block structure</i>	
<i>Block Size (Kb)</i>	Size of block in KB
<i>PCT free (%)</i>	Percentage free in block
<i>PCT used (%)</i>	Percentage used in block
<i>Transactions initial</i>	Number of initial transactions
<i>Transactions maximum</i>	Number of maximum transactions
<i>Header minimum (byte)</i>	Minimum header size in bytes
<i>Extent structure</i>	
<i>Initial extent (Kb)</i>	Size of initial extent in Kb
<i>Next extent (Kb)</i>	Size of next extent in Kb
<i>Min extents</i>	Minimum number of extents
<i>Max extents</i>	Maximum number of extents
<i>PCT increase (%)</i>	Percentage increase in extents
<i>Rows</i>	
<i>Total</i>	Total number of rows
<i>Chained</i>	Number of chained rows
<i>Avg. length (Bytes)</i>	Average row length in bytes
<i>Avg. length + header (Bytes)</i>	Average row length including header in bytes
<i>Avg. initial length (Bytes)</i>	Average initial row length in bytes

You can also view this information graphically in the *Histogram* sub-tab.

- The tab *History* contains the following information for the sub-tab *Days* (the columns are analogous for the sub-tabs *Weeks* and *Months*):

#### **i** Note

Each sub-tab on the *History* tab contains an average display separately at the top followed by detailed displays for each time period. The table below shows the average display. The detailed display is very similar.

<b>Column</b>	<b>Description</b>
<i>Date</i>	Date of analysis
<i>Owner</i>	Owner name
<i>Name</i>	Segment name
<i>Partition</i>	Partition name
<i>Type</i>	Segment type
<i>Tablespace</i>	Tablespace for segment
<i>Size (MB)</i>	Segment size in MB
<i>Chg. Size/day</i>	Change in segment size per day
<i># Extents</i>	Number of extents
<i>Chg. Extents/day</i>	Change in number of extents per day
<i># Blocks</i>	Number of blocks
<i>Chg. Blocks/day</i>	Change in number of blocks per day
<i>Next Extent [MB]</i>	Size in MB of the next extent to be allocated to the segment
<i>Chg. Next Extent/day</i>	Change in size of extents per day

## 3.7.10 Segments Detailed Analysis (Aggregated)

### Definition

This sub-monitor in the SAP/Oracle Database Monitor lets you display aggregated information on selected segments in the database and associated tables and indexes, partitions, and lobs, including history information and information.

For the *History* tab, it uses a collector job to periodically retrieve data from the cluster table MONI. For the other tabs, it gathers data with an SQL statement in real time.

## Use

You choose ► *Space* ► *Segments* ► *Detailed Analysis (Aggregated)* ► in the DBA Cockpit.

The system prompts you to enter selection criteria for the segments. You enter selection criteria to restrict the display to the required segments. When the main grid is populated, you can double-click any row to display it in a more concise form in the *Main data* tab.

## Structure

This screen contains the following information:

- *Main Data*

Column	Description
<i>Owner</i>	Name of segment owner
<i>Object name</i>	Object name
<i>Object type</i>	Object type
<i>Tablespace</i>	Tablespace name
<i>Partitioned</i>	Whether segment is partitioned
<i>Lobs</i>	Whether segment has lobs
<i>Size [MB]</i>	Size in MB
<i>Extents</i>	Number of extents
<i>Blocks</i>	Number of blocks
<i># Partitions</i>	Number of partitions
<i># Lobs</i>	Number of lobs
<i>Compress for</i>	Type of compression (as of Oracle 11.2)
<i>Created</i>	Indicates whether the object has been created on the database (as of Oracle 11.2)

- *Tables + Indexes*

Column	Description
<i>Owner</i>	Name of object owner

<b>Column</b>	<b>Description</b>
<i>Object</i>	Object name
<i>Object type</i>	Object type
<i>Tablespace</i>	Tablespace name
<i>Size [MB]</i>	Size of object in MB
<i>Extents</i>	Number of extents
<i>Blocks</i>	Number of blocks
<i>Partitioned</i>	Whether object is partitioned
<i># Partitions</i>	Number of partitions
<i>Lobs</i>	Whether object has lobs
<i># Lobs</i>	Number of lobs

- *Partitions*

<b>Column</b>	<b>Description</b>
<i>Owner</i>	Name of object owner
<i>Root name</i>	Root name of partition
<i>Sub component name</i>	Subcomponent name of partition
<i>Sub component type</i>	Subcomponent type of partition
<i>Tablespace</i>	Tablespace name of partition
<i>Size [MB]</i>	Size of partition in MB
<i># Extents</i>	Number of extents in partition
<i># Blocks</i>	Number of blocks in partition
<i>Initial extents</i>	Number of initial extents in partition
<i>Next extents</i>	Number of next extents in partition
<i>Minimum extents</i>	Minimum number of next extents in partition
<i>Maximum extents</i>	Maximum number of next extents in partition
<i>Percentage increase</i>	Percentage increase in number of extents allocated

Column	Description
<i>Buffer pool</i>	Buffer pool of the partition

- *Lobs*  
The information shown here is the same as the table above.
- *History*

### i Note

Each sub-tab on the *History* tab contains an average display separately at the top followed by detailed displays for each time period. The table below shows the average display. The detailed display is very similar.

Column	Description
<i>Date</i>	Date of analysis
<i>Owner</i>	Name of object owner
<i>Root name</i>	Root name of object
<i>Type</i>	Segment type
<i>Chg. Size</i>	Change in segment size per day
<i># Extents</i>	Number of extents
<i>Change in #Extents / day</i>	Change in number of extents per day
<i># Blocks</i>	Number of blocks
<i>Change in #Blocks / day</i>	Change in number of blocks per day
<i># Partitions</i>	Number of partitions
<i>Change in #Partitions / day</i>	Change in number of partitions per day
<i># Lobs</i>	Number of lobs
<i>Change in #Lobs / day</i>	Change in number of lobs per day

## 3.7.11 Collector Logs

### Definition

This monitor in the SAP/Oracle Database Monitor lets you display information about the collection job.



## Use

You choose ► [Space](#) ► [Additional Functions](#) ► [Collector Logs](#) ► in the DBA Cockpit.

## Structure

This screen contains the following information:

- The tab [Overview](#):

Column	Description
<a href="#">Dataset size (KB)</a>	Size in KB of the total amount of collected data stored in cluster table MONI
<a href="#">Avg. upload/mod (KB)</a>	Average collected data per collector module in KB
<a href="#">Avg. duration/mod</a>	Average runtime per module
<a href="#">Avg. upload/day (KB)</a>	Average collected data per day in KB
<a href="#">Avg. duration/day</a>	Average runtime per day
<a href="#">Max. upload (KB)</a>	Largest collection in terms of size in KB
<a href="#">Module ID</a>	Module ID
<a href="#">Date</a>	Date of largest collection in terms of size
<a href="#">Time</a>	Time of largest collection in terms of size
<a href="#">Max. duration</a>	Longest collection in terms of runtime
<a href="#">Module ID</a>	Module ID
<a href="#">Date</a>	Date of longest collection in terms of runtime
<a href="#">Time</a>	Time of longest collection in terms of runtime

- The tab [Modules](#) with more statistical information and logs for each collector module:

Column	Description
<a href="#">Module ID</a>	Collector module ID
<a href="#">Module name</a>	Collector module name
<a href="#">Status</a>	Current status

Column	Description
<i>Last col. date</i>	Date of last collection
<i>Last col. time</i>	Time of last collection
<i>Last upload (KB)</i>	Size of last collection in KB
<i>Last duration</i>	Duration of last collection
<i>Avg. upload (KB)</i>	Average upload within the previous LOG_LIFE days
<i>Avg. duration</i>	Average duration within the previous LOG_LIFE days
<i>Max. upload (KB)</i>	Largest upload within the previous LOG_LIFE days
<i>Date max. upload</i>	Date of largest upload
<i>Max. duration</i>	Longest runtime within the previous LOG_LIFE days
<i>Date max. duration</i>	Date of longest runtime

Where LOG\_LIFE appears in the above table, this refers to the number of days used in the analysis. LOG\_LIFE is from the table ORA\_MON\_DBCON and is set to 100 days by default.

## 3.7.12 BW Analysis

### Definition

This monitor in the SAP/Oracle DBA Cockpit lets you display SAP Business Information Warehouse (BW) objects belonging to the database.

You choose ► [Performance](#) ► [Additional Functions](#) ► [BW Analysis](#) ► in the DBA Cockpit.

### Structure

The following BW object types are each highlighted using a different row color, since they are the aggregated data for each BW area:

- PSA area
- ODS and Changelog
- Cubes and related
- BW master data
- Basis and Cross-Application

Double-click a row to display the BW objects belonging to the selected object type.

Double-click again a row to display the BW segments belonging to the selected BW object.

Double-click a BW segment displays the data below in the tabstrips.

The screen contains the following information:

- Top screen area

Column	Description
<i>BW Area</i>	BW area
<i>BW Object Type</i>	BW object type
<i>Details</i>	Details
<i># BW Objects</i>	Number of BW objects
<i># BW Root Objects</i>	Number of BW root objects
<i># Partitions</i>	Number of partitions
<i>Size [MB]</i>	Size in MB
<i>Extents</i>	Number of extents
<i>Blocks</i>	Number of blocks

- *Main Data*
- *Partitions and LOBs*

Column	Description
<i>Owner</i>	Name of partition or lob owner
<i>Root name</i>	Root name of partition or lob
<i>Sub component name</i>	Sub-component name of partition or lob
<i>Sub component type</i>	Sub-component type of partition or lob
<i>Tablespace</i>	Tablespace name of partition or lob
<i>Size [MB]</i>	Size of partition or lob in MB
<i># Extents</i>	Number of extents for partition or lob
<i># Blocks</i>	Number of blocks for partition or lob
<i>Initial extents</i>	Number of initial extents for partition or lob
<i>Next extents</i>	Number of next extents for partition or lob

Column	Description
<i>Minimum extents</i>	Minimum number of next extents for partition or lob
<i>Maximum extents</i>	Maximum number of extents for partition or lob
<i>Percentage increase</i>	Percentage increase in number of extents allocated
<i>Buffer pool</i>	Buffer pool of the partition

## 3.8 Jobs (Oracle)

### Use

You can use the DBA Cockpit for the following jobs:

- DBA Planning Calendar
- DBA logs
- Backup logs

### 3.8.1 Central Calendar

#### Use

The Central Calendar is part of the DBA Cockpit in the SAP system. It gives you a single point from which to manage database administration (DBA) actions in an integrated SAP environment. The actions available differ according to the database platform but the method of use is the same. Examples of actions are backups, database system checks, and so on.

#### **i** Note

The Central Calendar is only for viewing DBA actions by system.

However, you can easily switch to the DBA Planning Calendar for any SAP system registered in the DBA Cockpit to plan - that is, schedule, change, delete, or execute - DBA actions.

The Central Calendar gives you a single point from which to manage:

- Databases of different types and versions on remote SAP systems
- Databases for different versions of the SAP system
- Databases of non-ABAP SAP systems

#### Integration

The Central Calendar runs with all database platforms delivered as a standard part of the SAP system and supported by SAP (except DB2 for i5/OS, which has good equivalent tools).

### Features

- You can manage in real time systems directly administered from the system where the DBA Cockpit is running as well as remote systems, including non-ABAP systems.
- You can quickly check the color-coded status for each system to see if actions have executed successfully.
- You can quickly check the number of actions and number of actions with the highest status severity for each system, as shown in the following example:

#### → Tip

The entry for February 2007 in the cell for Thursday February, 08 is:

3 FUD 2

On system FUD for Thursday 8<sup>th</sup> January 2007 (in the past), there were 3 actions planned, 2 of which had the highest status severity. For example, the highest status severity might be "Warning", in which case the entry is displayed with a yellow background.

### Activities

1. On the system where you normally run the DBA Cockpit, you plan a regular job in the DBA Planning Calendar to update the results from remote systems using the action Central Calendar Log Collector. For example, you plan this job to run daily at 06:00.
2. You define the systems you want to monitor in the DBA Cockpit by setting the flag Collect Central Planning Calendar Data for each system.
3. You regularly check the results using the Central Calendar.
4. If you need to schedule, change, delete, or execute actions, you switch to the DBA Planning Calendar.

For more information, see Using the Central Calendar.

## 3.8.1.1 Using the Central Calendar

### Use

You can use the Central Calendar in the DBA Cockpit to view actions on all the databases of your SAP Systems.

### Prerequisites

- You have defined the systems to be displayed in the Central Calendar by double-clicking the required system in the screen DBA Cockpit: System Configuration Maintenance and selecting Collect Central Planning Calendar Data.  
For more information, see Configuring Systems for Remote Monitoring Using Remote Database Connections.
- In an ABAP system, make sure that you schedule the jobs for the remote database in the central monitoring system. Jobs that have been scheduled in the remote system are not displayed.
- In the DBA Planning Calendar of the DBA Cockpit where you call the Central Calendar, you have planned the action Central Calendar Log Collector to run regularly. This collects information from the defined remote systems for display in the Central Calendar.

For more information, see [Setting Up the DBA Planning Calendar](#).

### Procedure

1. Start the Central Calendar from the DBA Cockpit by choosing **Jobs** → **Central Calendar**.  
The Central Calendar is displayed. If you have already run or planned actions, you see entries by day, one for each system.  
Here is an example of entries for Thursday February, 08 affecting two systems, FUD and FIB:

→ Tip

3	FUD	2
4	FIB	1

On system FUD for Thursday 8<sup>th</sup> January, there were three actions planned, two of which had the highest status severity. For example, the highest status severity for FUD might be "Finished with warning", in which case the entry for FUD is displayed with a yellow background. This means that two actions ended with a warning.

On system FIB for the same day, there were four actions planned, one of which ended with the highest severity. For example, the highest severity for FIB might be "Finished with error", in which case the entry for FIB is displayed with a red background. This means that one action ended with an error.

The following table shows the color-coded statuses in the Central Calendar, which you can also see by choosing **Legend**:

Color	Status
Light blue	Planned
Dark blue	Running
Green	Finished successfully
Yellow	Finished with warning
Red	Finished with error
Dark yellow	No longer available
Dark red	Scheduling failed

2. To see a summary of the actions for a day, double-click the day header.  
The system displays a summary of the actions and status for each system on the day you selected, as in the following example:

→ Tip

System	Total	No longer available	Scheduled	Running	Finished	Warning	Error	Overdue
--------	-------	---------------------	-----------	---------	----------	---------	-------	---------

FUD	3	1	2	
FIB	4		3	1

3. To see the individual actions for a system, double-click the entry for the system on the required day. You see the relevant day in the DBA Planning Calendar. You can perform all usual functions in the DBA Planning Calendar.
4. To refresh the display for the system from which you called the Central Calendar, choose Refresh.
5. To refresh the display for all systems, choose Remote Refresh. You can remotely refresh the display as follows:

Method	How the Refresh Runs
Run in Dialog	Runs in dialog mode, which can take a long time, so not normally recommended
Start immediately	Runs immediately in the background as a job
Schedule at	Runs in the background at the time that you specify

### **i** Note

We recommend that you schedule action Central Calendar Log Collector to run regularly, as described above in "Prerequisites".

6. If required, you can customize the calendar display as follows:
  - Specify a factory calendar in Calendar ID. Holidays are displayed in the same background color as weekend days. This in no way restricts the planning of actions in the DBA Planning Calendar.
  - Switch to day, week, or month view by choosing Administration → View Day, Administration → View Week, or Administration → View Month.
  - Choose Save Settings and change Number of Weeks or Entries per Day in the display.

## 3.8.2 DBA Planning Calendar (Oracle)

### Use

You use the DBA Planning Calendar to automate database administration actions that have to be performed regularly. You are able to schedule operations such as online backups, have them automatically performed, and then check that the operation was successful.

The main function of the DBA Planning Calendar is to define the start times and parameters for database actions. Since these actions run without administrator interaction, you have to make sure in advance that the necessary resources are available.

## Integration

The DBA Planning Calendar is part of the Computing Center Management System (CCMS). You can start it from the DBA Cockpit.

Make sure that you meet the following requirements for BR\*Tools when using the DBA Planning Calendar:

- BR\*Tools 6.40: minimum patch level 42
- BR\*Tools 7.00: minimum patch level 24

If you are using the DBA Planning Calendar on a remote system, BR\*Tools running on that system must meet the above requirements. For more information, see SAP Note [1025707](#).

## Features

### Initial Screen

The initial screen of the DBA Planning Calendar is divided into the following frames:

- The frame on the left contains all information and parameters to select the set of actions to be displayed. You can:
  - Select the system from which you want to read planning data.
  - Select the category of an action:
    - *DBA Actions*  
These are actions that you can plan.
    - *External Actions*  
These are actions that are performed externally.
    - *All Actions*  
These are all actions.
  - Select the week to be displayed using the calendar control.  
The default is the current week. To navigate to another week, double-click the week you want to display.
  - Select a factory calendar.  
Specifying a factory calendar only has an impact on the calendar display. Holidays are the same color as weekend days. It does not result in any restrictions for planned actions.
- The frame on the right contains a list of all plannable actions that are currently available for your database.
- The frame in the lower half of the screen contains the calendar.  
By default, the week view with top-bottom layout is displayed, that is, one week starting with Monday. To change to a left-right split view, choose **► Logs ► Left-Right Split View Layout ►**. If you want to change from a week view to a day or month view, choose the corresponding button in the tool bar.

### i Note

You can only switch the layout for the week or month view. For the day view, only the left-right split view layout is available.

To change your preferred settings - that is, the layout and the view - choose [Save Settings](#).



The calendar shows the actions that were scheduled using background processing. These actions are then automatically executed. Once the action has run, the status is indicated using the following colors:

Color	Meaning
<i>Light blue</i>	The action has not yet started.
<i>Dark blue</i>	The action has not yet finished.
<i>Green</i>	The action has run successfully.
<i>Yellow</i>	The action has finished with a warning. Check the job log for details.
<i>Red</i>	An error has occurred and the action was interrupted. Check the job log for details and reschedule the action.
<i>Dark yellow</i>	No more information is available.
<i>Dark red</i>	Scheduling failed, that is, there is no status available and the action is overdue.

You can display the meaning of each color by choosing [Legend](#).

### Drag & Drop of Actions

You can move or copy actions within the calendar by using the drag & drop function.

## Related Information

[Setting Up the DBA Planning Calendar \(Oracle\) \[page 161\]](#)

### 3.8.2.1 Setting Up the DBA Planning Calendar (Oracle)

#### Use

You need to set up the DBA Planning Calendar because - when you start your system for the first time - there are no actions planned. For this you need to do the following:

The most important thing when setting up the DBA Planning Calendar is to choose a pattern of actions covering your regular database administration (DBA) needs, specifying any required action parameters and taking account of any dependencies between actions, as described in the procedure below.

## ❖ Example

You set up a pattern to do the following:

- Check and update database statistics daily at 05:00.
- Check database daily at 06:00.
- Back up the database online and the logs daily at 22:00.
- Clean up logs weekly at 07:00 on Sunday.

The DBA Planning Calendar offers you a number of preselected patterns, which you can later edit as required.

## ⚠ Caution

- Some of the actions available have an impact on database performance and availability. Check the start and end times of scheduled actions to make sure that they do not impact each other and that they do not impact other activities in your system.
- You **cannot** perform all required DBA actions from the DBA Planning Calendar or the DBA Cockpit. For more information on actions that you must perform with the SAP system down, such as offline database backup, see the *SAP Database Administration Guide* for your database

You can schedule the following actions with the DBA Planning Calendar:

Tasks	Actions
Database backup and recovery	<i>Whole database offline + redo log backup</i>
	<i>Whole database online backup</i>
	<i>Whole database online + redo log backup</i>
	<i>Full database offline + redo log backup</i>
	<i>Full database online backup</i>
	<i>Full database offline backup</i>
	<i>Full database online + redo log backup</i>
	<i>Incremental database offline + redo log backup</i>
	<i>Incremental database online backup</i>
	<i>Incremental database offline backup</i>
	<i>Incremental database online + redo log backup</i>
	<i>Partial database offline backup</i>
	<i>Partial database online</i>
	<i>Prepare for RMAN Backup</i>

Tasks	Actions
	<i>Redo log backup</i>
Space Management	<i>Adapt next extents</i>
	<i>Compress database</i>
Other	<i>Check database</i>
	<i>Check and update optimizer statistics</i>
	<i>Cleanup logs</i>
	<i>Initialize tape</i>
	<i>Validate structure</i>
	<i>Verify database</i>
	<i>Central Calendar Log Mirror</i>

## Prerequisites

Check the following before you start using the DBA Planning Calendar:

- SAP system authorizations  
Check that you have authorization for DBA and background job scheduling, which is provided by profiles S\_RZL\_ADMIN and S\_BTCH\_ALL.  
Check that external programs are able to run on the database server so that actions affecting the database can be performed from other application servers.
- Database system authorizations  
Check that actions requiring authorization to connect to the database are permitted. Some actions, such as online backup, require higher levels of authorization.  
Before scheduling database backups, see the information on how to back up the database in the *SAP Database Administration Guide* for your database.
- Hardware and backup media  
Check that you have enough hardware (such as tape drives) and backup media (such as tapes) for the backup strategy you intend to use.  
You make sure that the media are initialized and ready so that the operator does not have to interrupt scheduled backup runs. For example, check that tapes are already in the specified tape drive or tape changer. You might require different tape drives for database backups and log backups.

You also need to make sure that the for BR\*Tools - `init<DBSID>.sap` by default - is set up correctly for backup and archiving.

## Procedure

1. Start the DBA Planning Calendar from the DBA Cockpit by choosing **Jobs > DBA Planning Calendar**.
2. Choose *Pattern Setup*.
3. Follow the wizard to set up a pattern of recurring actions to cover your regular DBA needs. You can navigate between the actions in the pattern by choosing *Next* and *Previous*.
4. When you have finished defining the pattern, choose *Save* on the last screen to enter the pattern into the DBA Planning Calendar.

### ⚠ Caution

When you start using the DBA Planning Calendar in production operation, make sure that you check the results daily to make sure that actions have been executed correctly.

## 3.8.2.2 Configuring the Back End of the DBA Planning Calendar (Oracle)

### Use

You can configure the back end of the DBA Planning Calendar to control the execution of background jobs. You can configure the back end for all systems, for selected database platforms, or for single systems only.

The system is configured by using the first available configuration from the following:

1. The system-specific configuration
2. The configuration for the database platform
3. The configuration valid for all platforms
4. The standard configuration - current user, selection of background server by background dispatcher, high priority

## Procedure

1. Call the DBA Cockpit and choose **Jobs > Back End Configuration**.

### i Note

Alternatively, choose **Goto > Back End Configuration** from the menu in the DBA Planning Calendar.

2. In the *Selected Scope* group box, choose the scope of the configuration entry.
3. In *Background Dispatching* group box, choose appropriate values as follows:
4. Save your changes.

## 3.8.2.3 Scheduling an Action (Oracle)

### Procedure

1. To add new actions to the DBA Planning Calendar, you can use one of the following options:
  - Double-click a calendar cell.
  - Position the cursor on a calendar cell and choose *Add*.
  - Drag and drop an action from the action pad into a calendar cell.

#### Note

You can also use drag and drop to move actions within the calendar. If you want to copy an action, keep the **CTRL** key pressed while using drag & drop.

A dialog box appears with the details of an action.

2. If you chose the first or second option in the first step, you can select the action you want to schedule from the group box *Action Description*. In the *Planned Start* field, you can enter date and time when the action is to start. If you are entering an action for today and want to start the action immediately, choose *Execute*. If you chose the final option in the first step, the corresponding action is already listed as default. The parameters for the required action are displayed under *Action Parameters*. They vary depending on the action.
3. On the *Action Parameters* tab page, change or enter the basic parameters for the action.
4. On the *Recurrence* tab page, enter a recurrence pattern.

#### Caution

The system warns you if there is a conflict with an existing action, but it does **not** prevent you from inserting the new action.

You must decide whether the actions might conflict in terms of database access or performance. The system does not check for conflicts between actions with identical start times, but checks for actions within a range of approximately 30 minutes.

5. To schedule the action, choose *Add*.

## 3.8.2.4 Changing an Action (Oracle)

### Use

This section tells you how to change an action in the DBA Planning Calendar.

If you want to change a recurring action, the changes only affect recurrences of the action in the future. The action is split into two actions, one describing the old action and one the new action.

## Prerequisites

If you want to change an action, it must be in the state *Planned* (that is, not already executed).

### i Note

If an action has already been executed, you can only [display \[page 168\]](#) it.

## Procedure

1. Call the DBA Cockpit
2. Choose ► [Jobs](#) ► [DBA Planning Calendar](#) ► in the navigation frame of the DBA Cockpit.
3. Position the cursor on a calendar cell and choose *Edit*.  
A dialog box with the action parameters and recurrence pattern appears.
4. Apply your changes and activate them by choosing either *Change Current Occurrence* or *Change All Occurrences*.

### 3.8.2.5 Deleting an Action (Oracle)

## Prerequisites

If you want to delete an action from the DBA Planning Calendar, it must be in the state *Planned* (that is, not already executed).

### i Note

If an action has already been executed, you can only [display \[page 168\]](#) it.

## Procedure

1. Call the DBA Cockpit.
2. Choose ► [Jobs](#) ► [DBA Planning Calendar](#) ► in the navigation frame.
3. Double-click a calendar cell or position the cursor on a calendar cell and choose *Delete*.  
A dialog box appears with a list of all actions to be deleted, where you can decide if you want to delete only a single occurrence of a recurring action or all occurrences.
4. To delete an action, choose *Delete*.

## 3.8.2.6 Re-Executing an Action (Oracle)

### Use

You might have to re-execute an action in the DBA Planning Calendar. To do this, you choose an action that has already finished and repeat it immediately.

#### Example

Here are two examples when you might want to schedule an action for re-execution:

- An action such as a database backup has failed and you need to repeat it after correcting the error.
- To solve a resource bottleneck you need to re-execute an action immediately rather than wait until its next scheduled execution.

### Procedure

1. Double-click the action you want to re-execute.  
The *Display Details of Action* dialog box appears where you can see the action parameters.
2. Check and, if necessary, change the action parameters.
3. Choose *Re-Execute*.

#### Note

If the action parameters are correct, you only need to position the cursor on the action and choose *Execute*.

The action is scheduled for immediate re-execution.

## 3.8.2.7 Displaying the Status of a Day's Actions (Oracle)

### Procedure

1. Double-click the header cell for a particular day.  
The display switches to the day view. All scheduled actions are displayed.

#### Note

Unsuccessful or interrupted actions are shown in red.

2. If you want to view other days, select a new day by double-clicking a day on the calendar control at the left side of the screen.
3. To return to the week view, choose *Week*.

## 3.8.2.8 Displaying Scheduled Actions (Oracle)

### Use

You can view all action-related information from the DBA Planning Calendar:

- Action parameters
- Job logs if the action has already run  
These logs provide detailed information on the results of an action.
- Recurrence patterns

The status of an action is indicated by the color of the calendar cell where an action is inserted.

### Procedure

1. Select the action by double-clicking a calendar cell or by positioning the cursor on a cell and choosing [Action Details](#).  
The [Display Details of Action](#) dialog box appears. In the [Action Description](#) group box, scheduling information and the return status of the finished action is displayed.
2. Choose [Action Parameters](#) to see how the action is defined.  
Some actions do not have any parameters.
3. For recurring actions, choose [Recurrence](#) to see when the action is scheduled to be repeated.
4. For actions that have finished, choose [Action log](#) [Show Action/Detail Log](#) to see information about the action.

#### Caution

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.

5. For jobs that have finished, choose [Job Log](#) to see information about the background job that executed the action.  
To display long texts, if any are available, double-click a message.
6. For actions that write log files to the database server, choose [Program Log](#).  
Not all actions write program logs to the database server.



## 3.8.2.9 Troubleshooting (Oracle)

### Use

Since any action scheduled in the DBA Planning Calendar can fail, you must at least check the more critical actions such as database backups.

### Procedure

1. To check whether the background job was executed correctly, consult the job log. If no job log exists, the background job was probably not started.  
For more details, call transaction SM37 and display the job overview.

#### i Note

The names of all jobs scheduled in the DBA Planning Calendar start with *DBA*. The job log also tells you whether an external program was started.

2. If you are sure that the background job ran successfully, consult the job log or program log (if available).
3. After you have corrected the error, execute the action manually using *Execute*, making sure there are no conflicts with other scheduled actions.

#### i Note

If you want to completely clean up your jobs, choose the *Cleanup* pushbutton. This deletes all jobs, all scheduling data, and all related protocol records. It also resets the DBA Planning Calendar to its initial state.


We recommend that you clean up after an SAP system upgrade or if jobs have become corrupt.

## 3.8.3 Displaying DBA Logs (Oracle)

### Use

You can use the DBA Cockpit to view the database administration (DBA) logs created by various SAP tools during operations on the Oracle database.

### Procedure

1. Call the DBA Cockpit and choose .

2. Choose one of the following:
  - To look at the logs for operations created using BR\*Tools, choose [BRSPACE](#), [BRCONNECT](#), [BRARCHIVE](#), or [BRBACKUP](#).
  - To look at the logs for other operations, choose [Others](#).
  - To look at the logs for non-SAP data archiving, choose [Data Archiving](#).
  - To look at the logs for all operations, choose [All](#).

You see a list of operations in the logs as follows:

Heading	Description
<a href="#">Operation start</a>	Date and time for the start of the operation
<a href="#">Operation end</a>	Date and time for the end of the operation
<a href="#">FID</a>	Function ID indicating type of operation
<a href="#">RC</a>	Return code For more information, see .
<a href="#">Database Operation</a>	Description of the operation

If required, choose [Select Options](#) to restrict the logs displayed.

To see more detailed information about the database operation entries in any of the above logs, double-click the required line in the table.

3. Choose [Function IDs](#) for a list of all possible database operations, showing function ID, the program that executes the operation, and an explanation of the operation.  
You can double-click any line to jump to the list of logs for all database operations.

## 3.8.4 Display of the Backup Logs (Oracle)

### Use

This section tells you how to display information on backup for your Oracle database, including the status of the backups and the backup logs.

### Process

You choose one of the following functions:

- [Database backups](#)  
Use this to display the BRBACKUP logs, either as an overview or for the last successful or unsuccessful backup. See [Displaying Logs for Database Backups \(Oracle\) \[page 171\]](#).
- [Redo log backups](#)

Use this to display the location and status of the redo log directory as well as the amount of free space available in the directory, the backup status of offline redo logs, and BRARCHIVE backup action logs. See [Displaying Logs for Redo Log Backups \(Oracle\) \[page 172\]](#).

- *Recovery report*




Use this to check whether your backups are adequate for a database recovery and display the backup and redo logs currently required to recover the database. See [Displaying the Recovery Report \(Oracle\) \[page 174\]](#).

## 3.8.4.1 Displaying Logs for Database Backups (Oracle)

### Use

*Database backups* provide information about backups of database data files for the Oracle database. You can find the date and time of the last successful backup and the last unsuccessful backup (return code not equal to 0 or 1). You can get an overview of how up-to-date your backups are. We recommend that you keep several generations of recent successful backups.

### Procedure

1. Call the DBA Cockpit and choose  [Jobs](#)  [Backup Logs](#) .
2. In *Database backups*, choose one of the following:
  - To see the last backup, choose *Last successful backup* or - if an unsuccessful backup exists - *Last unsuccessful backup*.

You see the BRBACKUP action log:

Heading	Description
<i>Tape</i>	Tape name
<i>Position</i>	Position of the file on the tape
<i>Backup time</i>	Time the backup of the file ended
<i>Compressn</i>	Compression rate (if compression was used)
<i>DF ID</i>	Oracle data file ID
<i>Backint ID</i>	Backup ID of Backint interface (only if external backup program was used)
<i>Redo No.</i>	Log sequence number of the online redo log file
<i>Tablespace name</i>	Tablespace name

Heading	Description
<i>TS Status</i>	Oracle tablespace status
<i>DF Status</i>	Oracle data file status
<i>File Name</i>	Oracle data file name

To look at the system log for a data file in the backup, place the cursor on the relevant line and choose *Detail log*. For more information, see .

- To see an overview of all existing backup logs, choose *Overview of database backups* (backups with errors are highlighted in red):

Heading	Description
<i>Backup function</i>	Type of backup
<i>Start of backup</i>	Time the backup started
<i>End of backup</i>	Time the backup ended
<i>RC</i>	BRBACKUP return code For more information, see .
<i>Action ID/Log</i>	File system log name <Action ID>.<Function ID> For more information, see .

To see more information about a backup, double-click the appropriate line to display the BRBACKUP action log, which is described in the first table above.

To look at the system log for a data file in the backup, place the cursor on the relevant line and choose *Detail log*. For more information, see .

## 3.8.4.2 Displaying Logs for Redo Log Backups (Oracle)

### Use

*Redo log backups* provide information about backups of the offline redo log files for the Oracle database. 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 backed up.

## Procedure

1. Call the DBA Cockpit and choose **Jobs > Backup Logs**.
2. In *Redo log backups*, choose one of the following:
  - To display an overview of the offline redo log files, choose *Overview of redo log files*:

Heading	Description
<i>Redo No</i>	Log sequence number
<i>Archiving Time</i>	Time the offline redo log file was created
<i>Tape</i>	Tape name
<i>Position</i>	Position of the file on the tape
<i>Backup time</i>	Time the offline redo log file was backed up
<i>Backint ID</i>	Backup ID of Backint interface (only if external backup program was used)
<i>Offline Redo Log Name</i>	Directory and name of the offline redo log file

- To display an overview of all available backup logs for offline redo log backups, choose *Overview of redo log backups* (backups with errors are highlighted in red):

Heading	Description
<i>Function</i>	Type of backup
<i>Operation Start</i>	Time redo log backup started
<i>Operation End</i>	Time redo log backup ended
<i>RC</i>	BRARCHIVE return code For more information, see .
<i>Action ID/Log</i>	File system log name <Action ID>.<Function ID> For more information, see .

To see more information about a redo log backup, double-click the appropriate line to display the BRARCHIVE action log:

Heading	Description
<i>Function</i>	Backup functions: <ul style="list-style-type: none"> <li>• <a href="#">Save</a> Creates first copy of the offline redo log file</li> <li>• <a href="#">Copy</a> Creates second copy of the offline redo log file</li> <li>• <a href="#">Del</a> Deletes offline redo log file from the archiving directory</li> </ul>
<i>RedoNo.</i>	Log sequence number
<i>Archiving time</i>	Time the offline redo log file was created (that is, archived)
<i>Tape name</i>	Tape name
<i>Pos.</i>	Position of the file on the tape
<i>Backup time</i>	Time the offline redo log file was backed up
<i>Compr</i>	Compression rate (if compression was used)
<i>Backint ID</i>	Backup ID of Backint interface (only if external backup program was used)
<i>Offline redo log file name</i>	Directory and name of the offline redo log file

To look at the system log for a redo log file in the backup, place the cursor on the relevant line and choose [Detail log](#). For more information, see .

### 3.8.4.3 Displaying the Recovery Report (Oracle)

#### Context

The recovery report provides important information for use in the event of an Oracle database failure requiring database recovery. It displays the backups and redo logs currently required to recover the database .

## Procedure

1. Call the DBA Cockpit and choose [Jobs](#) > [Backup Logs](#) > [Recovery Report](#) .

The system displays information about the last successful backup, such as backup type and tape names. This tells you which backup to use for a recovery. The report also checks whether the required redo log files are available, backed up on tape or in the archiving directory. Therefore, you know which files to restore in the event of a recovery.

2. Check the recovery report regularly to detect possible gaps in your backups:
  - Missing redo log files mean that if an error occurs, you can no longer restore the database to the current point in time. In this case, perform a complete database backup as soon as possible.
  - If there are too many redo log files, a recovery to the current point in time might take too long. In this case, perform a complete database backup as soon as possible.

## Next Steps

# 4 Database System Check

## Use

You can use the database system check in the Computing Center Management System (CCMS) to identify existing and approaching critical situations in your Oracle database system in time. The database system check can be executed only with the database in the *OPEN* 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
- Results of database operations
- Oracle alert messages: `alert_<DBSID>.log`
- Oracle profile parameters: `init<DBSID>.ora`

## Integration

The database system check uses functionality.

## 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 \[page 177\]](#).
- Automate the database system check in the DBA Planning Calendar. If you have selected an action pattern in the DBA Planning Calendar, the database system check is automatically planned to run daily every morning. See entry *Check database* in the calendar.  
For more information, see [DBA Planning Calendar \[page 159\]](#) and [Scheduling an Action \[page 165\]](#).
- 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 \[page 182\]](#).



## 4.1 Configuring Database System Check (Oracle)

### Use

This section describes how you can configure the Oracle [database system check \[page 176\]](#) in the Computing Center Management System (CCMS) to:

- Add new conditions of type *ORA* or *PROF*
- Exclude individual conditions from the check
- Specify threshold values for the conditions
- Create object-specific conditions to exclude them from the check
- Create object-specific conditions to set individual threshold values
- Specify corresponding corrective actions
- Maintain the condition description

### Prerequisites

- The configuration data for the database system check is stored in the `DBCHECKORA` table.
- *Changed or new condition take effect the next time Check database* is executed by the [DBA Planning Calendar \[page 159\]](#) or you execute a .
- If a condition is met during a database system check, an alert is raised in the [Oracle database monitor \[page 202\]](#). For more information, see [Displaying Alert Messages from Database System Check \[page 182\]](#).

### Procedure

1. Choose **Administration > CCMS > DB Administration > DB System Check > Configuration** .  
Alternatively, call transaction `DB17`.  
In *Number of Database Check Conditions*, you can see an overview of the current status of the check conditions:
  - *Database Administration*: Number of database administration check conditions
  - *Database Operations*: Number of database operation check conditions
  - *Database Messages*: Number of Oracle alert check conditions
  - *Database Profile*: Number of Oracle parameter check conditions
  - *Active*: Number of active check conditions
  - *Inactive*: Number of inactive check conditions
  - *Total*: Total number of check conditions*In the table you can see the following columns:*

Column	Description
<i>Type</i>	<p>Type of alert the condition is assigned to:</p> <ul style="list-style-type: none"> <li>• <i>DBA</i> These conditions check database space (fill level and fragmentation), physical consistency (availability of data) and DBA-specific problems, in the database system.</li> <li>• <i>DBO</i> These conditions check DBA operations that: <ul style="list-style-type: none"> <li>• 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.</li> <li>• Have not been processed successfully since a specific (time) period</li> </ul> </li> <li>• <i>ORA</i> These are Oracle alert conditions. If there is an error, these are read from the Oracle alert file, <code>alert_&lt;DBSID&gt;.log</code>, and an alert is reported. You can define any number of new conditions (that is, Oracle error code or Oracle message strings).</li> <li>• <i>PROF</i> These profile conditions compare the values of the Oracle database conditions with the values in the <code>DBCHECKORA</code> configuration table. You can define any number of new conditions (that is, Oracle profile conditions).</li> </ul>
<i>Condition</i>	<p>The technical name of the condition.</p> <ul style="list-style-type: none"> <li>• Type <i>DBA</i>, see .</li> <li>• Type <i>DBO</i>, see .</li> <li>• Type <i>ORA</i>, see "Oracle Messages" in <a href="#">Database Health Alerts [page 211]</a>.</li> <li>• Type <i>PROF</i>, see "Profile Parameters" in <a href="#">Database Health Alerts [page 211]</a>.</li> </ul>

Column	Description
<i>Object</i>	<p>Object-specific instance of a condition of type <i>DBA</i> or <i>DBO</i>. The instance can be used to exclude the object from the check or to individualize the object settings.</p> <p>To specify object-specific instances of the condition, a corresponding master condition must exist. This master condition is the standard for all specific instances of the condition. This master condition does not contain any object specifications.</p> <div data-bbox="821 660 1402 862" style="background-color: #f0f0f0; padding: 5px;"> <p><b>i Note</b></p> <p>Only function IDs can be specified for conditions of type <i>DBO</i>. All function IDs used by DBA tools can be found in the table <i>DBAFID</i>.</p> </div>
<i>Actv.</i>	<p>Indicates whether the condition is active (<i>green</i>) or inactive (<i>red</i>)</p>
<i>Level</i>	<p>Severity of the alert message:</p> <p><i>E</i> Error</p> <p><i>W</i> Warning</p> <p><i>A</i> Exception due to a deviation from the SAP standard (for example, a condition for a <i>PROF</i> alert message is not within the value range we recommend)</p>

Column	Description
<i>Operands</i> , <i>Val.</i> , <i>Unit</i>	<p>Definition of the threshold value using operand, value and unit.</p> <p>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.</p> <p>The possible units are:</p> <ul style="list-style-type: none"> <li>• <i>D</i> Days</li> <li>• <i>G</i> Gigabyte</li> <li>• <i>H</i> Hours</li> <li>• <i>K</i> Kilo</li> <li>• <i>M</i> Megabyte or minutes</li> <li>• <i>P</i> Percent</li> <li>• <i>R</i> Rate</li> <li>• <i>S</i> Seconds</li> </ul>
	<div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p><b>❖ Example</b></p> <p>With <i>Operand</i> = &gt;, <i>Value</i> = 80 and <i>Unit</i> = <i>P</i>, an alert is triggered when the value of the condition checked exceeds 80%.</p> </div>
<i>Period</i> , <i>Unit</i>	These fields are not used by BRCONNECT.
<i>Date</i>	Date the last time the condition was changed
<i>User</i>	User who made the last changes to the condition
<i>CorrType</i>	<p>Tool to use for the corrective measures:</p> <ul style="list-style-type: none"> <li>• <i>D</i> DBA action</li> <li>• <i>E</i> Text editor</li> <li>• <i>H</i> Hotline</li> <li>• <i>P</i> DBA program</li> <li>• <i>R</i> SAP report</li> <li>• <i>T</i> SAP transaction</li> </ul>
<i>CorrMeasure</i>	Corrective measures to fix an alert for the condition
<i>Description</i>	Description of condition

1. To display details about a selected condition, choose *Details*.  
The system displays details of the selected condition, as shown in the table below for creating a new condition.
2. To create a new condition of type *ORA* or *PROF*:
  1. Select an existing condition of the required type *ORA* or *PROF*.

2. Choose *Create with template*.
3. Enter data in the fields displayed:

Screen Area	Field	Description
<i>Condition definition</i>	<i>Condition Type</i>	See <i>Type</i> in table above.  Choose <i>Database message</i> , to add a new condition of type <i>ORA</i> or <i>Database Profile</i> to add a new condition of type <i>PROF</i> .
	<i>Condition Name</i>	See <i>Condition</i> in table above.
	<i>Check object</i>	See <i>Object</i> in table above.
	<i>Active flag</i>	See <i>Actv.</i> in table above. Enter <i>Yes</i> to activate the parameter.
	<i>Condition level</i>	See <i>Level</i> in table above.
	<i>Check condition</i>	See <i>Operands, Val., Unit</i> in table above.
	<i>Description</i>	See <i>Description</i> in table above.
<i>Repetition period</i>	<i>Duration</i> and <i>Time Unit</i>	Not used
<i>Corrective measure</i>	<i>Correction type</i>	See <i>CorrType</i> in table above.
	<i>Corrective action</i>	See <i>CorrMeasure</i> in table above.
<i>Changed by</i>	<i>User</i> and <i>Date</i>	The user and date of the last change (you cannot enter data in these fields)

4. Choose *Save* to save your changes
3. To change a selected existing condition:
  1. Select an existing condition.
  2. Choose *Change*.
  3. Enter data in the fields displayed as described in the table above for creating a new condition.

#### **i** Note

When you change an existing condition, you cannot enter data in all fields.

4. To make a selected condition object-specific, choose *Copy for object* and enter the name of the object in *Check Object*, then do the following:
  - To exclude an object from the check, select *No* for *Active Flag*.
  - To set an individualized threshold, enter data as described in the table above for creating a new condition, making sure to enter data in the check condition fields.

### i Note

You cannot make all conditions object-specific.

- To activate or deactivate a condition, select the relevant condition and select the *Actv.* field for that condition.  
The condition switches to active or not active.

## 4.2 Displaying Alert Messages from Database System Check

### Use

This section describes how you can view alert messages for the Oracle [database system check \[page 176\]](#) in the Computing Center Management System (CCMS).

### ⚠ Caution

Always deal with alert messages as quickly as possible. Alert messages with severity *E* (error) are serious problems for which we recommend you to take immediate action.

For more information on alerts raised here, see [Database Health Alerts \[page 211\]](#).

### Procedure

- Choose ► *Administration* ► *CCMS* ► *DB Administration* ► *DB System Check* ► *Display alerts* ►.  
Alternatively, choose transaction DB16.  
The system shows the following display:

Screen Area	Field	Description
<i>Check results</i>	<i>Red light</i>	Number of error alerts
	<i>Yellow light</i>	Number of warning alerts
	<i>Gray light</i>	Number of exception alerts, that is, where there is a deviation from the SAP standard value
	<i>Total</i>	The sum of errors, warnings, and exceptions

Screen Area	Field	Description
<i>Settings</i>	<i>View the last</i>	Controls the period for which the alerts are displayed
	<i>Refresh every</i>	Controls the refresh interval for the alerts
	<i>Delete after</i>	Controls the period after which alerts are deleted
<i>Alert table</i>	<i>Level</i>	Severity of the alert message:  <i>E</i> : Error (for errors of alert type DBA take immediate action)  <i>W</i> : Warning  <i>A</i> : 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)  <i>C</i> : An error occurred, but it has been fixed
	<i>Date</i>	Date of alert
	<i>Time</i>	Time of alert
	<i>Days</i>	Age in days of alert
	<i>Type</i>	Alert message type ( <i>DBA</i> , <i>DBO</i> , <i>ORA</i> , and <i>PROF</i> ) to which the parameters of the database system check are assigned  For more information, see <a href="#">Configuring Database System Check [page 177]</a> .
	<i>Name</i>	The technical name of the condition.  Type DBA, see .  Type DBO, see .  Type ORA, see "Oracle Messages" in <a href="#">Database Health Alerts [page 211]</a> .  Type PROF, see "Profile Parameters" in <a href="#">Database Health Alerts [page 211]</a> .
	<i>Object</i>	An object-specific alert of type DBA or DBO.

Screen Area	Field	Description
	<i>Number</i>	Number of alerts of the same kind
	<i>Description</i>	Description of the alert

2. You can perform the following actions:

- Refreshing the alert display  
Choose [Refresh](#) to read the check results again and so refresh the alert display.  
To automatically refresh the display, choose ► [Settings](#) ► [Auto-refresh](#) ► [Activate](#) ►. To set the time interval for automatic refresh, choose ► [Settings](#) ► [Auto-refresh](#) ► [Time interval](#) ►.
- Starting database system check  
Choose [Start](#) to immediately start the database system check.
- Viewing standard or history
  - Choose [Standard](#) to view all open messages in the selected time period.
  - Choose [History](#) to view all messages.
- Configuring the database system check  
Choose [Configure](#) to [configure database system check \[page 177\]](#).
- Viewing the database operations monitor  
Choose [DBA Operations](#) to [display the DBA logs \[page 169\]](#).
- Working with individual alerts
  - To see full details of an alert, double-click on the alert and choose [Check log](#).
  - To delete a selected alert, choose ► [Edit](#) ► [Delete](#) ►.
  - To reset a selected alert that you have fixed, choose ► [Edit](#) ► [Reset](#) ►.
  - To refresh a selected alert, choose ► [Edit](#) ► [Refresh](#) ►.
- Configuring the alert display
  - Choose ► [Settings](#) ► [View](#) ► [Time Period](#) ► and enter the number of days for which you want to view alerts. For example, if you enter 5 days, you see standard or history alerts for the last five days.
  - Choose ► [Settings](#) ► [Auto-delete](#) ► [Activate](#) ► to automatically delete alerts after the specified time interval. To set the time interval, choose ► [Settings](#) ► [Auto-delete](#) ► [Time Interval](#) ►.
  - To save your settings, choose ► [Settings](#) ► [Save settings](#) ►.



# 5 Update Statistics for the Cost-Based Optimizer in CCMS (Oracle)

## Use

You can update statistics on the Oracle database using the Computing Center Management System (CCMS).

By running update statistics regularly, you make sure that the database statistics are up-to-date, so improving database performance. The Oracle cost-based optimizer (CBO) uses the statistics to optimize access paths when retrieving data for queries. If the statistics are out-of-date, the CBO might generate inappropriate access paths (such as using the wrong index), resulting in poor performance.

The CBO is a standard part of the SAP system. If statistics are available for a table, the database system uses the cost-based optimizer. Otherwise, it uses the rule-based optimizer.

You can also run update statistics for your Oracle database using BRCONNECT. Refer to . This is the recommended way to update statistics.

### i Note

Update statistics after installations and upgrades

You need to update statistics for all tables in the SAP system after an installation or an upgrade. This is described in the relevant installation or upgrade documentation.

## Process

1. You use the DBA Planning Calendar in CCMS to schedule regular execution of check statistics and, if necessary, update statistics. For more information, see [Updating Statistics in the DBA Planning Calendar \(Oracle\) \[page 186\]](#).
2. If required, you run one-off checks on 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 a long time. You can check, create, update, or delete statistics for:
  - [Single tables \[page 187\]](#)
  - [Groups of tables \[page 191\]](#)
3. If required, you [configure update statistics \[page 192\]](#) 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 BRCONNECT, 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. Do **not** add, delete, or change table entries unless you are aware of the consequences.

## 5.1 Updating Statistics in the DBA Planning Calendar (Oracle)

### Prerequisites

- You are [ready to use CCMS \[page 6\]](#).
- You know how to use the DBA Planning Calendar.  
For more information about scheduling an action (for example, update statistics) in the DBA Planning Calendar, see [Scheduling an Action \(Oracle\) \[page 165\]](#). Update statistics is one of the predefined action patterns in the DBA Planning Calendar, so you do not need to add it separately.
- You can prioritize the statistics for critical tables by changing entries in the control table DBSTATC. For more information, see [Configuring Update Statistics in CCMS \(Oracle\) \[page 192\]](#).
- If you want to update statistics for a single table or a group of tables, instead of for all tables (as in this procedure), see the following:
  - [Updating Statistics for a Single Table \(Oracle\) \[page 187\]](#)
  - [Updating Statistics for Groups of Tables in CCMS \(Oracle\) \[page 191\]](#)

### Context

You can use the DBA Planning Calendar in the Computing Center Management System (CCMS) to schedule check and update statistics for the cost-based optimizer in the Oracle database.

All tables in the database are then **automatically** checked and updated on a regular basis.

#### i Note

The action that you plan with this procedure now includes both check statistics and update statistics in a single integrated phase. This is equivalent to the two-phase approach previously used. For more information, see .

### Procedure

1. Choose **Administration** > **CCMS** > **DB Administration** > **DBA Planning Calendar** . Alternatively, call transaction DB13.
2. Choose the day when you want the statistics to be checked and updated.
3. Choose **Create action**.
4. Select **Check and update optimizer statistics** to check and update statistics.

5. Enter data as required in the fields *Start Time*, *Period (weeks)*, and *Calendar*.

#### → Recommendation

We recommend you to schedule this action every week, preferably at a time when the database is not under heavy load. It might take several hours to complete and can significantly load the system.

6. Choose *Continue*.

## Results

The check and update statistics are now scheduled. They are executed at the scheduled date and time. For more information about looking at the results, see [Displaying the Status of a Day's Actions \(Oracle\) \[page 167\]](#).

## 5.2 Updating Statistics for a Single Table (Oracle)

### Use

You can use this procedure to update the statistics for a single table in your Oracle database if required. For example, use this function when:


- You or SAP EarlyWatch Alert determine that the statistics for a particular table are out-of-date. That is, you have a performance problem relating to out-of-date table statistics.
- You import a large amount of data into a particular table (for example, using batch input). The additional records make the statistics inaccurate.

Update statistics is an online action that does not basically affect system operation, although it creates a certain amount of additional load on the database.

### Prerequisites

- This procedure uses .
- For more information on updating statistics in the DBA Planning Calendar, see [Updating Statistics in the DBA Planning Calendar \(Oracle\) \[page 186\]](#).

### Procedure

1. Choose **Administration** > **CCMS** > **DB Administration** > **Cost-based optimizer** > **Create statistics** . Alternatively, call transaction DB20.

2. Enter the name of the table in *Table* (do not enter a generic name or pattern) and choose *Refresh information* to display information about current table statistics. The system displays the following information:

Screen Area	Field	Description
<i>Status</i>	<i>Status message</i>	<p>The system displays a status message and a color-coded status indicator for the displayed table:</p> <p>Green: Statistics are up-to-date</p> <p>Yellow: Statistics are <b>not</b> up-to-date</p> <p>Red: Statistics are damaging or missing</p> <p>Gray: Ignore flag is set in DBSTATC for this table</p>
	<i>New method</i>	<p>The update statistics method to be used the next time update statistics runs:</p> <p><i>E</i> Estimate table statistics including indexes</p> <p><i>EH</i> Estimate table statistics and create histograms</p> <p><i>EI</i> Estimate table statistics and verify index structure</p> <p><i>EX</i> Estimate table statistics, create histograms, and verify index structure</p> <p><i>C</i> Compute table statistics including indexes</p> <p><i>CH</i> Compute table statistics and create histograms</p> <p><i>CI</i> Compute table statistics and verify index structure</p> <p><i>CX</i> Compute table statistics, create histograms, and verify index structure</p> <p>For more information, see .</p>
	<i>New sample size</i>	<p>The sample size to be used the next time update statistics runs. For more information, see .</p>
<i>Last refresh</i>	<i>Old method</i>	<p>The update statistics method used in the last run of update statistics. See <i>New method</i> above for possible values.</p>

	<i>Old sample size</i>	The sample size for create statistics in the last run of update statistics. For more information, see .
	<i>Date</i>	Date when the last statistics were created
	<i>Time</i>	Time when the last statistics were created
<i>Number of Table Entries</i>	<i>Old value</i>	Number of rows in the table when the last statistics were created
	<i>New value</i>	Current number of rows in the table
	<i>Deviation old / new</i>	Percentage deviation between <i>Old value</i> and <i>New value</i>
	<i>Deviation new / old</i>	Percentage deviation between <i>New value</i> and <i>Old value</i>
<i>DBSTATC Settings</i>	<i>Use</i>	<p>The service or application for which the table entry is used:</p> <p><i>A</i> (application monitor) For the SAP application monitor (ST07) and the optimizer</p> <p><i>O</i> (optimizer) Only for the optimizer (standard setting)</p>

*Active*

Control flag, indicating if and how the statistics are updated or created, and in which context they are used. The possible settings are:

*A* (Active) Statistics are checked and updated if needed.

*N* (No) Statistics cannot be created. You can only set this if *Use* is *O*. BRCONNECT deletes existing statistics during next run of brconnect -f stats.

*P* (Priority) Priority Statistics (*P*) are checked and updated if needed before statistics with lower priority (*A*).

*R* (Restrictive) Statistics can be temporarily created. Set this for pool or cluster tables if *Use* is *A*. BRCONNECT deletes statistics immediately after they have been stored in the DBSTAT-TORA and DBSTATORA tables.

*U* (Unconditional) Statistics are updated without checking during every run of brconnect -f stats.

*I* (Ignore) BRCONNECT does **not** check whether create or update statistics is required. If statistics are present, they are **not** updated or deleted.

---

<i>Analysis method</i>	See <i>New method</i> in screen area <i>Status</i> above.
<i>Sample size</i>	See <i>New sample size</i> in screen area <i>Status</i> above.
<i>History</i>	Statistics results are archived to the DBSTATHORA and DBSTAIHORA tables.
<i>Customer</i>	Suppresses warnings issued by BRCONNECT if pool or cluster tables were specified in the DBSTATC table to always have statistics. Only use this field in exceptional circumstances.
<i>Update</i>	BRCONNECT updates statistics once without check, even when <i>Active</i> is <i>I</i> , <i>N</i> , or <i>R</i> . When it has finished, brconnect -f stats resets the <i>Update</i> field.

---

*Check*

BRCONNECT checks once whether new statistics are required and updates them if necessary, even when *Active* is *I*, *N*, or *R*. If required, you can manually change the entries in the DBSTATC control table with transaction DB21. When it has finished, `brconnect -f stats` resets the *Check* field.

3. Choose *Create statistics* to update statistics for the table using the values in *Status*. We recommend you to only change the values in *Status* in exceptional circumstances.
4. The system updates statistics for the table and updates *Last statistics* and *Number of table entries*.
5. Choose *Delete statistics* to delete statistics for the table.
6. Choose *Check structure* to , using with the option `-t <table name> -v cascade`.
7. Choose *Standard Operations* to [update statistics for groups of tables in the database \[page 191\]](#).
8. Choose *Configure* to [configure update statistics \[page 192\]](#), that is, edit the DBSTATC table.
9. Choose *DBA Operations* to [display DBA logs \[page 169\]](#).

## 5.3 Updating Statistics for Groups of Tables in CCMS (Oracle)

### Prerequisites

This procedure uses .

### Context

You can use this procedure to update the statistics for groups of tables in your Oracle database.

#### → Recommendation

We normally recommend you to [schedule update statistics in the DBA Planning Calendar \[page 186\]](#), so that it is automatically executed at regular intervals.

## Procedure

1. Choose **Administration** > **CCMS** > **DB Administration** > **Cost-based optimizer** > **Create statistics** .  
Alternatively, call transaction **DB20**.
2. Choose **Standard operations**.
3. Select one of the following:
  - **Update Statistics**  
Deletes unwanted statistics, checks statistics for each table and updates them if required, using with the option `-t all`.
  - **Update Statistics (DBSTATC)**  
Updates statistics for tables with an entry in the DBSTATC control table, using with the option `-t dbstatc_tab`.
  - **Create missing**  
Creates statistics for all tables and indexes that ought to have statistics but do not currently, using with the option `-t missing`.
  - **HARMFUL**  
using with the option `-d harmful`.
4. Choose **Start** to activate the selected function.
5. To check the results, choose **DBA Operations**.

For more information, see [Displaying DBA Logs \(Oracle\) \[page 169\]](#).

## 5.4 Configuring Update Statistics in CCMS (Oracle)

### Use

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 . Only change the control table under special circumstances.

This procedure lets you:

- Display the control table, to check the settings for the CBO update statistics. Use this, for example, to check if a statistics update is planned, the last time settings were changed, or which method was used to update the statistics.
- Change the default method for collecting statistics

### Prerequisites

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

### ⚠ Caution

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 ► *Administration* ► *CCMS* ► *DB Administration* ► *Cost-based optimizer* ► *Configuration* ►. Alternatively, call transaction DB21.

2. Check the control table entries.

The table includes the following columns:

- *Database object*: Name of the SAP table, whose statistics are to be checked and updated
- *Database*: Database system type (for example, *ORACLE* or *space*, which means relevant for all database systems)
- *Use*: Use of the table in the SAP System:
  - *A*: For the application monitor ( *ST07*) and also for the optimizer
  - *O*: Only for the optimizer (default setting)
- *Active*: Control flag, indicating if and how the statistics are updated or created, and in which context they are used. The possible settings are:
  - *A* (Active)  
Statistics are checked and updated if needed.
  - *N* (No)  
Statistics cannot be created. You can only set this if *Use* is *O*. BRCONNECT deletes existing statistics during the next run of `brconnect -f stats`.
  - *P* (Priority)  
Priority Statistics (P) are checked and updated if needed before statistics with lower priority (A).
  - *R* (Restrictive)  
Statistics can be temporarily created. Set this for pool or cluster tables if *Use* is *A*. BRCONNECT deletes statistics immediately after they have been stored in the DBSTATTORA and DBSTATIORA tables.
  - *U* (*Unconditional*)  
Statistics are updated without checking during every run of `brconnect -f stats`.
  - *I* (Ignore)  
BRCONNECT does not check whether create or update statistics is required. If statistics are present, they are not updated or deleted.

### i Note

Only set this to *N* in emergencies. For the CBO, setting *R* is sufficient.

- **TODO:** When set, the statistics for the table are updated or created during the next update statistics run. To force an update of statistics for a table, set the TODO flag yourself.
  - **Analysis:** Analysis method used to update or create the statistics. You can manually change the setting if required. The possible analysis methods are:
    - **E**  
Estimate table statistics including indexes
    - **EH**  
Estimate table statistics and create histograms
    - **EI**  
Estimate table statistics and verify index structure
    - **EX**  
Estimate table statistics, create histograms, and verify index structure
    - **C**  
Compute table statistics including indexes
    - **CH**  
Compute table statistics and create histograms
    - **CI**  
Compute table statistics and verify index structure
    - **CX**  
Compute table statistics, create histograms, and verify index structure  
For more information, see .
  - **Sample:** Shows the sample size for update statistics. For more information, see .
  - **Hist:** Statistics results are archived to the DBSTATHORA and DBSTAIHORA tables.
  - **Cust:** Suppresses warnings issued by BRCONNECT if pool or cluster tables were specified in the DBSTATC table to always have statistics. Only use this field in exceptional circumstances.
3. Choose **New entries** to add tables to the control table DBSTATC as follows:

Screen Area	Field	Description
<i>Object information</i>	<i>Database object</i>	Table name (for example, <i>APQD</i> )
	<i>Object type</i>	Database object type (default setting: <i>O1</i> for table)
	<i>Object owner</i>	Owner of the object (default setting <i>SAPR3</i> ).
	<i>Database type</i>	Database system type (for example, <i>ORACLE</i> ). You do not normally have to enter this field
	<i>EDM Cluster</i>	An area in the Enterprise Data Model (EDM)

Screen Area	Field	Description
<i>Default settings</i>	<i>Use</i>	This identifies the service or application for which the table entry is used:  A (application monitor)  For the SAP application monitor (ST07) and the optimizer  O (optimizer)  Only for the optimizer (standard setting)
	<i>Active flag</i>	See <i>Active</i> above.
	<i>Changed on</i>	Date the entry in the DBSTATC control table was changed.
	<i>Analysis method</i>	See <i>Method</i> above
	<i>Sample size</i>	Sample size for updating statistics  For more information, see .
	<i>History</i>	Statistics results are archived to the DBSTATHORA and DBSTAIHORA tables
	<i>Customer flag</i>	Suppresses warnings issued by BRCONNECT if pool or cluster tables were specified in the DBSTATC table to always have statistics  Only use this field in exceptional circumstances.
<i>ToDo settings</i>	<i>Changed on</i>	Date the ToDo entry in the DBSTATC control table was changed.
	<i>Check</i>	BRCONNECT checks once whether new statistics are required and updates them if necessary, even when <i>Active</i> is <i>I</i> , <i>N</i> , or <i>R</i> .  When it has finished, <code>brconnect -f stats</code> resets the <i>Check</i> field.
	<i>Update</i>	BRCONNECT updates statistics once without check, even when <i>Active</i> is <i>I</i> , <i>N</i> , or <i>R</i> .  When it has finished, <code>brconnect -f stats</code> resets the <i>Update</i> field.

Save the new entry and exit the table maintenance function.

### i Note

An alternative method to add entries to the control table is to choose ► *Edit* ► *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.

## More Information

- 
- [Updating Statistics in the DBA Planning Calendar \(Oracle\) \[page 186\]](#)

# 6 Maintaining Database Parameters (Oracle)

## Use

This procedure lets you:

- 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 [Administration](#) > [CCMS](#) > [DB administration](#) > [DB parameters](#). Alternatively, call transaction DB26.

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 for the system (command `ALTER SYSTEM . . .`)
- Database parameters modifiable for the session (command `ALTER SESSION . . .`)
- Parameters modified since the last database start
- Database parameters in alert status, for which the current value is different than the value specified in the database system check (table DBCHECKORA)

The individual database parameters are displayed in a table. The table includes the following columns:

Column	Meaning
<i>Name</i>	Name of the database parameters
<i>Value</i>	Value of the database parameter

Column	Meaning
<i>Status</i>	<p>Status of the database parameter</p> <p><i>Red</i></p> <p>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 <i>Error</i>.</p> <p><i>Yellow</i></p> <p>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 <i>Warning</i>.</p> <p><i>Green</i></p> <p>The value of the database parameter is identical to the threshold value or within the validity range specified in the database system check</p> <p><i>Gray</i></p> <p>The parameter is not checked by the database system check.</p> <p>More information:</p> <p><a href="#">Configuring Database System Check [page 177]</a></p>
<i>Changed</i>	<p>Whether the parameter has been changed since the last database start</p> <p><i>FALSE</i></p> <p>The parameter has not been changed since the last database start.</p> <p><i>MODIFIED</i></p> <p>The parameter was changed using the command <code>ALTER SESSION . . .</code> since the last database start.</p> <p><i>SYSTEM_MOD</i></p> <p>The parameter was changed using the command <code>ALTER SYSTEM . . .</code> since the last database start.</p>

Column	Meaning
<i>Default</i>	<p>Whether the parameter set to its default value</p> <p><i>TRUE</i></p> <p>Parameter is set to default value</p> <p><i>FALSE</i></p> <p>Parameter is <b>not</b> set to default value</p>
<i>Ses.modif.</i>	<p>Whether the database parameter can be modified for the session?</p> <p><i>TRUE</i></p> <p>The parameter can be modified during the session using the command <code>ALTER SESSION . . .</code></p> <p><i>FALSE</i></p> <p>The parameter cannot be modified during the session.</p>

#### **i** Note

Modifying a parameter immediately affects the database user session in which the parameter was modified. In the SAP context this means that the change affects all SAP user sessions for which dialog steps are processed by this database work process.

Since this does not involve all SAP user sessions, changes to session-modifiable parameters are **not** supported.

Column	Meaning
<i>Sys.modif.</i>	<p>Can the database parameter be modified for the whole system?</p> <p><i>IMMEDIATE</i></p> <p>The parameter can be changed using the command <code>ALTER SYSTEM . . .</code>. This means that it takes effect for all current database user sessions and for the entire database.</p> <p><i>DEFERRED</i></p> <p>The parameter can be changed using the command <code>ALTER SYSTEM . . .</code> and it takes effect for all future database user sessions (for example, for SAP application servers started after the change went into effect).</p> <p><i>FALSE</i></p> <p>The parameter is not system-modifiable.</p> <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p><b>i Note</b></p> <p>To use the command <code>ALTER SYSTEM . . .</code> you must have an authorization for <code>ALTER SYSTEM</code> for the database.</p> </div>
<i>Adjusted</i>	<p>Was the parameter automatically adjusted by the database system?</p> <p><i>TRUE</i></p> <p>The parameter was automatically adjusted by the database system.</p> <p><i>FALSE</i></p> <p>The parameter was not automatically adjusted by the database system.</p>
<i>Description</i>	Oracle description for the database parameter

### Displaying Specific Views of Database Parameters

- 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 Oracle database system parameters with the default value, choose **Selection criteria** > **Default**.
- To display all Oracle database system parameters that are not set to the default value, choose **Selection criteria** > **Not default**.



- To display all parameters modified since the last database start, choose **► Selection criteria ► Modified since startup ►**.
- To display all parameters checked using the function *Database System Check* since the last database start, choose **► Selection criteria ► With check ►**.
- To display all parameters in alert status, choose **► Selection criteria ► In alert status ►**.

### 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*.

#### **i** Note

To make the change effective even after a database restart, you must maintain the database parameter in the `init<DBSID>.ora` Oracle initialization file or server parameter `spfile`.

You can usually find this file in:

- `$ORACLE_HOME/dbs` for UNIX
- `%ORACLE_HOME%\database` for Windows

We recommend you to use BRSPACE for this. For more information, see .

### Displaying the `init<DBSID>.ora` Oracle Initialization File

To display the Oracle initialization file `init<DBSID>.ora`, choose *Display Oracle profile*.

# 7 Monitoring the Oracle Database

## Use

With the Oracle database alert monitor, you can keep track of the following database functions for your SAP System:

- [Space management \[page 203\]](#) to monitor tablespaces and segments.
- [Performance \[page 206\]](#) to monitor optimizer statistics, buffers, logs, and checkpoints
- [Backup/restore \[page 208\]](#) to monitor database and redo log backup
- [Consistency \[page 209\]](#) to check the consistency between the indexes, database objects, and database tables included in the ABAP Dictionary and those in the Oracle data dictionary.
- [Health \[page 211\]](#) to monitor BRCONNECT database system checks

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.

For more information, see [Information on Oracle Database Alerts \[page 229\]](#).

## Prerequisites

Many alerts, especially those under *Health*, are based on [Database System Check](#), which delivers much of the data required for database monitoring to the SDBAD table. This avoids time-consuming access to the Oracle data dictionary view DBA\_SEGMENTS.

### → Recommendation

Therefore, we recommend you to schedule at least daily BRCONNECT database checks in the DBA Planning Calendar, so that the alerts in the database monitor are up-to-date.

For more information, see [Database System Check \[page 176\]](#) and [SAP Note 483659](#).

### i Note

Database system check in the SAP System uses BRCONNECT by default from SAP Web Application Server 6.10 onwards. If you have an older release that still uses SAPDBA and you want to change to BRCONNECT, see [SAP Note 403704](#).

## Process

1. To start the Oracle database monitor, choose [Administration](#) > [CCMS](#) > [Control/Monitoring](#) > [Alert Monitor](#) > [SAP CCMS Monitor Templates](#) > [Database](#) . Alternatively, call transaction RZ20.
2. Open the monitors and check the results.
3. If any alerts have been raised, take the necessary action to correct the problem.

### i Note

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.

## 7.1 Space Management Alerts

### Definition

You might see the alerts described here when [monitoring your Oracle database \[page 202\]](#).

### i Note

[BRCONNECT] below means that the alert is raised by the .

[OPTIONAL] below means that the alert can be turned off here and shown only in the *Health* monitoring tree. Such alerts always have corresponding *Health* alerts, as described in [Database Health Alerts \[page 211\]](#).

For more information, see [Information on Oracle Database Alerts \[page 229\]](#).

### Structure

#### Tablespaces

<tablespace name>

**Cause:** Tablespace alerts can be raised as follows (data file autoextend is taken into account before alert is raised):

- *Freespace*: Amount of freespace in the tablespace
- *Used space*: Percentage of used space in the tablespace
- *Status*: Tablespace status (online or offline)

**Action:** , if necessary, based on your estimate of anticipated growth.

If you add a tablespace, run the RSDBMON0 report from Transaction SE38 so the new tablespace is recognized and represented as a new node in the *Tablespaces* subtree.

If you drop a tablespace, delete the node representing that tablespace.

## Segments

Segments with too few allocatable extents

**Cause:** 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*) are counted and included in this node. Only segments in the SAP schema for which PCTINCREASE is set to zero are considered.

**Action:** 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 data file to the tablespace or activate the feature to autoextend data files.

The collection method that resides at this node feeds other nodes as well.

### i Note

Running the collection and analysis methods associated with this node can be very resource intensive if you do not run the BRCONNECT database system check at least once a day.

Fewest allocatable extents for a segment

**Cause:** The smallest number of extents left for any segment has fallen below the configured threshold. Only segments in the SAP schema for which PCTINCREASE is set to zero are considered.

**Action:** 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 data file to the tablespace or activate the feature to autoextend data files.

### i Note

Running the collection and analysis methods associated with this node can be very resource intensive if you do not run the BRCONNECT database system check at least once a day.

Segments approaching MAX\_EXTENTS

**Cause:** The number of segments that trigger red or yellow alerts for the node *Fewest extents left before max\_extents* has exceeded the configured threshold.

**Action:** Increase the segment's storage parameter MAX\_EXTENTS if you expect the segment to grow.

### i Note

Running the collection and analysis methods associated with this node can be very resource intensive if you do not run the BRCONNECT database system check at least once a day.

Fewest extents left before MAX\_EXTENTS

**Cause:** The smallest number of allocatable extents for any segments before MAX\_EXTENTS is reached has fallen below the configured threshold.

**Action:** 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 can be very resource intensive if you do not run the BRCONNECT database system check at least once a day.

Most allocated extents in any segment

**Cause:** 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.

**Action:** If you prefer segments with fewer allocated extents, you can use BRSPACE to reorganize the object.

Segments with non-zero PCTINCREASE

**Cause:** 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 the SAP 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.

**Action:** Evaluate whether the segment's storage parameter can be set to zero.

Segments in wrong tablespace [BRCONNECT] [OPTIONAL]

**Cause:** A segment is in the wrong tablespace, as raised by the BRCONNECT database check. See condition IN\_WRONG\_TABLESPACE in .

**Action:** Reassign the segment to the correct tablespace using online reorganization with BRSPACE.

The non-standard equivalent to this optional alert is *Segment in wrong tablespace* in [Database Health Alerts \[page 211\]](#).

Rollback segment extension failed [BRCONNECT] [OPTIONAL]

**Cause:** There is the following message in the Oracle alert log:

ORA-01562 Failed to extend rollback segment

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 and PSAPUNDO, this is most likely the tablespace affected.

**Action:** If the rollback segment's NEXT storage parameter is set higher than intended, decrease its value. Otherwise, use BRSPACE to add a data file to the affected tablespace (most likely PSAPROLL or PSAPUNDO).

The non-standard equivalent to this optional alert is *Failed to extend rollback* in [Database Health Alerts \[page 211\]](#).

## 7.2 Database Performance Alerts

### Definition

You might see the alerts described here when [monitoring your Oracle database \[page 202\]](#).

#### i Note

[BRCONNECT] below means that the alert is raised by the .

[OPTIONAL] below means that the alert can be turned off here and shown only in the *Health* monitoring tree. Such alerts always have corresponding *Health* alerts, as described in [Database Health Alerts \[page 211\]](#).

For more information, see [Information on Oracle Database Alerts \[page 229\]](#).

### Structure

#### Optimizer

Last successful update statistics

**Cause:** Threshold is exceeded for number of days since the last successful run of with options `-t all`. Return codes 0 (success) or 1 (success with warning) indicate a successful update statistics.

**Action:** Run `brconnect -f stats -t all`.

Last brconnect -f stats run

**Cause:** Failed last run of , regardless of the options used. For more information, see LAST\_STATS\_FAILED in .

**Action:** Run `brconnect -f stats`.

Harmful statistics [BRCONNECT] [OPTIONAL]

**Cause:** There are tables or indexes that have statistics, although they should not have these (for example, pool and cluster tables).

**Action:** Remove statistics from these tables or indexes. For more information, see .

The non-standard equivalent to this optional alert is *Harmful statistics* in [Database Health Alerts \[page 211\]](#).

Missing statistics [BRCONNECT] [OPTIONAL]

**Cause:** There are tables or indices that do not have any statistics, although they should have these.

**Action:** Create statistics for these tables or indexes. For more information, see .

The non-standard equivalent to this optional alert is *Missing statistics* in [Database Health Alerts \[page 211\]](#).

## Buffers

Buffer cache

**Cause:** The buffer cache hit ratio fell below the configured threshold.

**Action:** 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:** The library cache hit ratio exceeded the configured threshold.

**Action:** 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<DBSID>.ora` parameter `SHARED_POOL_SIZE`. The application should use identical SQL statements whenever possible. For more information, see the Oracle documentation.

Redo log buffer

**Cause:** 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.

**Action:** Increase the size of the redo log buffer by increasing the `init<DBSID>.ora` parameter `log_buffer` until the number of redo entries per redo log space request stops increasing. .

## Locks

Age of oldest exclusive transaction lock

**Cause:** Age of the oldest exclusive transaction lock.

**Action:** Identify why the exclusive transaction lock is so old by identifying which transactions or programs own the lock.

Deadlocked resource [BRCONNECT] [OPTIONAL]

**Cause:** Result of the search in the alert log for the following Oracle message:

```
ORA00060: Deadlock while waiting for resource.
```

**Action:** Identify why the resource is deadlocked by searching for more information in the Oracle trace files.

The non-standard equivalent to this optional alert is *Deadlock waiting for resource* in [Database Health Alerts \[page 211\]](#).

## Checkpoints

Checkpoint not complete [BRCONNECT] [OPTIONAL]

**Cause:** Result of the search in the alert log for the Oracle message: `Checkpoint not complete`. 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.

**Action:** 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.

The non-standard equivalent to this optional alert is *Checkpoint not complete* in [Database Health Alerts \[page 211\]](#).

## 7.3 Database Backup and Restore Alerts

### Definition

You might see the alerts described here when [monitoring your Oracle database \[page 202\]](#).

#### i Note

[BRCONNECT] below means that the alert is raised by the .

[OPTIONAL] below means that the alert can be turned off here and shown only in the *Health* monitoring tree. Such alerts always have corresponding *Health* alerts, as described in [Database Health Alerts \[page 211\]](#).

For more information, see [Information on Oracle Database Alerts \[page 229\]](#).

### Structure

#### Archiving

Last successful archive log backup

**Cause:** The threshold for the number of days since the last successful backup of the offline redo log files, which includes BRARCHIVE return codes of 0 (successful) or 1 (successful but with warnings), has been exceeded.

**Action:** Run BRARCHIVE to back up the offline redo log files. For more information, see .

Last brarchive run

**Cause:** The last BRARCHIVE run has an invalid status. This includes **all** BRARCHIVE runs, regardless of whether the offline redo log files were actually backed up or not. For example, this might include a run to only delete offline redo log files.

For more information, see LAST\_ARCHIVE\_FAILED in .

**Action:** Correct the problem and repeat the BRARCHIVE run. For more information, see .

Archiver destination full

**Cause:** The number of archive logs that will still fit into the archiver destination is below the preconfigured threshold.

**Action:** Either back up and delete offline redo log files with BRARCHIVE or add more space to the archiver destination to accommodate more archive logs.



Archiving off

**Cause:** Your database is running, but online redo logs are not being archived.

**Action:** Using BRSPACE, and .

### Backup Status

Last successful complete database backup

**Cause:** The threshold for the number of days since the last successful complete database backup, which includes BRBACKUP return codes of 0 (successful) or 1 (successful but with warnings), has been exceeded.

**Action:** Run BRBACKUP to perform a complete database backup. For more information, see .

Last brbackup run

**Cause:** The last BRBACKUP run has an invalid status. This includes **all** BRBACKUP runs, regardless of whether a complete database backup was performed. For example, this might include a run to perform an incremental backup.

For more information, see LAST\_BACKUP\_FAILED in .

**Action:** Correct the problem and repeat the BRBACKUP run. For more information, see .

Tablespace in backup mode [BRCONNECT] [OPTIONAL]

**Cause:** The tablespaces named in the alert message are currently in backup mode, which reduces the performance of the SAP System.

**Action:** Confirm that no database backup is running, either with BRBACKUP or an external tool. Then to cancel the backup mode.

The non-standard equivalent to this optional alert is *Tablespace in backup mode* in [Database Health Alerts \[page 211\]](#).

## 7.4 Database Consistency Alerts

### Definition

You might see the alerts described here when [monitoring your Oracle database \[page 202\]](#).

### Structure

#### Consistency Check

**Cause:** Threshold for number of days since the last successful run of the consistency check is exceeded.

**Action:** Run the consistency check. For more information, see [Database System Check \[page 176\]](#).

#### Objects Missing in Database

Primary indexes, Secondary indexes, Tables, Views

**Cause:** 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.

**Action:** Unless there has been either an incomplete transport or a canceled conversion, simply create the object in the database.

### Unknown Objects in ABAP Dictionary

Database tables, Database indexes, Database views, Database tables without unique index

**Cause:** Objects were found in the database that are not defined in the ABAP Dictionary. Since these objects do not belong to the SAP System or cause problems, this list is primarily for information. 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 the SAP System is based on ABAP Dictionary definitions, data can be displayed only via the database.

**Action:** 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 SAP System and that problems such as collisions can be avoided during upgrading.

### Inconsistent Objects

Primary indexes, Secondary indexes, Tables, Views

**Cause:** Objects exhibit inconsistencies between their ABAP Dictionary definitions and their database data dictionary definitions.

**Action:** 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:** 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 SAP 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:** 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.

**Action:** Determine if these indexes have either been created or not been created as intended.

## 7.5 Database Health Alerts

### Definition

You might see the alerts described here when [monitoring your Oracle database \[page 202\]](#). You can see which database objects are causing an alert by looking at the contents of the message container in the corresponding node.

#### i Note

[BRCONNECT] below means that the alert is raised by the .

[NON-STANDARD] below means that the alert has an equivalent alert in another monitoring tree. If required, you can turn off the equivalent alert in the other monitoring tree and only show the alert here.

For more information, see [Information on Oracle Database Alerts \[page 229\]](#).

### Structure

#### Database health check

Last successful database check

**Cause:** The threshold for the number of days since the last successful database check, which includes BRCONNECT return codes of 0 (successful) or 1 (successful but with warnings), has been exceeded.

**Action:** Run .

Last brconnect -f check run

**Cause:** The last run of `brconnect -f check` has an invalid status.

**Action:** Correct the problem and repeat .

Last evaluation of check results

**Cause:** Using information stored in the DBMSGORA table by the last run of , the collection method of this node feeds all others marked with [BRCONNECT].

**Action:** You can start the collection method of this node manually if you want to immediately evaluate the last database check run.

Start of database check

This lets you directly start the database check with .

#### Database administration

The alerts in this section correspond to the checks stored in the DBCHECKORA table for database administration.

Archiver destination full [BRCONNECT] [NON-STANDARD]

**Cause:** The used space in the archiver destination directory is above the preconfigured threshold. For more information, see ARCHIVER\_STUCK in .

**Action:** Either back up and delete offline redo logs using BRARCHIVE or add more space to the archiver destination to accommodate more logs.

The standard equivalent to this alert is *Archiver destination full* in [Database Backup and Restore Alerts \[page 208\]](#).

Control file not mirrored [BRCONNECT]

**Cause:** The control file is not mirrored. For more information, see CONTROL\_FILE\_MIRROR in .

**Action:** Edit the `init<DBSID>.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, to activate your changes.

### i Note

If you only use hardware mirroring, you can inactivate this check. For more information, see [Configuring Database System Check \(Oracle\) \[page 177\]](#).

Missing control file [BRCONNECT]

**Cause:** The control file is missing. For more information, see CONTROL\_FILE\_MISSING in .

**Action:** Look for the parameter CONTROL\_FILES, in the `init<DBSID>.ora` initialization file to determine which directories the control files are stored in. Make sure you can display all the files and that they are available in the file system. Restore access to any non-accessible files.

Critical autoextendable file [BRCONNECT]

**Cause:** There are data files with an activated auto extend feature that would consume the entire remaining free disk space if extended. A check is made to see whether the file system can be brought to overflow, due to the existing parameter setting ( NEXT and MAXSIZE) during the automatic file extension. For more information, see CRITICAL\_FILE in .

**Action:** Add more space to the file system or reduce the autoextend parameters.

Critical segment [BRCONNECT] [NON-STANDARD]

**Cause:** There are segments with too few allocatable extents due to lack of free space in the tablespace. “Too few” is defined in the CHKVAL field of the DBCHECKORA table (range is 1 to 5).

For more information, see CRITICAL\_SEGMENT in .

**Action:** 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 data file to the tablespace.

The standard equivalent to this alert is *Segments with too few allocatable extents* in [Space Management Alerts \[page 203\]](#).

Data file mismatch [BRCONNECT]

**Cause:** There are data files that are flagged as MISSING in the Oracle dictionary.

For more information, see DATA\_FILE\_MISMATCH in .

**Action:** Execute `CREATE CONTROLFILE` using Oracle SQLPLUS, specifying either the correct file or no file at all.

Missing data file [BRCONNECT]

**Cause:** There are data files that no longer exist in the file system.

For more information, see `DATA_FILE_MISSING` in .

**Action:** Perform .

Database file offline [BRCONNECT]

**Cause:** There are data files or online redo log files that are offline.

For more information, see `FILE_OFFLINE` in .

**Action:** Close all applications and enter the following commands using Oracle SQLPLUS:

1. `connect / as sysdba`
2. `shutdown immediate`
3. `startup mount`
4. `alter database data file <data file> online;`
5. `alter database open`

Alternatively, use .

File system full [BRCONNECT]

**Cause:** The fill level of file systems on the database host is too high. All file systems are checked by default against the same threshold value. However, you can define different threshold values for individual file systems by specifying the file system in the `OBJECT` field of the `DBCHECKORA` table (transaction DB17).

For more information, see `FILE_SYSTEM_FULL` in .

**Action:** Increase the amount of space in the file system.

Harmful statistics [BRCONNECT] [NON-STANDARD]

**Cause:** There are tables or indexes that have statistics, although they should not have these (for example, pool and cluster tables).

For more information, see `HARMFUL_STATISTICS` in .

**Action:** Remove statistics from these tables or indexes. For more information, see .

The standard equivalent to this alert is *Harmful statistics* in [Space Management Alerts \[page 203\]](#).

Invalid file type [BRCONNECT]

**Cause:** There are database files that have an illegal operating system type, for example, block raw files on UNIX or compressed files on Windows.

For more information, see `INVALID_FILE_TYPE` in .

**Action:** Troubleshoot to determine the cause of the problem. If you are not sure how to proceed, contact SAP technical support.

Segment in wrong tablespace [BRCONNECT] [NON-STANDARD]

**Cause:** A segment is in the wrong tablespace, as raised by the BRCONNECT database check. For more information, see `IN_WRONG_TABLESPACE` in .

**Action:** Use `ALTER TABLE`, which moves the tables into a tablespace of the correct type.

The standard equivalent to this alert is *Segment in wrong tablespace* in [Space Management Alerts \[page 203\]](#).

Missing index [BRCONNECT] [NON-STANDARD]

**Cause:** There are tables that do not have any indexes and are not specified in the DBDIFF table exception.

For more information, see `MISSING_INDEX` in .

**Action:** Create the missing indexes using transaction SE11.

The standard equivalents to this alert in [Database Consistency Alerts \[page 209\]](#) are:

- *Objects missing in database: Primary indexes*
- *Objects missing in database: Secondary indexes*
- *Unknown objects in ABAP dictionary: Tables without unique index*

Missing statistics [BRCONNECT] [NON-STANDARD]

**Cause:** There are tables or indexes that do not have any statistics, although they should have these.

For more information, see `MISSING_STATISTICS` in .

**Action:** Create statistics for these tables or indexes. For more information, see .

The standard equivalent to this alert is *Missing Statistics* in [Database Performance Alerts \[page 206\]](#).

Database in NOARCHIVELOG mode [BRCONNECT] [NON-STANDARD]

**Cause:** The database is running, but online redo logs are **not** being archived.

For more information, see `NOARCHIVELOG_MODE` in .

**Action:** Using `BRSPACE`, and .

The standard equivalent to this alert is *Archiving Off* in [Database Backup and Restore Alerts \[page 208\]](#).

Segment with non-zero PCTINCREASE [BRCONNECT] [NON-STANDARD]

**Cause:** Segments have a `PCTINCREASE` value other than zero. Since 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 the SAP schema are considered.

For more information, see `PCTINCREASE_NOT_ZERO` in .

**Action:** Evaluate whether the segment's storage parameter can be set to zero.

The standard equivalent to this alert is *Segments with non-zero PCTINCREASE* in [Space Management Alerts \[page 203\]](#).

Redolog file not mirrored [BRCONNECT]

**Cause:** The online redo logs are not mirrored. Mirroring is required for safe operation. For more information, see `REDOLOG_FILE_MIRROR` in

**Action:** Using Oracle SQLPLUS, activate mirroring of the online redo logs in your database, with the statement `ALTER DATABASE ADD LOGFILE MEMBER`.

Missing redolog file [BRCONNECT]

**Cause:** An online redo log is missing, which reduces the effect of mirroring. Mirroring is required for safe operation. For more information, see `REDOLOG_FILE_MISSING` in .

**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 might 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 Oracle SQLPLUS, execute:

#### **ALTER SYSTEM SWITCH LOGFILE**

Then, check if the status of the online redo logs in the table `V$logfile` is correct, and execute:

#### **ALTER DATABASE ADD LOGFILE MEMBER**

Tablespace full [BRCONNECT] [NON-STANDARD]

**Cause:** A tablespace is full. All tablespaces are checked by default against the same threshold value. However, you can define different threshold values for individual tablespaces by specifying the tablespace name in the `OBJECT` field of the `DBCHECKORA` table, using transaction DB17.

For more information, see `TABLESPACE_FULL` in .

**Action:** , if necessary, based on your estimate of anticipated growth.

Tablespace in backup mode [BRCONNECT] [NON-STANDARD]

**Cause:** A tablespace is in backup mode, which reduces the performance of the SAP System. For more information, see `TABLESPACE_IN_BACKUP` in .

**Action:** Confirm that no database backup is running, either with `BRBACKUP` or an external tool. Then cancel the backup mode by using `BRSPACE` to .

The standard equivalent to this alert is *Tablespace in backup mode* in [Database Backup and Restore Alerts \[page 208\]](#).

Tablespace offline [BRCONNECT]

**Cause:** A tablespace is offline. For more information, see `TABLESPACE_OFFLINE` in .

**Action:** Use `BRSPACE` to .

Segment with too many extents [BRCONNECT] [NON-STANDARD]

**Cause:** There are tables or indexes, for which the number of allocated extents exceeds the specified threshold value. All tables or indices are checked by default against the same threshold value. However, you can define different threshold values for individual tablespaces by specifying the tablespace name in the `OBJECT` field of the `DBCHECKORA` table (transaction DB17).

For more information, see `T00_MANY_EXTENTS` in .

**Action:** Increase the `MAXEXTENTS` limit for the table or index using SQLPLUS with one of the following commands:

```
ALTER TABLE <name> STORAGE (MAXEXTENTS <number>);
```

**ALTER INDEX <name> STORAGE (MAXEXTENTS <number>);**

If you prefer segments with fewer allocated extents, you can use BRSPACE to .

Alternatively, use the `-1|limit` option of .

The standard equivalent to this alert is *Segments approaching MAX\_EXTENTS* in [Space Management Alerts \[page 203\]](#).

## DBA Operations

The alerts in this section correspond to the checks stored in the DBCHECKORA table for database operations.

Last successful archive log backup too old [BRCONNECT] [NON-STANDARD]

**Cause:** The last successful backup of the offline redo log files with BRARCHIVE is too old. For more information, see ARCHIVE\_TOO\_OLD in .

**Action:** Perform an .

The standard equivalent to this alert is *Last successful archive log backup* in [Database Backup and Restore Alerts \[page 208\]](#).

Last successful complete database backup too old [BRCONNECT] [NON-STANDARD]

**Cause:** The last successful complete backup of the database with BRBACKUP is too old. For more information, see BACKUP\_TOO\_OLD in .

**Action:** Perform a .

The standard equivalent to this alert is *Last successful complete database backup* in [Database Backup and Restore Alerts \[page 208\]](#).

Last archive log backup failed [BRCONNECT] [NON-STANDARD]

**Cause:** The last backup of the offline redo logs with BRARCHIVE has an invalid status. For more information, see LAST\_ARCHIVE\_FAILED in .

**Action:** Correct the problem and repeat the BRARCHIVE run. For more information, see .

The standard equivalent to this alert is *Last brarchive run* in [Database Backup and Restore Alerts \[page 208\]](#).

Last complete database backup failed [BRCONNECT] [NON-STANDARD]

**Cause:** The last complete database backup with BRBACKUP has an invalid status. For more information, see LAST\_BACKUP\_FAILED in .

**Action:** Correct the problem and repeat the BRBACKUP run. For more information, see .

The standard equivalent to this alert is *Last brbackup run* in [Database Backup and Restore Alerts \[page 208\]](#).

Last operation failed [BRCONNECT]

**Cause:** The last DBA operation failed, which is identified by a function ID specified in the OBJECT field of the DBCHECKORA table. For more information, see LAST\_OPERATION\_FAILED in .

**Action:** Correct the problem and repeat the operation. For more information, see the corresponding function in .

Last update statistics failed [BRCONNECT] [NON-STANDARD]



**Cause:** The last update of the optimizer statistics with BRCONNECT failed. For more information, see LAST\_STATS\_FAILED in .

**Action:** Correct the problem and repeat . For more information, see .

The standard equivalent to this alert is *Last brconnect -f stats run* in [Database Performance Alerts \[page 206\]](#).

Last successful operation too old [BRCONNECT]

**Cause:** The last successful DBA operation, identified by a function ID specified in the OBJECT field of the DBCHECKORA table, is too old.. For more information, see OPERATION\_TOO\_OLD in .

**Action:** Correct the problem and repeat the operation. For more information, see the corresponding function in the .

Last successful update statistics too old [BRCONNECT] [NON\_STANDARD]

**Cause:** The last successful update statistics run is too old. For more information, see STATS\_TOO\_OLD in .

**Action:** Correct the problem and repeat . For more information, see .

The standard equivalent to this alert is *Last successful update statistics* in [Database Performance Alerts \[page 206\]](#).

## Oracle Messages

The alerts below correspond to the DBCHECKORA check conditions for database messages in the Oracle alert log. You can find the complete messages with time stamp and text in the message container of the corresponding Monitoring Tree Element (MTE) in the alert monitor.

Deadlock while waiting for resource [BRCONNECT] [NON\_STANDARD]

**Cause:** There is the following message in the Oracle alert log:

ORA-00060: Deadlock detected while waiting for resource

**Action:** Check the Oracle log / trace file.

The standard equivalent to this alert is *Deadlocked resource* in [Database Performance Alerts \[page 206\]](#).

Error archiving log file [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00255: Error archiving log

**Action:** Check the archive directory.

Archiver error [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00257: Archiver error

**Action:** Check the archive directory.

Error creating archive log [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00270: Error creating archive log

**Action:** Check to determine whether the storage medium is functioning. Also check whether enough storage space is available.

Error writing archive log [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00272: Error writing archive log

**Action:** Check to determine whether the storage medium is functioning. Also check whether enough storage space is available.

Database file cannot be read [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00376: File cannot be read at this time

**Action:** 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.

Fatal error in background process [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00447: Fatal error in background process

**Action:** Check the Oracle alert log / trace file.

LGWR process terminated with error [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00470: LGWR process terminated with error

**Action:** Check the Oracle alert log / trace file.

DBWR process terminated with error [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00471: LGWR process terminated with error

**Action:** Check the Oracle alert log / trace file.

PMON process terminated with error [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00472: PMON process terminated with error

**Action:** Check the Oracle alert log / trace file.

ARCH process terminated with error [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00473: ARCH process terminated with error

**Action:** Check the Oracle alert log / trace file.

SMON process terminated with error [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00474: SMON process terminated with error

**Action:** Check the Oracle alert log / trace file.

Oracle internal error [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00600: Internal error code

**Action:** Contact Oracle or SAP technical support.

Oracle server session terminated by fatal error [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-00603: ORACLE server session terminated by fatal error

**Action:** Check the Oracle alert log / trace file.

I/O error writing database file [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01114: IO error writing block to file **Action:** Check to determine whether the storage medium is functioning. Also check whether enough storage space is available.

I/O error reading database file [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01115: IO error reading block from file

**Action:** Check to determine whether the storage medium is functioning.

File verification check failed [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01122: Database file failed verification check

**Action:** Make sure you are using the correct data file and control file for the database and check the last recovery or database copy.

Cannot shutdown - data file in backup mode [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01149: Cannot shutdown - file has online backup set

**Action:** Terminate the backup.

Snapshot too old [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01555: Snapshot too old

**Action:** Change the configuration of the rollback segments to enlarge them. Only execute this operation when the database load is low.

Failed to extend rollback [BRCONNECT] [NON\_STANDARD]

**Cause:** There is the following message in the Oracle alert log:

ORA-01562: Failed to extend rollback segment

**Action:** Use BRSPACE to PSAPROLL or PSAPUNDO.

The standard equivalent to this alert is *Rollback segment extension failed* in [Space Management Alerts \[page 203\]](#).

Database block corrupted [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01578: ORACLE data block corrupted

**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. For more information, see .

Max. extents reached for rollback segment [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01628: Max extents reached for rollback segment

**Action:** Increase the MAXEXTENTS value.

Max. extents reached saving undo for tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01629: Max extents reached saving undo for tablespace

**Action:** Increase the MAXEXTENTS value.

Max. extents reached in temp segment in tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01630: Max. extents reached in temp segment in tablespace

**Action:** Increase the MAXEXTENTS value.

Max. extents reached in table [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01631: Max. extents reached in table

**Action:** Increase the MAXEXTENTS value.

Max. extents reached in index [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01632: Max. extents reached in index

**Action:** Increase the MAXEXTENTS value.

Unable to extend rollback segment in tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01650: Unable to extend rollback segment in tablespace

**Action:** Extend the tablespace.

Unable to extend save undo segment for tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01651: Unable to extend save undo segment for tablespace

**Action:** Extend the tablespace.

Unable to extend temp segment in tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01652: Unable to extend temp segment in tablespace

**Action:** Extend the tablespace.

Unable to extend table in tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01653: Unable to extend table in tablespace

**Action:** Extend the tablespace.

Unable to extend index in tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01654: Unable to extend index in tablespace

**Action:** Extend the tablespace.

Unable to extend cluster in tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01655: Unable to extend cluster in tablespace

**Action:** Extend the tablespace.

Max. extents reached in cluster [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01656: Max. extents reached in cluster

**Action:**

Unable to extend lob segment in tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01680: Unable to extend LOG segment in tablespace

**Action:** Extend the tablespace.

Max. extents reached in LOB segment in tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01681: Max. extents reached in LOB segment in tablespace

**Action:** Increase the MAXEXTENTS value.

Unable to extend index partition [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01683: Unable to extend index partition

**Action:** Extend the tablespace.

Max. extents reached in table partition [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01684: Max. extents reached in table partition

**Action:** Increase the MAXEXTENTS value.

Max. extents reached in index partition [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01685: Max. extents reached in index partition

**Action:** Increase the MAXEXTENTS value.

Unable to extend table partition [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01688: Unable to extend table partition

**Action:** Extend the tablespace.

Unable to extend lob segment in tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01691: Unable to extend lob segment in tablespace

**Action:** Extend the tablespace.

Unable to extend lob segment partition in tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01692: Unable to extend lob segment partition in tablespace

**Action:** Extend the tablespace.

Max. extents reached in lob segment [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01693: Max. extents reached in lob segment

**Action:** Increase the MAXEXTENTS value.

Max. extents reached in lob segment partition [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-01694: Max. extents reached in lob segment partition

**Action:** Increase the MAXEXTENTS value.

Exception encountered: core dump [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-07445: Exception encountered: core dump

**Action:** Call the hotline.

Destination not available for archiving log file [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-16014: Log sequence not archived, no available destination s

**Action:** Check the archive destination.

Log file cannot be archived [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-16038: Log sequence cannot be archived

**Action:** Check the archive destination.

Write error on log file [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-19502: Write error on file

**Action:** Check the archive destination.

Failed to create log file [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-19504: Failed to create file

**Action:** Check the archive destination.

Failed to set size of blocks for file [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-19510: Failed to set size of blocks for file

**Action:** Check the disk.

Unable to write the header block of file [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-27044: Unable to write the header block of file

**Action:** Check the disk.

Number of bytes read/written is incorrect [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-27063: Number of bytes read/written is incorrect

**Action:** Check the disk.

File I/O error [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-27072: I/O error

**Action:** Check the disk.

Unable to extend segment in undo tablespace [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

ORA-30036: Unable to extend segment in undo tablespace

**Action:** Extend the tablespace.

Checkpoint not complete [BRCONNECT] [NON-STANDARD]

**Cause:** There is the following message in the Oracle alert log:

Checkpoint not complete

**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.

The standard equivalent to this alert is *Checkpoint not complete* in [Database Performance Alerts \[page 206\]](#).

Corrupt block [BRCONNECT]

**Cause:** There is the following message in the Oracle alert log:

Corrupt block: Database block is corrupted

**Action:** Check the consistency of database blocks.

### Profile Parameters

The alerts in this section correspond to the checks stored in the DBCHECKORA table for the database profile. The parameters and the thresholds are contained in the DBCHECKORA table, and correspond to the SAP recommendations on delivery. However, we might later change the recommendations.

**Only Oracle 10g profile parameters** are shown here. For parameters relating to earlier Oracle releases, see previous versions of this documentation.

#### → Recommendation

If there are new recommendations, as specified in SAP Notes 124361 and 180605, change the DBCHECKORA table accordingly.

compatible [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Database for compatibility with Oracle 10

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

control\_file\_record\_keep\_time [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).



Retention time for information on RMAN backups in the control file

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

cursor\_space\_for\_time [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in transaction DB17x.

Database system uses more space for cursors

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in [transaction DB17 \[page 177\]](#).

db\_block\_buffers [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Number of blocks in the database buffer cache

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

db\_block\_size [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Size (bytes) of the Oracle database blocks

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

db\_files [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Maximum number of data files

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

db\_file\_multiblock\_read\_count [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Number of blocks read during a complete table scan I/O operation

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

filesystemio\_options [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

I/O operations on file system files

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

log\_archive\_start [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

The setting LOG\_ARCHIVE\_START = TRUE means that automatic archiving is activated when the Oracle instance is started.

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

log\_buffer [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Size (bytes) of the redo log buffer in the SGA

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

log\_checkpoints\_to\_alert [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Log checkpoint begin/end to alert file

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

max\_dump\_file\_size [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Maximum size of the dump file

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

open\_cursors [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Maximum number of open cursors per user process

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

optimizer\_mode [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Controls how the optimizer behaves (for example, rule-based = RULE, or cost-based = CHOOSE).

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

parallel\_execution\_message\_size [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Message buffer size for parallel execution

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

parallel\_threads\_per\_cpu [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Number of parallel execution threads per CPU

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

recyclebin [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Disable recyclebin processing

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

remote\_os\_authent [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Allow remote clients to use auto-logon accounts

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

replication\_dependency\_tracking [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Tracking dependency for replication parallel propagation

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

shared\_pool\_size [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Size in bytes of the shared pool in the System Global Area (SGA)

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

star\_transformation\_enabled [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Enable the use of star transformation

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

statistics\_level [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Statistics level

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

timed\_statistics [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Generation of time-related statistics

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

trace\_enabled [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Enable tracing

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

undo\_management [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Enable automatic undo management

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

undo\_tablespace [BRCONNECT]

**Cause:** Current setting is outside the range of values preconfigured for this check in [transaction DB17 \[page 177\]](#).

Undo tablespace name

**Action:** Verify that this value is set as intended. If so, readjust the preconfigured range of values in transaction DB17.

## 7.6 Information on Oracle Database Alerts

### Use

Some monitoring tree elements (MTEs) for alerts when you [monitor the Oracle database \[page 202\]](#), especially [Database Health Alerts \[page 211\]](#), are marked in this documentation as follows:

- BRCONNECT

These MTEs are based on the most recent , using the command `brconnect -f check`. The conditions for the corresponding checks are specified in the DBCHECKORA table. If required, you can [configure the database system check \[page 177\]](#).

For more information on the check conditions, see [SAP Note 435290](#).

BRCONNECT stores the results of the checks in the DBMSGORA table. The MTE [Last evaluation of check results](#) evaluates the results of the most recent BRCONNECT checks and, if required, generates alerts in the corresponding MTE nodes.

You can also run a one-off database system check using the MTE [Start of database check](#) in the [Health](#) monitoring tree to immediately start a new database system check and update the database monitor.

You can also view the results of the database check directly, without using the database alert monitor. For more information, see [Displaying Alert Messages from Database System Check \[page 182\]](#).

The BRCONNECT MTEs are dynamically set up during the initialization of the monitoring infrastructure, based on the check conditions set up in the DBCHECKORA table. If one of these MTEs does not have a corresponding check condition in the DBCHECKORA table (because the check is inactive), it does not appear in the Oracle database monitor.

Therefore, these MTEs can dynamically appear and disappear, depending on whether the corresponding check conditions exist. The MTE always appears at the relevant position in the monitoring tree, as specified in this documentation.

You can set up new check conditions in the following categories:

- DBO: Concrete database operations (function ID)
- ORA: Oracle messages (message code, message text)
- PROF: Oracle profile parameters (parameter names)

You can only set up new check conditions for database administration when the corresponding BRCONNECT functionality is available. For conditions that do not use the OBJECT field in the DBCHECKORA table, you can set up your own name for the corresponding MTE. To do this, specify **#RZ20** followed by the new name, as shown in the following example:

#### Example

This is an example of how to specify a new dynamic MTE:

*TYPE* field: **DBO**

*PARAM* field: **DBV**

*OBJECT* field: **#RZ20 Database verification**

The result is that a new MTE called *Database verification* is set up in the Oracle database monitor under [Health > DBA operations](#).

For more information on the BRCONNECT check conditions, see:

- 
- 

Then do the following to re-initialize the Oracle context for database alert monitoring in the RZ20 transaction, so as to incorporate your new check conditions:

1. Find out on which application servers the Oracle context exists by calling transaction RZ21 and entering [Topology > Context Overview > Segment Name](#), then [Topology > Segment Overview > Destination](#).
  2. Log on to the application server where the Oracle context exists.
  3. Start the Oracle database monitor by choosing [Administration > CCMS > ControlMonitoring > Alert-Monitor > SAP CCMS Monitor Templates > Database](#), or call transaction RZ20.
  4. Choose [Extras > Activate Maintenance Function](#).
  5. Select *Oracle* and reset all alerts by choosing [Edit > Nodes \(MTE\) > Reset](#).
  6. Select *Oracle* and delete the Oracle tree by choosing [Edit > Nodes \(MTE\) > Delete](#).
  7. Recreate the Oracle context on this application server by calling transaction SE38 in and entering **RSDBMON0** in *Program* then choosing *Execute*.
- NON-STANDARD and OPTIONAL  
NON-STANDARD MTEs from the *Health* monitoring tree are not set up in the standard database monitor because identical or similar MTEs are already present at another point in the monitoring tree. However, if you want to set up the NON-STANDARD but not the OPTIONAL MTEs, use SQLPLUS to add the following row to the SORAMON table:

```
connect sapr3/<password> or connect sap<sid>/<password>  
insert into soramon (category, timestamp, value)  
values ('SYSCHECK ALL_COND', '20020101000000', 1)
```

Then re-initialize the Oracle context as described above.

The result is that all NON-STANDARD alerts are reported only in the *Health* part of the database monitoring tree.

You can also have the alerts reported at both points of the monitoring tree by using SQLPLUS to add the following row to the SORAMON table:



```
connect sapr3/<password> or connect sap<sid>/<password>  
insert into soramon (category, timestamp, value)  
values ('SYSCHECK ALL_COND', '20020101000000', 2)
```

# Important Disclaimers and Legal Information

## Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

## Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

## Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

## Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

## Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2023 SAP SE or an SAP affiliate company. All rights reserved.

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

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.