Database Administration Using the DBA Cockpit: Oracle

SAP NetWeaver 7.1
February 2008

Copyright

Icons in Body Text

Icon	Meaning
⚠	Caution
❖	Example
💡	Note
⬆	Recommendation
<>	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help* → *General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

Typographic Conventions

Type Style	Description
Example text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.
`Example text`	Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, `F2` or `ENTER`.

The DBA Cockpit

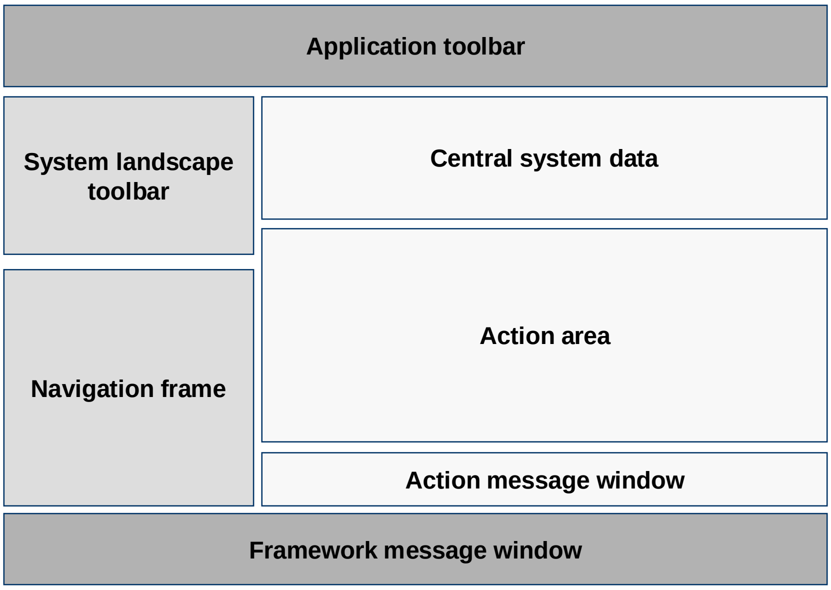
Use

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 all 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 or 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*.

    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.

Central System Data

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

Where relevant, you can select the following in central system data:

- *Save Data*

  You can create and save a data snapshot.

- *Confirm*

  You can confirm a new selection for *Since* and *Up to* (see end of table below), which updates the data displayed accordingly.

Structure

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

Field	Description
DB Name	Database name
DB Server	Database server name
DB Release	Database release number
Started	Date and time when database was started
Since and *Up to*	Start and finish date and time for the period of the displayed data You can select different time periods. The default is *Since Database Start* and *Up to Now*.

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

> The following **only** applies to Oracle and IBM DB2 for Linux, UNIX, and Windows.

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.

For more information, see *Displaying the Audit Log*.

Displaying the Audit Log

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
Date	Start date of the action

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

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 current status of a system. You can change the status of a system by choosing *Stop* or *Go*.

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

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 *Maintaining Database Connections*.

- 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.

  - **MS SQL Server only:**

    For a database connection that is localized in another ABAP system. That is, the DBA Cockpit can use the RFC connection together with the database connection.

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

See also:

Configuring Systems for Remote Monitoring Using Remote Database Connections

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

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.

> 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

- The system(s) you want to monitor must have a database release that is compatible with the database release of your local database.

- 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 *Stop*, *Go*, or *Inactive* icon, which shows the current system status.

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:

   a) In the *System* field, enter the name of the system you want to monitor.

   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.

   b) Select *Database Connection*.

   c) Enter the name of the database connection. If the database connection does not yet exist, you are directed to the *System Administration Details* screen where you can specify all relevant data for the new connection.

   After you have saved your entries, you are redirected to the screen *DB Connections: Add Connection Entry*.

   For more information, see *Maintaining the Database Connection*.

   d) 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.

   e) If required, enter the name of the RFC connection.

   After the connection data has been completed, it is displayed on the *System Data* tab page. You can enter additional data on the *Administration Data* tab page as follows:

   ○ Enter a description of your system.

   ○ Depending on the database platform, select the options for how you want to collect monitoring data:

   ■ If alerts are to be provided for the RZ20 alert monitor, select *Collect Alert Data*.

   ■ If data about the performance or the size of database objects is to be collected, select *Collect History Data*.

   ■ If the task of collecting monitoring data is running on the remote system, select *Data Collection by Remote System*.

   ■ If data for the central planning calendar is to be provided, select *Collect Central Planning Calendar Data*.

4. 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*.

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
Remote Database Connections	Name of database connection This is a unique name that you can freely choose.
DB Name	Name of database
DB Host	Name of database host
DB Schema	Name of the database schema to be monitored
User	Name of the connection user
Permanent	Specifies whether the connect user must be permanently available
Max. Connections	Maximum allowed number of open connections
Opt. Connections	Optimal number of connections

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

   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.

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:

- *Adding a Database Connection*
- *Changing a Database Connection*
- *Deleting a Database Connection*
- *Testing a Database Connection*

Adding a Database Connection

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.

   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).

   End of the note.
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.

    End of the caution.

    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
TNS Name	Name of Oracle Transparent Network Substrate (TNS)
Character Set	Name of character set
National Character Set	Name of national character set

7. Note

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

9. 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.

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

11. End of the note.

12. 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.

Changing a Database Connection

1. Call the DBA Cockpit.
2. In the system landscape toolbar, choose *DB 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 *Adding a Database Connection*.

5. Save your changes.

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 *DB Connections*.
3. Select a system and choose *Test*.

   The result is displayed in the message window below.

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*.

   ⚠

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

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.

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.

   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 *Configuration of Database Connections*.

Performance (Oracle)

You can use the *Performance* menu in the DBA Cockpit to monitor your Oracle database:

- Performance overview
- Wait event analysis

- SQL statement analysis
- Statistical information
- Feature monitoring
- Additional functions
- RAC statistics

SAP/Oracle Database Monitor

You can use the SAP/Oracle database monitor in the DBA Cockpit to monitor your database running with Oracle 9i or Oracle 9i Real Application Cluster (RAC) or later. It is an *expert* tool.

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).

End of the note.

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

To generate history information for the monitor, you must have planned the jobs RSORAHIST or RSORAHCL using transaction SM36.

To display full historical information, you must not have restarted the database during the relevant period.

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 with an overview of the database
  - Detailed analyses using sub-monitors, grouped as follows:
    - Wait event analysis
    - SQL statement analysis
    - Statistical information

    - Feature monitoring
    - Additional functions
    - RAC statistics
- Fully supports Oracle Real Application Cluster (RAC)
- Does *not* support Multiple Components in One Database (MCOD)

Activities

1. Start the database monitor from the DBA Cockpit by choosing *Performance* ▶ *Performance Overview*.

   Note

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

   End of the note.

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 see monitoring information for a specific history period, in *Selected History* you double-click *Since* and *Up to* and set them as required.

   For more information, see Viewing History Information.

4. To refresh the display, choose *Refresh*.

Performance Overview

This is the main screen in the SAP/Oracle Database Monitor. It gives you an overview of the performance of your Oracle database.

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:

- *Help* for more information on the meaning of the field
- *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.

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. The limit values are maintained in table ST04N_LIM.

End of the note.

Structure

The fields are grouped as follows:

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

Field	Description
DB instance	Name of the current database instance This is the SAP-SID in non-RAC environments.
DB node	Host name of the selected DB node This is the database server in non-RAC environments.
DB release	Release of the current database
Day, time	Current day and time
Start up at	Date and time when the current database instance started
Sec. since start	Seconds since start of the current database instance
Sec. btw. snaps.	Seconds between snapshots

- *Data buffer*

Field	Description
Size	Size of the data buffer in KB o Non-RAC or RAC detail: data buffer size of the current instance o RAC total: total instance-related database buffers Data source: V$SGA
Quality	Data buffer quality, calculated as follows: 100% – ((`physicalreads` – `physicalreads_direct` – `physicalreads_directlob`) / (`sess_logicalreads` – `physicalreads_direct` – `physicalreads_directlob`)) Non-RAC or RAC detail: data buffer quality of the current instance RAC total: average of quality for all instance-related data buffer Data source: V$SYSSTAT
Size default pool	Size of the default buffer pool in KB, calculated as follows: `select value from v$parameter where name = 'db_cache_size'` Data source: V$PARAMETER

Field	Description
Size keep pool	Size of the keep buffer pool in KB, calculated as follows: `select value from v$parameter where name = 'db_keep_cache_size'` Data source: V$PARAMETER
Size others	Sum of the sizes of all other buffer pools in KB, calculated as follows: `select (a.sum_a - b.sum_b) as buff_others from (select sum(value) as sum_a from v$parameter where name = 'db_cache_size' or name = 'db_keep_cache_size' or name = 'db_2k_cache_size' or name = 'db_4k_cache_size' or name = 'db_8k_cache_size' or name = 'db_16k_cache_size' or name = 'db_32k_cache_size' or name = 'db_recycle_cache_size') a, (select sum(value) as sum_b from v$parameter where name = 'db_cache_size' or name = 'db_keep_cache_size') b` Data source: V$PARAMETER
Logical reads	Number of logical read operations o Non-RAC or RAC detail: logical reads of the current instance o RAC total: total logical read operations for all instances Data source: V$SYSSTAT
Physical reads	Number of physical read operations o Non-RAC or RAC detail: physical reads of the current instance o RAC total: total physical read for all instances Data source: V$SYSSTAT
Physical writes	Number of physical write operations o Non-RAC or RAC detail: physical writes of the current instance o RAC total: total physical write operations for all instances Data source: V$SYSSTAT
Buffer busy waits	Number of buffer busy wait situations o Non-RAC or RAC detail: total waitstat counters of the current instance o RAC total: total waitstat counters for all instances Data source: V$WAITSTAT

Field	Description
Buffer wait times (s)	Sum of buffer busy wait times o Non-RAC or RAC detail: sum of wait times for all wait counters of the current instance o RAC total: sum of wait times for all wait counters for all instances Data source: V$WAITSTAT

- *Shared pool*

Field	Description
Size (kB)	Shared pool size in KB o Non-RAC or RAC detail: shared pool size of the current instance o RAC total: total shared pool size for all instances Data source: V$SGA_DYNAMIC_COMPONENTS
DD-cache quality (%)	Data dictionary cache quality as percentage, calculated as follows: 100% – (`totalget_misses` / `totalgets`) o Non-RAC or RAC detail: data buffer quality of the current instance o RAC total: average cache quality for all instances Data source: V$ROWCACHE
SQL area getratio (%)	Ratio of `gethits` to `gets` as a percentage, calculated as follows: sum (`gethits`) / sum (`gets`) x 100 o Non-RAC or RAC detail: get ratio of the current instance o RAC total: average of all instance-related get ratios Data source: V$LIBRARYCACHE
SQL area pinratio (%)	Ratio of `pinhits` to `pins` as a percentage, calculated as follows: sum (`pinhits`) / sum (`pins`) x 100 o Non-RAC or RAC detail: pin ratio of the current instance o RAC total: average of all instance-related pin ratios

Field	Description
	Data source: V$LIBRARYCACHE
SQLA Reloads/pins (%)	Ratio of `reloads` to `pins` as a percentage, calculated as follows: sum (`reloads`) / sum (`pins`) x 100 o Non-RAC or RAC detail: reloads per pin ratio of the current instance o RAC total: average of all instance-related pin ratios Data source: V$LIBRARYCACHE

- *Log buffer*

Field	Description
Size (kB)	Size of the redo log buffer in KB o Non-RAC or RAC detail: redo log buffer size of the current instance o RAC total: total instance-related redo log buffer sizes Data source: V$SGA
Entries	Number of redo log buffer entries o Non-RAC or RAC detail: redo log buffer entries of the current instance o RAC total: sum of all instance-related redo log buffer entries Data source: V$SYSSTAT
Allocation retries	Number of redo buffer allocation retries o Non-RAC or RAC detail: allocation retries of the current instance o RAC total: total instance-related log buffer allocation retries Data source: V$SYSSTAT
Alloc fault rate (%)	Redo buffer allocation retries as a percentage of redo entries o Non-RAC or RAC detail: allocation fault rate of the current instance o RAC total: average of all instance-related allocation fault rates Data source: V$SYSSTAT

Field	Description
Redo log wait (s)	Redo log wait in seconds o Non-RAC or RAC detail: Redo log wait time of the current instance o RAC total: sum of instance-related redo log wait times Data source: V$SYSSTAT
Log files (in use)	Number of active log files o Non-RAC or RAC detail: active log files for the current instance o RAC total: total active log files The figure in brackets refers to the number of active log files in use. Data source: V$LOGFILE

- *Calls*

Field	Description
User calls	Number of user calls o Non-RAC or RAC detail: user calls for the current instance o RAC total: total user calls for all instances Data source: V$SYSSTAT
User commits	Number of user commits o Non-RAC or RAC detail: user commits for the current instance o RAC total: total user commits for all instances Data source: V$SYSSTAT
User rollbacks	Number of user rollbacks o Non-RAC or RAC detail: user rollbacks for the current instance o RAC total: total user rollbacks for all instances Data source: V$SYSSTAT

- *Time Statistics*

Field	Description

Field	Description
Busy wait time (s)	Busy wait time in seconds, calculated as the sum of the time waited for all non-idle events. o Non-RAC or RAC detail: busy wait time for the current instance o RAC total: total instance-related busy wait times Data source: V$SESSION_EVENT
CPU time session (s)	CPU time session in seconds, calculated as sum of CPU time used by this session o Non-RAC or RAC detail: total CPU time for the current instance o RAC total: total CPU time for all instances Data source: V$SYSSTAT
Time/User call (ms)	Time for each user call in milliseconds, calculated as follows: (busy wait time + CPU time) / user calls o Non-RAC or RAC detail: time for the current instance o RAC total: average time for all instances Data source: V$SESSION_EVENT and V$SYSSTAT
Sessions busy (%)	Busy sessions as a percentage, calculated as follows: (busy wait time + CPU time) / total wait time o Non-RAC or RAC detail: percentage for the current instance o RAC total: average percentage for all instances Data source: V$SESSION_EVENT and V$SYSSTAT
CPU usage (%)	CPU usage as a percentage, calculated as follows: CPU time / (elapsed time x CPU count) o Non-RAC or RAC detail: percentage for the current instance o RAC total: average percentage for all instances Data source: V$SYSSTAT, V$INSTANCE, and V$PARAMETER
Number of CPUs	Number of CPUs o Non-RAC or RAC detail: CPUs for the current instance

Field	Description
	o RAC total: total CPUs for all instances Data source: V$PARAMETER

- *Redo Logging*

Field	Description
Redo writes	Number of redo log writes o Non-RAC or RAC detail: redo log writes for the current instance o RAC total: total redo log writes for all instances Data source: V$SYSSTAT
OS blocks written	Number of operating system redo blocks written o Non-RAC or RAC detail: redo blocks written for the current instance o RAC total: sum of written redo blocks for all instances Data source: V$SYSSTAT
Latching time (s)	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$SYSSTAT
Redo write time (s)	Redo write time in seconds o Non-RAC or RAC detail: redo write time for the current instance o RAC total: total redo write time for all instances Data source: V$SYSSTAT
MB written	Number of MB written o Non-RAC or RAC detail: redo log data written for the current instance o RAC total: total redo log data written for all instances Data source: V$SYSSTAT

- *Table scans and fetches*

Field	Description

Field	Description
Short table scans	Number of short table scans o Non-RAC or RAC detail: short table scans for the current instance o RAC total: total short table scans for all instances Data source: V$SYSSTAT
Long table scans	Number of long table scans o Non-RAC or RAC detail: long table scans for the current instance o RAC total: total long table scans for all instances Data source: V$SYSSTAT
Table fetch by rowid	Number of table fetches by row ID o Non-RAC or RAC detail: table fetches by row ID for the current instance o RAC total: total table fetches by row ID for all instances Data source: V$SYSSTAT
Fetch by contin. row	Number of fetches by continued row o Non-RAC or RAC detail: table fetches by continued row for the current instance o RAC total: total table fetches by continued row for all instances Data source: V$SYSSTAT

- *Sorts*

Field	Description
Sorts (memory)	Number of sorts in memory o Non-RAC or RAC detail: sorts in memory for the current instance o RAC total: total sorts in memory for all instances Data source: V$SYSSTAT
Sorts (disk)	Number of sorts in disk o Non-RAC or RAC detail: sorts on disk for the current instance o RAC total: total sorts on disk for all instances

Field	Description
	Data source: V$SYSSTAT
Sorts (rows)	Number of sorted rows o Non-RAC or RAC detail: sorted rows in memory for the current instance o RAC total: total rows for all instances Data source: V$SYSSTAT
WA exec. optim. mode	Number of work area executions in optimal mode o Non-RAC or RAC detail: work area executions in optimal mode for the current instance o RAC total: total work area executions in optimal mode for all instances Data source: V$SYSSTAT
WA exec. one pass m.	Number of work area executions in one-pass mode o Non-RAC or RAC detail: work area executions in one-pass mode for the current instance o RAC total: total work area executions in one-pass mode for all instances Data source: V$SYSSTAT
WA exec. multipass m.	Number of work area executions below the one-pass memory requirement o Non-RAC or RAC detail: work area executions in multipass mode for the current instance o RAC total: total work area executions in multipass mode for all instances Data source: V$SYSSTAT

- *Instance Efficiency*

Field	Description
Soft parse ratio	Soft parse ratio is calculated as follows: 1 – (parse count hard / parse count total) This shows whether there are many hard parses on the system. The ratio should be compared to the raw statistics to ensure accuracy. For example a soft parse ratio of 0.2 typically indicates a high hard parse rate. However, if the total number of parses is low, you can disregard the ratio. o Non-RAC or RAC detail: ratio of the current instance

Field	Description
	o RAC total: average ratio for all instances Data source: V$SYSSTAT
In-memory sort ratio	In-memory sort ratio is calculated as follows: sorts in memory / (sorts in memory + sorts on disk) This shows the proportion of sorts that are performed in memory. Optimally, in an operational online transaction processing (OLTP) system, most sorts are small and can be performed solely as in-memory sorts. o Non-RAC or RAC detail: ratio of the current instance o RAC total: average ratio for all instances Source: V$SYSSTAT
Parse to exec. ratio	Parse to execute ratio is calculated as follows: 1 – (parse count total / execute count) In an operational environment, optimally a SQL statement should be parsed once and executed many times. Therefore an ideal value is close to 1. o Non-RAC or RAC detail: ratio of the current instance o RAC total: average ratio of all instances Source: V$SYSSTAT
Parse CPU to total	Parse-CPU-to-total ratio is calculated as follows: 1 – (parse time CPU / CPU used by this session) This shows how much of the total CPU time used was spent on activities other than parsing. When this ratio is low, the system is performing too many parses. o Non-RAC or RAC detail: ratio of the current instance o RAC total: average ratio of all instances Source: V$SYSSTAT
PTime CPU / PT elps	Parse time CPU / parse time elapsed ratio is calculated as follows: parse time CPU / parse time elapsed This can often indicate latch contention. The ratio indicates whether the time spent parsing is allocated to CPU cycles (that is, productive work) or whether the time spent parsing was not spent on CPU cycles. Time spent parsing not on CPU cycles usually indicates that the time was spent sleeping due to latch contention. o Non-RAC or RAC detail: ratio of the current instance

Field	Description
	o RAC total: average ratio of all instances Source: V$SYSSTAT

Viewing History Information

You can view history information – or “snapshot” data – when using many of the submonitors in the SAP/Oracle Database Monitor. You can specify a “since” and an “up to” date and time.

Prerequisites

You are using a sub-monitor that offers history information.

Not all sub-monitors offer history information.

Procedure

Specify *Since* and *Up to* in the screen area *Selected History* to get the required result in the sub-monitor display as follows:

Since	Up To	Result in Sub-Monitor Display
DB start	Now	Displays the changes from database start to the current time
Your selected snapshot	Now	Displays the changes from your selected snapshot to the current time.
DB start	Your selected snapshot	Displays the changes from database start to your selected snapshot.
Your selected snapshot	Your selected snapshot	Displays the changes between your selected snapshots.

Wait Event Analysis

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

Session Monitor

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.

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

Structure

This screen contains the following information:

Column	Description
SID	Session ID
ORA proc.	Oracle shadow process ID
SAP instance name	SAP instance name
Client system	Client system
Client proc.	Client process ID
Status	Session status
Event	Event name
SQL Text	Text of SQL statement
Logical reads	Logical reads
Phys. Reads	Physical reads
Block Changes	Block changes
Consistent Changes	Consistent changes
Last wait time	Last wait time (>0 wait time,0 wait,-1 wait short,-2 unknown
Seconds in Wait	Seconds in wait
P1 Text	Description of first parameter
P1	First additional parameter
P2 Text	Description of second parameter
P2	Second additional parameter
P3 Text	Description of third parameter
P3	Third additional parameter
Block Gets	Block gets

Column	Description
Consistent Gets	Consistent gets
User	Database user of the session
PGA_USED_MEM	PGA memory currently used
PGA_ALLOC_MEM	PGA memory currently allocated
PGA_FREEABLE_MEM	Allocated memory that can be freed
PGA_MAX_MEM	Maximum PGA memory ever allocated
SAP_USER (Kernel 7.1)	SAP logon user
Client_Info (Kernel 7.1)	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$SESS_IO to null. This means that you set a start point.
- *Since Reset* to display the results since the start point
- *Since DB-start* to display the results since database start from GV$SESS_IO

Integration

This monitor is based on the following views:

- GV$SQL
- GV$SQLTEXT

Buffer Busy Waits

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).

You choose *Performance* *Wait Event Analysis* *Buffer Busy Wait* in the DBA Cockpit.

You can view history information in this monitor.

Structure

Note

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

End of the note.

- *Buffer Busy Waits*

  This tab page contains information on buffer busy waits:

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

  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*

  End of the note.

- 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.

System Event

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

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 view history information in this monitor.

Structure

Note

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

End of the note.

- *Busy Waits Summary*

  This tab page displays a summary of busy waits:

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

- *Wait event details*

  This tab page displays details of wait events:

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

- *GV$SYSTEM_EVENT*

  This tab page displays details from the Oracle view GV$SYSTEM_EVENT:

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

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.

End of the note.

System Event History

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

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
Weekday	Day of the week when the snapshot was created
Date	Date of the snapshot
From	Start of snapshot interval
To	End of snapshot interval
Event	Name of the wait event
Time Waited (ms)	Total amount of time waited for the event in milliseconds
Total Waits in %	Percentage of time waited compared to total value
Average Wait (ms)	Average amount of time waited for the event in milliseconds
Total Timeouts	Total number of timeouts for the event
Total Waits	Total number of waits for the event
Time Waited (micro seconds)	Total amount of time waited for the event in microseconds

Filesystem Requests

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

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.

You cannot view history information in this monitor.

Structure

Note

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

End of the note.

- *IO per File*

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

Column	Description
File#	File number
Inst id RAC only	Instance ID
Full path	Full file name including path
Reads	Number of reads
Blk Reads	Number of block reads
Blk/Rd	Number of blocks per read
Rd Avg(ms)	Read average time in milliseconds
Rds/File(%)	Percentage of reads per file
Sgl Blk Rds	Number of single block reads
Sgl Blk Rds Avg(ms)	Average time for single block reads in milliseconds

Column	Description
Writes	Number of writes
Blk wrts	Number of block writes
Wrt Avg(ms)	Average time for writes in milliseconds
BBW	Number of buffer busy waits
Avg BBW(ms)	Average buffer busy wait time in milliseconds
BBW/File(%)	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
File#	File number
Inst id RAC only	Instance ID
Name / Device	Full file name including path
Reads	Number of reads
Blk Rds	Number of block reads
Blk/Rd	Number of blocks per read
Rd Avg(ms)	Read average time in milliseconds
Rds/File(%)	Percentage of reads per file
% of Ttl Blk Rds	Percentage of total block reads
Sgl Blk Rds	Number of single block reads
Sgl Blk Rds Avg(ms)	Average for single block reads in milliseconds
Writes	Number of writes
Blk wrts	Number of block writes

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

Enqueue Stat

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

You choose *Performance* *Wait Event Analysis* *Enqueue Stat* in the DBA Cockpit.

Note

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

End of the note.

You can view history information in this monitor.

Structure

- *v$enqueue stat*

  This tab page contains the following information:

Column	Description
Instance Id RAC only	Database instance ID
Enqueue Type	Type of enqueue requested
Total Requests	Total number of enqueue requests or enqueue conversions for this type of enqueue
% Requests	Requests for this enqueue type as a percentage of total requests
Total Waits	Total number of times an enqueue request or conversion resulted in a wait
% Waits	Waits for this enqueue type as a percentage of total waits
Total Grants	Number of times an enqueue request or conversion was granted
% Grants	Grants for this enqueue type as a percentage of total requests
Total Fails	Number of times an enqueue request or conversion failed
% Fails	Fails for this enqueue type as a percentage of total fails
Cumulative Wait Time	Cumulative (that is, total) amount of time in milliseconds spent waiting for the enqueue or enqueue conversion
% Wait Time	Wait time for this enqueue type as a percentage of total wait time
% Wait / Uptime	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.

Lock Monitor

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

You choose ▶ *Performance* ▶ *Wait Event Analysis* ▶ *Lock Monitor* ◀ in the DBA Cockpit. You can display active locks in the following ways:

- Without DBA Objects

Note

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

End of the note.

You can view history information in this monitor.

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.

  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.

End of the note.

- 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.

Latch Monitor

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.

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

You can view history information in this monitor.

Structure

Note

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

End of the note.

- *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
Name	Latch name
Inst Id RAC only	Instance ID
Wait time	Elapsed time waiting for the latch in microseconds
% Wait time	Wait time as a percentage of total wait time
Gets	Number of times the latch was requested in willing-to-wait mode and

Column	Description
	the requestor had to wait
Misses	Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
Misses/Gets	Ratio of *Misses* to *Gets*
Sleeps	Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
% Sleeps	Sleeps as a percentage of total sleeps
Immediate Gets	Number of times a latch was requested in no-wait mode
Immediate Misses	Number of times a no-wait latch request did not succeed (that is, missed)
Spin Gets	Willing-to-wait latch requests which missed the first try but succeeded while spinning
Sleep 1	Waits that slept once
Sleep 2	Waits that slept twice
Sleep 3	Waits that slept three times
Sleep 4	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
Name	Latch name
Inst ID RAC only	Instance ID
SID	ID of the session that owns the latch

- *Latch Children*

This tab page shows the number of children for the latches shown

Column	Description
Name	Latch name
Inst ID RAC only	Instance ID
Count	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
Name	Latch name
Inst ID RAC only	Instance ID
SID	ID of the session that owns the latch

- *Cache Buffers Chains*

  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.

  End of the note.

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

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

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

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

Active Session History

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.

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

Column	Description
Dictionary object number of the object	Object ID of the object that the session is referencing
Object name	Object name
File number	File number
Block number	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
SQL_ID	SQL identifier of the SQL statement that the session was executing at the time of sampling
SQL String	SQL string
SQL Child #	SQL child ID
SQL statement hash	SQL hash ID
Session serial number	Session serial number
Oracle user identifier	Oracle user ID
OPCODE	Indicates what phase of operation the SQL statement was in
Service Hash	Hash to identify service
Sess. Type	Session type
Operating system Program name	Name of the operating system program
Name of the currently executing module	Name of the currently executing module
Currently executing action	Name of the currently executing action
Client ID	Client ID of the session
Blocking Session ID	Session identifier of the blocking session Populated only when the session was waiting for enqueues or a buffer busy wait.

Column	Description
Bl. Sess. St.	Status of the blocking session
Blck Session Serial #	Serial number of the blocking session
QC Session ID	Query coordinator session ID This information is only available if the sampled session is a parallel query slave.
QC Instance ID	Query coordinator instance ID This information is only available if the sampled session is a parallel query slave.
Sample ID	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
Key fig. (y axis)	Key figure used as the value for the y axis. You can choose between the number of *Active sessions* or the *Time waited*.
Series type	Characteristic represented by a line or series in the chart You can choose between *Wait event* or *Wait event Class*.
Select Series	Filter for the wait events or wait event classes displayed in the chart
TOP n	Filter to restrict the display to the top n number of elements
Time interval	Time interval of the timestamps for summation of values in the chart If you choose the interval size *millisec (sampled info)* 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.
Refresh chart	Refreshes the display immediately to reflect any changes made to the parameters above
Save chart to frontend file	Saves the chart to a front-end file You can choose a file type and a resolution in pixels.

Feature	Description
Print graphic	Prints the chart on your front-end printer
Graphic settings on / off	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.
Load settings from frontend *Save settings to frontend*	Defines graphic settings in front-end files You can use this to transport settings between different systems.
Load settings in database *Save settings in database* *Delete settings in database*	Prompts you for an ID The ID is used to process the current graphic settings by: ▪ Loading the settings from the database ▪ Saving the settings in the database ▪ Deleting the settings from the database 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.

Workload Reporting

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

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

Note

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

End of the note.

You need to enter the following parameters for the report:

- *Begin Snapshot* and *End Snapshot* for all reports
- *Begin Snapsh. 2* and *End Snapshat 2* for compare period reports only
- *SQL ID* for SQL reports
- *AvgBlReadT*, “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.

SQL Statement Analysis

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

Shared Cursor Cache

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.

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
- *Since DB-start* to display the results since database start

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
Executions	Total number of executions
Disk Reads	Number of disk reads
Reads/Exec	Number of disk reads per execution
Buffer Gets	Number of buffer gets 999999999999 means overflow
Bgets/Exec.	Number of processed rows
Proc. Rows	Processed rows for execution
Rproc/Exec	Number of buffer gets per row
Elapsed time	Elapsed time in microseconds
Elapsed Time/Exec	Elapsed time per execution
SQL Statement	SQL statement
CPU Time	CPU time in microseconds
CPU Time/Exec	CPU time per execution
Wait Time	Wait time in microseconds
Wait Time/Exec	Wait time per execution

Column	Description
SQL Sort	Sort orders for statement
Direct Writes	Direct writes
Open curs.	Number of open cursor versions
Users	Number of users of cursor version
Versions	Number of loaded cursor versions
Share mem	Total shareable memory
Pers. mem.	Persistent memory
Runtime mem.	Total runtime memory
Number of "invalidations"	Number of invalidations
Loads	Number of loads or reloads
Parse call	Number of parse calls
Program Name	Name of calling program
User	User
Curs.Ex	Number of users executing this cursor
Inst. ID	Instance ID
First Load time	First cursor load time
Last load time	Last cursor load time
Last active time	Last active time of query plan
Application wait time	Application wait time in milliseconds
Concurrency wait time	Concurrency wait time in milliseconds
Cluster wait time	Cluster wait time in milliseconds
User I/O wait time	User I/O wait time in milliseconds

Table Access

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.

You choose ▶ *Performance* ▶ *SQL Statement Analysis* ▶ *Table Access* ◀ in the DBA Cockpit.

You cannot view history information in this monitor.

Structure

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.

End of the note.

- *Summary on Table Level*

  This screen displays the following information:

Column	Description
Instance ID RAC only	Instance ID
Owner	Table owner
Table	Table name
Size (kB)	Table size in kilobytes
Type	Type of object (v$object dependency)
Buffer pool	Default buffer pool of the object
Executions	Total number of executions of all statements accessing this table
Cache Hit Ratio	Cache hits calculated with the following formula: 100 x (1– (sum of disk reads / sum of buffer gets))
Disk Reads	Total number of disk reads executed on this table
Disk Read Ratio	Sum of (disk reads / physical reads)
Buffer Gets	Total number of buffer gets for the table
Logical Read Ratio	Sum of buffer gets / (sum of buffer gets + sum of disk reads)
Rows Proc.	Total number of rows returned or accessed by the statements on this table

Column	Description
Rows/Exec	Number of rows processed / number of executions
Bufgets/Record	Number of buffer gets / number of rows processed
Sorts	Total number of sorts done by the statements on this table
CPU Time (ms)	Total CPU time in milliseconds used by all statements on this table for parsing, executing, or fetching.
Users Opening	Total number of users executing the statement
Opening Version	Total number of statements related to this table having the child cursor locked
Loaded Version	Total number of statements related to this table having the context heap locked
# Childs	Maximum number of the child cursor
Sharable Mem.	Total amount of shared memory used by the statement in bytes
Persistent Mem.	Sum of the fixed amount of memory used for the lifetime of this statement in bytes
Runtime Mem.	Sum of the fixed amount of memory required during the execution of this statement in bytes
# Invalidations	Total number of times any cursor has been invalidated
Parse Calls	Total number of parse calls for all the statements
ITL Waits	Number of Interested Transaction List (ITL) waits for this table
Buffer Busy Waits	Number of buffer busy waits for this table
DB Block Changes	Number of database blocks changed for this table
Global Cache CR Blocks Served RAC specific	Number of global cache CR blocks served for this table
Global Cache Current Blocks Served RAC specific	Number of global cache current blocks served for this table
Logical Reads	Number of logical reads for this table
Physical Reads	Number of physical reads for this table

Column	Description
Physical Reads Direct	Number of direct physical reads for this table
Physical Writes	Number of physical writes for this table
Physical Writes Direct	Number of direct physical writes for this table
Row Lock Waits	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

Column Usage

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.

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

System Statistics for the CBO

This sub-monitor in the SAP/Oracle Database Monitor lets you collect 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. 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

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$

You can view history information in this monitor, but only for the tab page *System Statistics*.

Structure

- *System Statistics*

  This tab page displays the entire contents of SYS.AUXSTAT$:

Column	Description
SName	Statistics name
PName	Parameter name
PVal1	Parameter value 1
PVal2	Parameter value 2

The entries in the column PName have the following meanings:

PName	Meaning
STATUS	*AUTOGATHERING, COMPLETED, or BADSTATS*
DSTART	Start time for statistic collection
DSTOP	Stop time for statistic collection
FLAGS	Oracle internal flags
SPREADTM	Wait time to read a single block in milliseconds
MREADTM	Wait time to read a multiblock in milliseconds

PName	Meaning
CPUSPEED	CPU speed in millions of cycles per second
MBRC	Average multiblock read count for sequential reads, in blocks
MAXTHR	Maximum I/O system throughput, in bytes/sec
SLAVETHR	Average slave I/O throughput, in bytes/sec

- *Collected Statistics*

  This tab page displays all tables that are storing system statistics. These tables were created with the PL/SQL procedure DBMS_STATS.CREATE_STAT_TABLE:

  Note

  If the SAPR3 user does not have SELECT permissions on this table, the column ACCESS has the entry *NO* and the system statistics columns are empty.

  The columns and types in this table are not relevant as it should be accessed solely through the procedures in the DBMS_STATS package.

  End of the note.

Column	Description
Table Name	Table name
Stat_ID	Statistic ID
Version	Oracle internal
Flags	Oracle internal
C1 – C5	Oracle internal
N1 – N12	Oracle internal
D1	Oracle internal
R1 – R2	Oracle internal
CH1	Oracle internal

Statistical Information

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

SGA Monitor

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

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

You can view history information in this monitor.

Structure

Note

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

End of the note.

- *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
Comp .grp.	SGA component group
Inst Id RAC only	Instance ID
Instance Name RAC only	Instance name
Mem. size	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
Pool	Pool in which the memory of this *SGA component name* resides
SGA component name	SGA component name

Column	Description
Inst Id RAC only	Instance ID
Instance Name RAC only	Instance name
Mem. size	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
Component name	SGA component name
Inst ... RAC only	Instance ID
Instance Name RAC only	Instance name
Op. type	Type of operation, grow or shrink
Op. mode	Mode of operation, manual or automatic
Parameter for the resize op.	Parameter used in the resize operation
Parameter value at the start	Parameter value at the start of the operation
Desired param. value	Desired parameter value after the resize
Curr. value	Current value of the parameter
Start time of the operation	Start time of the operation
Start date	Start date of the operation
Upd. time	Last time progress was made for the operation
Upd. date	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
Component	SGA component name
Inst ... RAC only	Instance ID
Instance Name RAC only	Instance name
Curr. value	Current size of the component
Min. size	Minimum size of the component since instance startup
Max. size	Maximum size of the component since instance startup
Operations	Number of operations since instance startup
Last op.	Last completed operation for the component
Last mode	Mode of last completed operation for the component
Start time	Start time of the last completed operation for the component
Op. Date	Start date of the last completed operation for the component
Granul.	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
Component name	SGA component name
Inst ... RAC only	Instance ID
Instance Name RAC only	Instance name
Op. type	Type of operation, grow or shrink
Op. mode	Mode of operation, manual or automatic
Parameter for the resize op.	Parameter used in the resize operation
Parameter value at the start	Parameter value at the start of the operation
Desired param. value	Desired parameter value after the resize
Real. value after resize	Real parameter value after the resize
Status	Completion status of the operation: normal, cancel, or error
Start Time	Start time of the operation
End Time	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
Pool ID	Buffer pool identifier
Pool name	Buffer pool name
Inst ... RAC only	Instance ID
Instance Name RAC only	Instance name
Block Size	Block size in bytes for buffers in this pool
Status	Status of the advisory: ▪ ON – currently running ▪ OFF – disabled
Cache (MB)	Cache size for prediction in MB
SizeFactor	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.
CacheSize	Cache size for prediction in buffers
Phys.R. F.	Physical read factor. This is the ratio of the number of estimated physical reads to the number of reads in the real cache.
PhysReads	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
InstID... RAC only	Instance ID
Instance Name RAC only	Instance name
Size (MB)	Shared pool size for estimate
Sizefactor	Size factor with respect to the current shared pool size
lib. cache	Estimated memory in use by the library cache
Mem. Obj.	Number of library cache memory objects
Parse time	Estimated elapsed parse time saved (in seconds) because library cache memory objects are found in a shared pool of the specified size
ParseFact	Estimated parse time saved factor with respect to the current shared pool size
CacheHits	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
Inst ... RAC only	Instance ID
Instance Name RAC only	Instance name
Pool ID	Buffer pool identifier
Name	Buffer pool name

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

PGA Monitor

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

You choose ▶ *Performance* ▶ *Statistical Information* ▶ *PGA Monitor* ◀ in the DBA Cockpit.

You can view history information in this monitor.

Structure

Note

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

End of the note.

- *Status*
  - o *PGA Status*

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

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

  - o *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
Inst Id RAC only	Instance ID
Inst Name RAC only	Instance name
Workarea Address	Address of the parent cursor handle
Parent address	Address of the work area handle. This is the "primary key" for the view.
Hash Value	Hash value of the parent statement in the library cache
Chil...	Number of the child cursor that uses this work

Column	Description
– Number of this child cursor	area
Operation Type	Type of operation using the work area (`SORT`, `HASH JOIN`, `GROUP BY`, `BUFFERING`, `BITMAP MERGE`, or `BITMAP CREATE`)
Oper. ... *– Operation ID*	Unique number used to identify the operation in the execution plan
Sizi *– Sizing*	Sizing policy for this work area (`MANUAL` or `AUTO`)
EstOptSiz *– Estimated Optimal Size*	Estimated size (in KB) required by this work area to execute the operation completely in memory (optimal execution).
Est 1pSiz *– Estimated Onepass Size*	Estimated size in KB required by this work area to execute the operation in a single pass
Last Mem *– Memory Used for Last Execution*	Memory in KB used by this work area during the last execution of the cursor
Last Exec *– Last Execution*	Indicates whether this work area runs using `OPTIMAL`, `ONE PASS`, or `ONE PASS` memory requirement (or `MULTI-PASS`), during the last execution of the cursor
Last De... *– Degree of parallelism of last exec*	Degree of parallelism used during the last execution of this operation
Tot. Execs *– Total Executions*	Number of times this work area was active
Opt. Execs *– Optimal Executions*	Number of times this work area ran in optimal mode
Onepass *– Onepass Executions*	Number of times this work area ran in one-pass mode

Column	Description
Multipa... *– Multipasses Executions*	Number of times this work area ran below the one-pass memory requirement
Act. Time *– Average Active Time*	Average time this work area is active in hundredths of a second
Max Tseg... *– Maximum Temporary Segment Size*	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.
Last Tseg *– Last Temporary Segment Size*	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
SQL Text	First thousand characters of the SQL text for the current cursor
Optimal ... *– Optimal Executions*	Number of times this work area ran in optimal mode
Optimal Pe *– Optimal Percentage*	*Optimal Executions* as a percentage of *Total Executions*
Onepass ... *– Onepass Executions*	Number of times this work area ran in one-pass mode
Onepass Pe	*Onepass Executions* as a percentage of *Total Executions*

Column	Description
– Onepass Percent	
Multipass ... *– Multipasses Executions*	Number of times this work area ran below the one-pass memory requirement
Multipass ... *– Multipasses Percent*	*Multipasses Executions* as a percentage of*Total Executions*
Total Exec *– Total Executions*	Number of times this work area was active

- o *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
Inst Id RAC only	Instance ID
Inst Name RAC only	Instance name
Low bound	Lower bound for the optimal memory requirement of work areas included in this row (bytes)
High Bound	Higher bound for the optimal memory requirement of work areas included in this row (bytes)
Opt. Execs *– Optimal Executions*	Number of times this work area ran in optimal mode
Onepass *– Onepass Executions*	Number of times this work area ran in one-pass mode
Multipa – *Multipass Executions*	Number of times this work area ran below the one-pass memory requirement

Column	Description
Tot. Execs *– Total Executions*	Number of times this work area was active

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

Column	Description
Executions	

- *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
Inst Id RAC only	Instance ID
Inst Name RAC only	Instance name
Name	Statistic name
Value	Statistic value
%	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
Inst Id RAC only	Instance ID
Inst Name RAC only	Instance name
SID	Session identifier
Oper. Type	Type of operation using the work area (sort, hash join, group by, buffering, bitmap merge, or bitmap create)
Exp. Size *– Expected workarea size*	Expected size in KB for the work area
Act. Used *– PGA Memory Currently*	Amount of PGA memory in KB currently allocated for this work area

Column	Description
Max Mem *– Maximum memory used*	Maximum amount of memory used by this work area
Passes *– Number of Passes*	Number of passes for this work area
TmpSeg ... *– Temporary Segment Size*	Size in bytes of the temporary segment used for this work area.
SQL Text	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
OS Program Name	Operating system program name
Inst Id RAC only	Instance ID
Inst Name RAC only	Instance name
PGA Memory	PGA memory currently used by the process
PGA Memory	PGA memory currently allocated by the process
Max PGA Memory	Maximum PGA memory allocated by the process
Process St.	Process status
SQL Text	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
Inst Id RAC only	Instance ID
Inst Name RAC only	Instance name
TARGET (MB)	Operating system program name
Val *– Estimated value of the cache hit percent*	Value of PGA_AGGREGATE_TARGET for this prediction (in bytes)
Overall.Cn. *– Overalloc. count*	Estimated number of PGA memory over-allocations if the value of PGA_AGGREGATE_TARGET is set to PGA_TARGET_FOR_ESTIMATE.

- *Advice Histogram*

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

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

Undo Statistics

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

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

You cannot view history information in this monitor.

Structure

Note

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

End of the note.

- *Daily Summaries*

  This tab page displays daily summaries of undo statistics:

Column	Description
Instance Id RAC only	Database instance ID
begin date	Begin date for the analysis
begin time	Begin time for the analysis
end date	End date for the analysis
end time	End time for the analysis
last active undo tablespace	Last active undo tablespace. If more than one undo tablespace was active in the analysis

Column	Description
	period, the active undo tablespace that was active at the *end* of the period is reported.
total number of undo blocks	Total number of undo blocks 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.
total number of transactions	Total number of transactions
length of the longest query (sec)	Length of the longest query in seconds You can use this statistic to estimate the proper setting of the `UNDO_RETENTION` initialization parameter.
highest Nbr. of TAs executed Concurrently	Highest number of transactions executed concurrently
attempts to obtain undo space	Number of attempts to obtain undo space by stealing unexpired extents from other transactions
unexpired undo blocks removed	Number of unexpired undo blocks removed so they can be used by other transactions
unexpired undo blocks reused	Number of unexpired undo blocks reused by other transactions
attempts to steal expired undo blocks	Number of attempts to steal expired undo blocks from other undo segments
expired undo blocks stolen from segments	Number of expired undo blocks stolen from other undo segments
expired undo blocks reused	Number of expired undo blocks reused within the same undo segments
number of occurrences of ORA-01555	Number of occurrences of ORA-01555 You can use this statistic to decide whether or not the `UNDO_RETENTION` 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 `UNDO_RETENTION`.
nbr of times space was requested	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.

Column	Description
number of transactions per second	Number of transactions per second

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.

End of the note.

- *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.

  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.

  End of the note.

- *Max Space Consumption*

  This tab page displays maximum space consumption for undo tablespaces:

Column	Description
Instance Id RAC only	Database instance ID
undo tablespace name	Name of the undo tablespace for which the maximum space consumption has been computed
total undo tablespace size in MB	Total size of the undo tablespace in MB
max. used undospace in MB	Maximum used undo space in MB
max. used in %	Used undo space as a percentage of the maximum

Performance Database

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

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
Weekday	Day of the week for the snapshot
Date	Date of the snapshot
Time	Time of the snapshot
User Calls	Number of user calls such as login, parse, fetch, or execute
Recursive Calls	Internal SQL statement or SQL statement in PL/SQL statement
User / Recursive Calls	Ratio of user to recursive Calls
User Commits	Number of commits and rollbacks
Parses	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.
Reads / User Call	Amount of logical reads per user call. Should be less than 20.
Logical Reads	Sum of "db block gets" and "consistent gets"
Physical Reads	Number of physical reads
Physical Reads Direct	Number of reads directly from disk, bypassing the buffer cache
Physical Reads	Number of LOB reads directly from disk, bypassing the buffer cache

Column	Description
Direct (LOB)	
Buffer Quality	Percentage of how many db blocks are found in the db cache and haven’t to read from disk.
Physical Writes	Number of physical reads
Table Fetch by RowID	Number of rows accessed by RowID, including all rows accessed using indexes
Table Fetch Continued Row	Number of times when second row piece of chained rows is fetched. A high number indicates that rows are chained.
Table Scans Rows Gotten	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.
Table Scans Blocks Gotten	Number of blocks accessed by all full table scans
Redo Blocks Written	Total number of redo blocks written
Redo Write Time (ms)	Total redo write time since database start in milliseconds
Avg. Redo Write time (msec)	Average time the LGWR needs to write the redo log information from buffer to disk in milliseconds
Buffer Busy Waits	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.
Buffer Busy Waits Time (sec)	Total buffer busy wait time since database start in seconds
Avg. Buffer Busy Waits Time (msec)	Average time a session has to wait for the event buffer busy waits in milliseconds
Full Table Scans	Sum of full table scans for long and short tables: • Short table scans are against tables with 4 or less database blocks. • Long table scans are against tables with 5 or more database blocks.

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.

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.

End of the note.

- 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.

DB Overview Script

This sub-monitor in the SAP/Oracle Database Monitor gives you the result of an underlying SQL script.

You choose ▶ *Performance* ▸ *Statistical Information* ▸ *DB Overview Script* ◀ and *Online* to generate the list online or *Spool* to generate a background job. To use the spool option you need to have the required authorization for job generation.

You can choose a remote system prior to executing the sub-monitor, which means that the remote system executes the underlying SQL script.

Recommendation

Due to the long run time, we recommend you to run this in the background.

You cannot view history information in this monitor.

You cannot view different instances in this monitor.

You cannot execute this sub-monitor on a preselected database instance

System Metrics

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.

You choose ▶ *Performance* ▸ *Statistical Information* ▸ *System Metrics* ◀ in the DBA Cockpit.

Note

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

End of the note.

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
Snapshot ID	Unique snapshot ID
Database ID	Database ID for snapshot
Instance number	Instance number for snapshot
Begin Time	Begin of snapshot time interval
End Time	End of snapshot time interval
Int Size	Time interval in hundredths of a second
Group ID	Metric group ID
Metric ID	Metric ID
Metric Name	Metric name
Value	Metric value
Metric Unit	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.

System Summary Metrics

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.

You choose ▶ *Performance* ▶*Statistical Information* ▶*System Summary Metrics* ◀in the DBA Cockpit.

Note

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

End of the note.

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

Use this to display data from the *GV$-VIEW* or the *DBA View*.

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.

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
Snapshot ID	Unique snapshot ID
DB ID	Database ID for snapshot
Inst_ID_CH	Instance number for snapshot
Snapshot begin time	Begin of snapshot time interval
Snapshot end time	End of snapshot time interval
Int size	Time interval in hundredths of a second
Group ID	Metric group ID
Metric ID	Metric ID

Column	Description
Metric Name	Metric name
Metric Unit	Metric unit description
Num Interv	Number of intervals observed
Minval	Minimum value observed
Maxval	Maximum value observed
Average	Average over the period
StdDeviat	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.

File Metrics

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.

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.

End of the note.

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
Snapshot ID	Unique snapshot ID
DB ID	Database ID for snapshot
Inst_ID_CH	Instance number for snapshot
Fst file#	File number
File name	File name
Tablespace	Tablespace name
CreatTime	Timestamp of file creation
Snapshot begin time	Begin of snapshot time interval
Snapshot end time	End of snapshot time interval
Int size	Time interval in hundredths of a second
Group ID	Metric group ID
AvgRdTime	Average file read time
AvgWrtTime	Average file write time

Column	Description
Phys rds	Number of physical reads
Phys wrt	Number of physical writes
Blk rd	Number of physical block reads
Blk wr	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.

Wait Class Metrics

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.

You choose *Performance* ▶ *Statistical Information* ▶ *Wait Class Metrics* in the DBA Cockpit.

Note

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

End of the note.

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
Snapshot ID	Unique snapshot ID
DB ID	Database ID for snapshot
Inst_ID_CH	Instance number for snapshot
W Class ID	Identifier for the class of the wait event
W Class NA	Name for the class of the wait event
Snapshot begin time	Begin of snapshot time interval
Snapshot end time	End of snapshot time interval
Intsize	Time interval in hundredths of a second
Group ID	Metric group ID

Column	Description
AvgWtCnt	Average wait count
DBTimeWait	Database wait time
Time waited	Elapsed wait time waited
Wait Cnt	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.

Feature Monitoring

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

Automatic Segment Space Management

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.

You choose ▶ *Performance* ▶ *Feature Monitoring* ▶ *Automatic Segment Space Management* ◀ in the DBA Cockpit and then the required tab page.

You *cannot* view history information in this monitor.

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

Note

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

End of the note.

- *Tablespaces with ASSM*

  This tab page displays information on tablespaces with ASSM:

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

- *All Tablespaces*

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

  - With ASSM: *Segment Space Mngt* is *AUTO*
  - Without ASSM: *Segment Space Mngt* is *MANUAL*

- *Tables with ASSM*

  This tab page displays information on tablespaces with ASSM:

Column	Description
Table Name	Name of table
Tablespace Name	Name of tablespace
Used Space	Used space in the table in bytes

Column	Description
(Bytes)	
Unused Space (Bytes)	Unused space in the table in bytes
Meta Data Blocks	Total blocks reported by DBA_TABLES minus sum of values reported by PL/SQL routine SPACE_USAGE

Note

Choose *Select Table* to display information on a selected single table or a group of tables.

End of the note.

- *Tables without ASSM*

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

Online Redefinition

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.

You choose *Performance* ▸ *Feature Monitoring* ▸ *Online Redefinition* in the DBA Cockpit and then the required tab page.

You cannot view history information in this monitor.

Structure

Note

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

End of the note.

- *Tables in redefinition mode*

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

Column	Description

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

Resumable Space Allocation

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.

You choose ▶ *Performance* ▶ *Feature Monitoring* ▶ *Resumable Space Allocation* ↵.

You can view history information in this sub-monitor.

Structure

Note

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

End of the note.

This screen displays the following information:

Column	Description
User ID	User ID of the resumable statement owner

Column	Description
Username	User name of the resumable statement owner
Session ID	Session identifier
Inst ID	Instance ID of resumable statement
Coord Inst_ID	Inst ID on which the Parallel Coordinator is running
Coord Sess ID	Session ID of the Parallel Coordinator
Status	Statement status. Possible values: *RUNNING*, *SUSPENDED*, *ABORTED*, *ABORTING*, *TIMEOUT*
Timeout	Timeout of the resumable statement
Start Time	Local start time of the resumable statement
Suspend Time	Local last time when the resumable statement was suspended
Resume Time	Local last time when the resumable statement was resumed
Name	The name given in the resumable clause of this resumable statement.
SQL Text	SQL text of the resumable statement
Error Number	The error code of the last correctable error
Error Parameter 1	Parameter for error message 1
Error Parameter 2	Parameter for error message 2
Error Parameter 3	Parameter for error message 3
Error Parameter 4	Parameter for error message 4
Error Parameter 5	Parameter for error message 5
Error Message	The error message corresponding to *Error Number*.

Parallel Query

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.

You choose ▶ *Performance* ▶*Feature Monitoring* ▶*Parallel Query* ◀in the DBA Cockpit.

You cannot view history information in this monitor.

Structure

This screen displays the following information:

Column	Description
Parallel Coor.	Parallel coordinator number
SID	System ID number
Username	User name
Inst ID	Instance ID
Server Group	Server group
Server Set	Server set
log. Nb.DB Proc.	Logical number of DB process
Inst of Coord.	Instance of coordinator
Degree	Degree of parallelism
Req. Degree	Required degree of parallelism

Data Guard

This sub-monitor in the SAP/Oracle Database Monitor lets you monitor the Oracle data guard functionality.

You choose ▶ *Performance* ▶*Feature Monitoring* ▶*Data Guard* ◀in the DBA Cockpit.

You cannot view history information in this monitor.

Structure

- *Overview*

  This tab page gives you an overview of the data guard sub-monitor:

- *STATUS*:

  Shows the current status of the data guard functionality.

  Possible values: *ALL*, *STANDBY*, or *NONE*.

- *ERROR*:

  Shows the worst error registered during the last hours.

  Possible values: *INFORMATIONAL* (green), *CONTROL* (green), *WARNING* (Yellow), *ERROR* (red), or *FATAL* (red).

- *Database*

  This tab page displays the full contents of the Oracle view V$DATABASE. For database guard analysis the field *GUARD_STATUS* is most relevant here. It displays the current status of the data guard functionality.

Column	Description
Instance ID	Instance ID
Database ID	Database ID calculated when the database is created and stored in all file headers
Name	Name of the database
Created (Date)	Creation date
Created (Time)	Creation time
Resetlogs change#	Change number at open resetlogs
Resetlogs (Date)	Date of open resetlogs
Resetlogs (Time)	Time of open resetlogs
Prior resetlogs change#	Change number at prior open resetlogs
Prior resetlogs (Date)	Date of prior open resetlogs
Prior resetlogs (Time)	Time of prior open resetlogs
Log mode	Archive log mode (*NOARCHIVELOG* or *ARCHIVELOG*)
Checkpoint change	Last SCN checkpointed
Archive	Last SCN archived

Column	Description
change #	
Controlfile type	Type of control file: o *STANDBY*– Indicates that the database is in standby mode o *CLONE*– indicates a clone database o *BACKUP* | *CREATED*– indicates the database is being recovered using a backup or created control file o *CURRENT*– the control file changes to this type following a standby database activate or database open after recovery
Controlfile created (date)	Date that control file was created
Controlfile created (time)	Time that control file was created
Controlfile sequence#	Control file sequence number incremented by control file transactions
Controlfile change#	Last change number in backup control file (null if the control file is not a backup)
Controlfile (date)	Last date in backup control file (null if the control file is not a backup)
Controlfile (time)	Last time in backup control file (null if the control file is not a backup)
Open Resetlogs	Indicates whether the next database open allows or requires the resetlogs option: *NOT ALLOWED*, *ALLOWED*, or *REQUIRED*
Version (Date)	Version date
Version (Time)	Version time
Open mode	Open mode information
Protection mode	Protection mode currently in effect for the database: o *MAXIMUMPROTECTION* Database is running in maximized protection mode o *MAXIMUM AVAILABILITY*– Database is running in maximized availability mode o *RESYNCHRONIZATION*– Database is running in resynchronization mode

Column	Description
	o *MAXIMUM PERFORMANCE*– database is running in maximized protection mode o *UNPROTECTED*– Database is unprotected (this normally occurs when the primary database is mounted and not open)
Remote archive	Value of the `REMOTE_ARCHIVE_ENABLE` initialization parameter
Activation	Number assigned to the database instantiation
Database role	Current role of the database, either primary or standby
Archivelog change	Highest `NEXT_CHANGE#` (from the V$ARCHIVED_LOG view) for an archived log
Switchover status	Indicates whether switchover is allowed: o *NOT ALLOWED*– either this is a standby database and the primary database has not been switched first or this is a primary database and there are no standby databases. o *SESSIONS ACTIVE*– there are active SQL sessions attached to the primary or standby database that need to be disconnected before the switchover operation is permitted. Query the V$SESSION view to identify the specific processes that need to be terminated. o *SWITCHOVER PENDING*– this is a standby database and the primary database switchover request has been received but not processed. o *SWITCHOVER LATENT*– the switchover was in pending mode, but did not complete and went back to the primary database. o *TO PRIMARY*– this is a standby database and is allowed to switch over to a primary database. o *TO STANDBY* – this is a primary database and is allowed to switch over to a standby database. o *RECOVERY NEEDED*– this is a standby database that has not received the switchover request
Guard status	Protects data from being changed: o *ALL*– all users other than SYS are prevented from making changes to any data in the database. o *STANDBY*– all users other than SYS are prevented from making changes to any database object being maintained by logical standby. *NONE*– normal security for all data in the database

Column	Description
Supplemental log data min	Makes sure that LogMiner has sufficient information to support chained rows and various storage arrangements such as cluster tables.
Supplemental log data pk	For all tables with a primary key, makes sure that all columns of the primary key are placed into the redo log whenever an update operation is performed.
Supplemental log data ui	For all tables with a unique key, makes sure that if any unique key columns are modified, all other columns belonging to the unique key are also placed into the redo log.
Force logging	Whether the database is under force logging mode (*YES*) or not (*NO*)

- *Dataguard Status*

  This tab page displays the following information based on the view V$DATAGUARD_STATUS. This view displays and logs events that would typically be triggered by any message to the alert log or server process trace files.

Column	Description
Instance ID	Instance ID
Facility	Facility that encountered the event. Possible values are: o *CRASH RECOVERY* o *LTS* o *LAS* o *RMS* o *REMOTE FILE SERVER* o *FETCH ARCHIVE LOG* o *DATA GUARD* o *NETWORK SERVICES*
Severity	Severity of the event. Possible values are: o *INFORMATIONAL* – informational message o *WARNING* – warning message o *ERROR* – indicates the process has failed o *FATAL* o *CONTROL* – an expected change in state such as the start or completion of an archival, log recovery, or switchover operation
Destination	Destination ID number of the event. If the event does not have a

Column	Description
ID	particular destination, the value is 0.
Message number	A chronologically increasing number giving each event a unique number
Error code	Error ID of the event
Callout	Indicates whether the current entry is a callout event (*YES*) or not (*NO*) o *YES*means that this event may require the DBA to perform some action. Examine the *ERROR_CODE* and *MESSAGE* columns for more information. o *NO* generally corresponds to an *INFORMATIONAL* or *WARNING* event, which does not require any action.
Date	Date of the event
Time	Time of the event
Message	Text message describing the event

- *Managed Standby*

  This tab page displays the following information based on the view V$MANAGED_STANDBY. This view displays current status information for Oracle database server processes on physical standby databases in the Data Guard environment.

Column	Description
Instance ID	Instance ID
Process	Type of process for which information is being reported: o ARCH – archiver process o RFS – remote file server o MRP0 – detached recovery server process o MR(fg) – foreground recovery session
Process ID	Operating system process identifier of process
Status	Current process status. Possible values are: o *UNUSED*– no active process o *ALLOCATED*– process is active but not currently connected to a primary database client o *CONNECTED*– network connection established to a primary database client o *ATTACHED*– process is actively attached and

Column	Description
	communicating to a primary database client o *IDLE*– process is not performing any activities o *ERROR*– process has failed o *OPENING*– process is opening the archived redo log o *CLOSING*– process has completed archival and is closing the archived redo log o *WRITING*– process is actively writing archived redo log data o *RECEIVING*– process is receiving network communication o *ANNOUNCING*– process is announcing the existence of a potential dependent archived redo log o *REGISTERING*– process is registering the existence of a completed dependent archived redo log o *WAIT_FOR_LOG*– process is waiting for the archived redo log to be completed o *WAIT_FOR_GAP*– process is waiting for the archive gap to be resolved o *APPLYING_LOG*– process is actively applying the archived redo log to the standby database
Client Process	Identifies the corresponding primary database process. Possible values are: o *ARCHIVAL*– foreground (manual) archival process (SQL) o *ARCH*– background ARCn process o *LGWR*– background LGWR process
Client Pid	Operating system process identifier of the client process
Client DBid	Database identifier of the primary database
Group#	Standby redo log group
Thread#	Archived redo log thread number
Sequence#	Archived redo log sequence number
Block#	Last processed archived redo log block number
Blocks	Size of the archived redo log in 512-byte blocks
Delay (min)	Archived redo log delay interval in minutes

Column	Description
Know agents	Total number of standby database agents processing an archived redo log
Active agents	Number of standby database agents actively processing an archived redo log

Additional Functions

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

SQL Command Editor

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.

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

You cannot view history information in this monitor.

Structure

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.

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 +, -, *, /, ||. Also unary functions (`LN, MIN, AVG ...`), null, or numbers, are allowed.

- A table reference must have the following syntax:

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

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.

  End of the caution.

  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 …
```

As 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
```

End of the example.

- Every column alias that is specified in the select list of a sub-query can be used like 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.

  End of the caution.

  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
  ```

  End of the example.

- 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())
  ```

Display GV$-Views

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`.

You choose ▶ *Performance* ▶ *Additional Functions* ▶ *Display GV$ Views* ◀ in the DBA Cockpit.

You can view history information in this monitor.

Structure

The GV$ views are arranged in a two-column list in alphabetic order of the views' names. This sub-monitor provides a history for each GV$ view with a corresponding table in SAP dictionary. These views are displayed in *lowercase* in the table.

You can display the history, where relevant, by entering the *Since* and *Up To* timestamps. This makes sense for parameters such as memory space, statistics, or other parameters with a periodic aspect. For parameters where it makes no sense to accumulate data, you only need choose a value for *Up To* to see the value of the parameter at the specified point in time.

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 (`logmnr_*`) cannot be queried unless the service is up and running.

Display DBA Tables

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.

You choose *Performance* ▸ *Additional Functions* ▸ *Display DBA Tables* in the DBA Cockpit.

Structure

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

Database Parameters

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.

- 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. You cannot view history information in this monitor.

Structure

- *Active Parameters*

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

Column	Description
Instance ID	Instance ID
SID	Name of the RAC instance
Parameter	Name of the active parameter
Parameter value	Value of the parameter
Value in SPFILE	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
Instance ID	Instance ID
SID	Name of the RAC instance

Column	Description
Parameter	Name of the parameter
Value	Value of the parameter
Timestamp	Timestamp for this value of the parameter
Scope	Indicates whether the parameter change is only temporary, or persistent and in memory
Target instance	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
SID	Name of the instance
Parameter	Name of the parameter
Value	Value of the parameter
ISpecified	Whether the parameter was specified in the server parameter file
Ordinal	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.

Alert Log

This sub-monitor in the SAP/Oracle Database Monitor lets you check the database message log.

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.

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.

End of the note.

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 `alert_log`	BRTOOLS must be installed on the database instances. All administrative files, such as `servernames`, must be maintained correctly.
BRTOOLS using file type `alert_log` with specified `sysid`	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 `file`	BRTOOLS have to be installed on the database instances.

Access Method to Message Log	Requirement
	This method is more tolerant of missing administrative file entries than the previous methods.
RFC call and READ dataset	The instance server must also host a SAP application server. The data file must be accessible by the application server.

- 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 Notes* 80689 and 446172.

  Make sure that the trace file has not been deleted if you want to view it.

Checkpoints

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

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.

You cannot view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Checkpoint Number	Number of the checkpoint
Start Date	Time the checkpoint ended

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

Note

Parallel checkpoints are indicated by:

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

End of the note.

Oracle Net

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

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 (local)*
- *sqlnet.ora (local)*
- *protocol.ora (local)*
- *listener.log*
- *listener.ora*
- *sqlnet.ora*

User End to End Trace

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.

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.

End of the caution.

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>`.

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.

   Note

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

   End of the note.
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`).

User End to End Stats Trace

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

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>`.

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.

   Note

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

   End of the note.
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.

Program End to End Trace

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.

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.

End of the caution.

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>`.

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.

   Note

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

   End of the note.
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`).

Program End to End Stats Trace

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

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>`.

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.

   Note

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

   End of the note.
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.

RAC Statistics

These sub-monitors in the SAP/Oracle Database Monitor show statistics for Oracle Real Application Cluster (RAC).

Real Cache Get Hit Ratio

This sub-monitor in the SAP/Oracle Database Monitor displays the real cache get hit ratio, which is the proportion of buffer accesses that found a buffer in the local cache. It lets you monitor the caching efficiency of the local instance and monitor progress when optimizing SQL for buffer accesses.

You choose ▶ *Performance* ▶ *RAC Statistics* ▶ *Real Cache Get Hit Ratio* ◀ in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
Real Cache Hit Ratio [%]	Real cache hit ratio as percentage
GC Curr Blocks Receive	Number of global cache current blocks received
GC Curr Blocks Receiv [%]	Global cache current blocks received as percentage
GC CR Blocks Receive	Number of global cache consistent read blocks received
GC CR Blocks Receive [%]	Global cache consistent read blocks as percentage
Physical R	Number of physical reads
Physical Reads [%]	Physical reads as percentage
Physical Reads Direct	Number of direct physical reads
Physical Reads Direct [%]	Direct physical reads as percentage
Physical Reads Direct (lob)	Number of direct physical reads (lob)
Physical Reads Direct (lob) [%]	Direct physical reads (lob) as percentage
Logical Reads	Number of logical reads
Logical Reads [%]	Logical reads as percentage

Global Cache Fusion Block Hit Ratio

This sub-monitor in the SAP/Oracle Database Monitor displays the global cache fusion block hit ratio, which is the proportion of global cache operations resulting in buffer copies from other instances. It lets you drill down into the global cache work distribution and identify the efficiency of the global cache.

You choose ▶ *Performance* ▶ *RAC Statistics* ▶ *Global Cache Fusion Block Hit Ratio* ◀ in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Instance ID	Instance ID
GC Fusion Block Hit Ratio [%]	Global cache fusion block hit ratio as percentage
GC Curr Blocks Receive	Number of global cache current blocks received
GC Curr Blocks Receive [%]	Global cache current blocks received as percentage
GC CR Blocks Receive	Number of global cache consistent read blocks received
GC CR Blocks Receive [%]	Number of global cache consistent read blocks received as percentage
GC Gets	Number of global cache gets
GC Gets [%]	Global cache gets as percentage
GC Converts	Number of global cache converts
GC Converts [%]	Global cache converts as percentage

Block Shipping for Current Blocks

This sub-monitor in the SAP/Oracle Database Monitor displays the statistics on block shipping for current blocks.

You choose *Performance* *RAC Statistics* *Block Shipping for Current Blocks* in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst id	Instance ID
Owner	Owner name

Column	Description
Object	Object
Type	Type
Current Blocks Served	Number of current blocks served
Current Blocks Served per Inst [%]	Current blocks served per instance as percentage
Current Blocks Served [%]	Current blocks served as percentage
Current Blocks Served in % of LIO	Current blocks served as percentage of logical IO

Block Shipping for CR Blocks

This sub-monitor in the SAP/Oracle Database Monitor displays the statistics on block shipping for consistent read (CR) blocks.

You choose ▶ *Performance* ▶ *RAC Statistics* ▶ *Block Shipping for CR Blocks* ◀ in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst id	Instance ID
Owner	Owner name
Object	Object
Type	Type
CR Blocks Served	Number of consistent read blocks served
CR Blocks Served per Inst [%]	Consistent read blocks served per instance as percentage
CR Blocks Served [%]	Consistent read blocks served as percentage
CR Blocks Served in % of LIO	Consistent read blocks served as percentage of logical IO

Fusion to Physical Writes Ratio

This sub-monitor in the SAP/Oracle Database Monitor displays the ratio of fusion to physical writes ratio, which is the proportion of dirty buffers that was globally dirty when written to disk. This indicates how much RAC-related work the database writer (DBWR) performs.

You choose *Performance* ▸ *RAC Statistics* ▸ *Fusion to Physical Writes Ratio* in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
Fusion to Physical Writes Ratio [%]	Ratio of fusion to physical writes as percentage
Physical Writes	Number of physical writes
Physical Writes [%]	Physical writes as percentage
DBWR Fusion Writes	Number of DBWR fusion writes
DBWR Fusion Writes [%]	DBWR fusion writes as percentage

Global Cache CR Blocks Received

This sub-monitor in the SAP/Oracle Database Monitor displays the global cache consistent read blocks received. This counter is incremented in a foreground or background process when a search for a consistent read (CR) version of a block fails to find a buffer with the required snapshot of the data. It also means that the current version for that particular block is not cached locally. Therefore, an attempt is made to get the block from a remote instance. The requestor can receive either a CR or a current version of the data block.

These statistics represent the accumulated end-to-end elapsed time or latency for a request. The unit is hundredths of a second. Each individual request is timed from the point when the request is made until it completes. The start time of the request is stored in a global cache structure representing the cluster-wide state of a cached data block.

Normally, CR requests are not blocked and their latencies are almost entirely bounded by processing and messaging time. Therefore, most of the recommendations for current blocks apply in this case as well. A certain amount of wait time is incurred when redo needs to be flushed to disk before shipping a CR block as a result of changes made to a block while creating a consistent copy, but this event is normally rare. If a significant proportion (> 5%) of CR blocks require redo flushing before being shipped, the CR service time might be affected generally by log files sync waits and log write time, for which the rules of log-buffer and log-file tuning apply.

You choose *Performance* ▶ *RAC Statistics* ▶ *Global Cache CR Blocks Received* in the DBA Cockpit.

You can view history information in this monitor.

Structure

Column	Description
Inst Id	Instance ID
GCS CR Blocks Received	Number of blocks received for global cache service (GCS) CR
GCS CR Blocks Received [%]	Number of GCS CR blocks
GCS CR Block Receive Time [sec]	Receive time for GCS CR blocks in seconds
GCS CR Block Receive Time [%]	Receive time for GCS CR blocks as percentage
Avg CR Block Receive Time [ms]	Average receive time for CR blocks in milliseconds

Global Cache Current Blocks Receive

This sub-monitor in the SAP/Oracle Database Monitor displays the receive time for global cache current blocks.

This counter is incremented by a foreground or background process when a request for a current block is made for modification. When the current block is received, the completion handler increments the statistic. The statistic is related to database blocks gets, which is a logical read for a current block. If a current read fails to find a current buffer for the block in the local cache, it attempts to obtain it from a remote instance. The ratio of global cache current blocks received to database block gets represents the remote cache hit ratio for current blocks.

In a system with a lot of concurrent writes, this ratio can be high. In this case we advise you to monitor object statistics to identify the database objects with remote concurrent writes and the SQL statements causing this. The statistics on the receive time for global cache current blocks represent the cumulated end-to-end elapsed time or latency for a current block request in hundredths of second. Each individual request is timed from the point when the request is made until it completes. The start time of the request is stored in a global cache structure representing the cluster-wide state of a cached data block.

You choose *Performance* ▶ *RAC Statistics* ▶ *Global Cache Current Blocks Receive* in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
GCS Curr Blocks Receive	Number of global cache service (GCS) current blocks received
GCS Curr Blocks Receive [%]	Number of GCS current blocks as percentage
GCS Current Block Receive Time [sec]	Receive time for GCS current blocks in seconds
GCS Current Block Receive Time [%]	Receive time for GCS current blocks as percentage
Avg Current Block Receive Time [ms]	Average receive time for current blocks in milliseconds

Global Cache CR Block Send

This sub-monitor in the SAP/Oracle Database Monitor displays the sending information on global cache consistent read (CR) blocks.

The statistics shown here represent a count of the CR block requests served by this instance and maintained by the global cache server processes (for example, `LMS0`, `LMS1`). The serving of a CR block is usually a very fast operation and is not blocked. It includes the flushing of pending redo to disk.

The send time is measured when a block is handed to the IPC layer to be queued for sending after it has been processed. On some platforms and with certain protocol implementations, send completions must be returned, because sends are implemented as synchronous operations. Therefore, the send time is measured from when a block is handed off to the IPC layer until the protocol returns a completion. A completion is a status indicating that it has taken control of the send operation. The average latency is usually very small, less than a millisecond.

You choose *Performance* *RAC Statistics* *Global Cache CR Block Send* in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
GCS CR Blocks Send	Number of global cache service (GCS) CR blocks sent
GCS CR Blocks Send [%]	Number of GCS CR blocks as percentage
GCS CR Block Send Time [sec]	Send time for GCS CR blocks in seconds
GCS CR Block Send Time [%]	Send time for GCS CR blocks as percentage
Avg CR Block Send Time [ms]	Average send time for CR blocks in milliseconds

Global Cache GES Convert

This sub-monitor in the SAP/Oracle Database Monitor displays information on global cache for global enqueue service (GES) convert.

The global cache convert counter is incremented on completion of a request for a change in access privileges to a data block, that is, from shared to exclusive mode. In a large cluster, this operation can result in multiple messages, as blocking asynchronous system trap (BAST) notifications need to be sent to multiple cluster nodes that had been accessing the block in shared mode before the up-convert started.

Most global cache converts occur on frequently read data structures such as segment header blocks or bitmap blocks, but converts are also characteristic for index branch or leaf blocks. Normally, these converts are from S to X. Global cache convert time is the time it takes from the start of a convert – usually from shared to exclusive mode – until the requestor receives the grant. High average convert times are mostly due to high system load.

You choose ▶ *Performance* ▶ *RAC Statistics* ▶ *Global Cache GES Convert* ◀ in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
GCS Cache Converts	Number of global cache service (GCS) cache converts
GCS Cache Converts [%]	GCS cache converts as percentage
GCS Cache Convert Timeouts	Number of timeouts during GCS cache convert

Column	Description
GCS Cache Convert Time [sec]	Time of GCS cache convert in seconds
GCS Cache Convert Time [%]	Time of GCS cache convert as percentage
Avg Cache Convert Time (ms)	Average cache convert time in milliseconds

Global Cache Get Hit Ratio

This sub-monitor in the SAP/Oracle Database Monitor displays information on the ratio of global cache get hits, which is the proportion of global cache operations resulting in the setting of access permissions for an un-cached block. It lets you drill down into the global cache work distribution and identify the efficiency of the global cache.

You choose ▶ *Performance* ▶ *RAC Statistics* ▶ *Global Cache Get Hit Ratio* ↵ in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
GC Get Hit Ratio [%]	Global cache (GC) get hit ratio as percentage
GC Curr Blocks Receive	Number of GC current blocks received
GC Curr Blocks Receive [%]	GC current blocks received as percentage
GC CR Blocks Receive	Number of GC consistent read (CR) blocks received
GC CR Blocks Receive [%]	GC CR blocks received as percentage
GC Gets	Number of GC gets
GC Gets [%]	GC gets as percentage
GC Converts	Number of GC converts
GC Converts [%]	GC converts as percentage

Global Cache GCS Gets

This sub-monitor in the SAP/Oracle Database Monitor displays information on global cache for global cache service (GCS) get hits.

You choose ▶ *Performance* ▶ *RAC Statistics* ▶ *Global Cache GCS Gets* ◀ in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
Avg Get Acquired Time [ms]	Average time for get acquired in milliseconds
GCS Gets Acquired	Number of GCS gets acquired
GCS Gets Acquired [%]	GCS gets acquired as percentage
GCS Gets Acquired Time [%]	Time for GCS gets acquired as percentage
GCS Gets Acquired Time [s]	Time for GCS gets acquired in seconds

Global Cache Hit Ratio

This sub-monitor in the SAP/Oracle Database Monitor displays information on the global cache hit ratio, which is the proportion of logical reads resulting in a cache fusion transfer or cache coherency operation.

This ratio is an indicator of how much work is done at the buffer cache level to maintain the global cache

You choose ▶ *Performance* ▶ *RAC Statistics* ▶ *Global Cache Hit Ratio* ◀ in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description

Column	Description
Inst Id	Instance ID
GC Hit Ratio	Global cache hit ratio
GC Current Blocks Receive	Number of global cache current blocks received
GC Current Blocks Receive [%]	Number of global cache current blocks received as percentage
GC CR Blocks Receive	Number of global cache constant read (CR) blocks received
GC CR Blocks Receive [%]	Global cache CR blocks received as percentage
GC Gets	Number of global cache gets
GC Gets [%]	Global cache gets as percentage
GC Converts	Number of global cache converts
GC Converts [%]	Global cache converts as percentage
Logical Reads	Number of logical reads
Logical Reads [%]	Logical reads as percentage

Global Cache Flushes to Redo

This sub-monitor in the SAP/Oracle Database Monitor displays information on global cache flushes. The constant read (CR) block flush time is part of the service time for a CR buffer.

A block with pending redo records needs to be flushed to the log file by the log writer (LGWR) before the lock management server (LMS) can send it. The operation is asynchronous in that LMS queues the request, posts LGWR, and continues to process other blocking asynchronous system traps (BASTs).

LMS checks the log flush queue for completions and then sends the block, or goes to sleep and is posted by LGWR. The redo log write time and redo log sync time can significantly influence the overall service time. Each individual start time is maintained in the request structure that is put on the log flush queue, the log flush duration is computed, and the flush time statistics incremented when the request completes and is taken off the log flush queue. Most CR block transfers do not require log flushes.

You choose *Performance* ▸ *RAC Statistics* ▸ *Global Cache Flushes to Redo* in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
Flush time [s]	Flush time in seconds
# of Flushes	Number of flushes
Avg Flush Time [ms]	Average flush time in milliseconds
# of Total Blocks Served	Number of total blocks served
CR Blocks Flushed [%]	CR blocks flushed as percentage

Global Cache CR Block Hit Ratio

This sub-monitor in the SAP/Oracle Database Monitor displays information on the global cache constant read (CR) block hit ratio, which is the proportion of global cache operations resulting in CR buffer copies from other instances. This is a subset of global cache fusion block hit ratio. It lets you drill down into the global cache work distribution to identify the efficiency of the global cache.

You choose *Performance* *RAC Statistics* *Global Cache CR Block Hit Ratio* in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
GC CR Block Hit Ratio [%]	Global cache CR block hit ratio as percentage
GC Current Blocks Receive	Number of global cache current blocks received
GC Current Blocks Receive [%]	Global cache current blocks received as percentage
GC CR Blocks Receive	Number of global cache CR blocks received
GC CR Blocks Receive [%]	Global cache CR blocks received as percentage

Global Cache Defer Ratio

This sub-monitor in the SAP/Oracle Database Monitor displays information on the global cache defer ratio, which is the proportion of buffers served that had to be deferred. This indicates high concurrency for a small set of buffers.

You choose *Performance* *RAC Statistics* *Global Cache Defer Ratio* in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
GC Defer Hit Ratio [%]	Global cache defer hit ratio
GC Current Blocks Receive	Number of global cache current blocks received
GC Current Blocks Receive [%]	Global cache current blocks received as percentage
GC Defers	Number of global cache defers
GC Defers [%]	Global cache defers as percentage

Global Cache Current Block Hit Ratio

This sub-monitor in the SAP/Oracle Database Monitor displays information on the global cache current block hit ratio, which is the proportion of global cache operations resulting in current buffer copies from other instances, a subset of global cache fusion block hit ratio. It lets you drill down into the global cache work distribution and identify the efficiency of the global cache.

You choose *Performance* *RAC Statistics* *Global Cache Current Block Hit Ratio* in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
GC Current Block Hit Ratio [%]	Global cache current block bit ratio as percentage
GC Current Blocks Receive	Number of global cache current blocks received
GC Current Blocks Receive [%]	Global cache current blocks received as percentage
GC CR Blocks Receive	Number of global cache constant read (CR) blocks received
GC CR Blocks Receive [%]	Global cache CR blocks received as percentage

Global Cache Current Blocks Send

This submonitor in the SAP/Oracle Database Monitor displays information on the current block requests served by this instance.

The global cache server processes such as `LMS0` and `LMS1` maintain this information as a result of a fusion blocking asynchronous system trap (BAST), which is the notification mechanism used by the global cache service. The send time is measured when a block is handed to the IPC layer to be sent.

On some platforms and with certain protocol implementations, it is necessary to wait for send completions and the requests are queued in a completion queue. Therefore, the send time is measured from when a block is handed to the IPC layer until the protocol returns a completion, which indicates a successful send. The average latency is usually very small, in the microsecond range.

You choose ▶ *Performance* ▶ *RAC Statistics* ▶ *Global Cache Current Block Send* ◀ in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
GCS Current Blocks Send	Number of global cache service (GCS) blocks sent
GCS Current Blocks Send [%]	GCS blocks sent as percentage
GCS Current Blocks Send Time [s]	Send time for GCS current blocks in seconds
GCS Current Blocks Send Time	Send time for GCS current blocks as percentage

Column	Description
[%]	
Avg Current Blocks Send Time [ms]	Average send time for GCS current blocks in milliseconds

Global Cache GES Lock Gets

This sub-monitor in the SAP/Oracle Database Monitor displays information on global cache global enqueue service (GES) lock gets.

You choose *Performance* ▸ *RAC Statistics* ▸ *Global Cache GES Lock Gets* in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
Global Lock Gets	Number of global lock gets
Global Lock Gets [%]	Global lock gets as percentage
Global Lock Get Time [s]	Time for global lock get in seconds
Global Lock Get Time [%]	Time for global lock get as percentage
Avg Global Lock Get Time [ms]	Average time for global lock get in milliseconds

Overview System Statistics

This sub-monitor in the SAP/Oracle Database Monitor displays an overview of system statistics.

You choose *Performance* ▸ *RAC Statistics* ▸ *Overview System Statistics* in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
Statistic #	Statistic number
Statistic Name	Name of statistic
Statistic Value	Value of statistic
Part per Inst [%]	Contribution to the instance as percentage

Overview Library Cache

This sub-monitor in the SAP/Oracle Database Monitor displays an overview of the library cache.

You choose *Performance* *RAC Statistics* *Overview Library Cache* in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information:

Column	Description
Inst Id	Instance ID
Name	Instance name
Gets	Number of gets
Get Hits	Number of get hits
Get Hit Ratio	Get hit ratio
Pins	Number of pins
Pin Hits	Number of pin hits
Pin Hits Ratio	Pin hits ratio
Reloads	Number of reloads

Column	Description
Invalidations	Number of invalidations

Note

For more information on the meaning of terms such as *Pin* and *Invalidations*, see your Oracle documentation.

End of the note.

Current Block Server

This sub-monitor in the SAP/Oracle Database Monitor displays the contents of the view GV$CURRENT_BLOCK_SERVER, which provides information about the shipping of current blocks.

The monitor shows the amount of waits that have been incurred for each instance for log flushes, pins, and writes. These are categorized according to the length of the wait time. Pin times greater than 2 milliseconds indicate block contention at processing time.

You choose ▶ *Performance* ▶ *RAC Statistics* ▶ *Current Block Server* ◀ in the DBA Cockpit.

You can view history information in this monitor.

Structure

This screen contains the following information (the columns are shown with the variable <number> to avoid repetition):

Column	Description
Instance Id	Instance ID
Pin <number>	Number of waits less than or equal to <number> of milliseconds
Flush <number>	Number of flushes less than or equal to <number> of milliseconds
Write <number>	Number of waits less than or equal to <number> of milliseconds

Example

If the entry in column *Pin 1000* is *6*, this means that a block was pinned for 1000 milliseconds (that is, for one second) six times.

End of the example.

Note

For more information on the meaning of terms such as *Pin* and *Flush*, see your Oracle documentation.

End of the note.

Space (Oracle)

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

Space Overview

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
Database	
Name	Database name *DEFAULT* means the local database.
DB system	*ORA* for Oracle
Size	Size of the database in GB
Free size	Free space in the database in GB
Used	Percentage used space in database

Field	Description
	Equal to 100 * *Free size* / *Size*.
Users	
Total number	Number of database users
Maximum size	Maximum size used by a user The user name is in brackets.
Maximum # segments	Maximum number of segments used by a user The user name is in brackets.
Tablespaces	
Total number	Total number of tablespaces
Maximum size	Size of the biggest tablespace The tablespace name is in brackets.
Minimum free space	Minimum free space not used by a tablespace The tablespace name is in brackets.
Segments	
Number	Number of the objects *Tables*, *Indexes*, *Others*
Size	Total size of the objects in MB
More 100 ext.	Number of objects with more than 100 extents
Compressed	Number of compressed objects
No logging	Number of objects with “no logging”
Critical	Number of critical objects

Space Statistics Collector

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: XXXXXXXXXXXXXXXXXXXXXXXX`

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* ◀.

  End of the example.

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
`CON_NAME`	Database connection name, as stored in table DBCON
`STATUS`	Status of the database monitoring
`MAX RUNTIME`	Maximum runtime in seconds desired for collector modules Default: 1800 seconds
`MAX UPLOAD`	Maximum size in KB desired to be written to MONI per collector module Default: 1,000,000 KB
`LOG LIFE`	Time in days desired to retain logs Default: 100 days
`DESCRIPTION`	Short text

- We recommend that you do *not* normally modify table DB02_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 DB02_COLL_PLAN looks as follows:

Column	Description
`CON_NAME`	Database connection name, as stored in table DBCON
`SMON_ID`	Sub-monitor ID for this collector module

Column	Description
SMON_NAME	Sub-monitor name for this collector module
FUNCNAME	Function module name
MONIFILL	MONI filler, part of the MONI key
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.

Database Overview

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.

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
Date	Date of last analysis
Name	Database name
Size (Gb)	Database size in GB
Chg. Size/day	Change of database size per day in GB
Free (Gb)	Database free space in GB
Chg. Free/day	Change in database free space per day in GB
Used (%)	Percentage used space in the database
Chg. Used/day	Change in database used space per day
Users	Number of database users
Chg. Users/day	Change in number of users per day
Tablespaces	Number of tablespaces
Chg. Tablespaces/day	Change in number of tablespaces per day
Segments	Number of segments

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

Users Overview

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.

You choose ▶ *Space* ▶ *Users* ▶ *Overview* ◀ in the DBA Cockpit.

Structure

This screen contains the following information:

- The tab *Main Data*:

Column	Description
User name	User name
User id	User ID
Size (MB)	Size of data assigned to user in MB

Column	Description
# Segments	Number of segments owned by the user
# Tables	Number of tables owned by the user
# Indexes	Number of indexes owned by the user
# Extents	Number of extents owned by the user
Def. Tablespace	Name of default tablespace for user
Temp. tablespace	Name of temporary tablespace for user
Account status	Status of user account
Profile	Name of user profile
Creation Date	Date user account was created
Creation Time	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
Begin analysis date	Date when the analysis begins
End analysis date	Date when the analysis ends
User name	User name
User id	User ID
Size (MB)	Size in MB
Chg. Size/day	Average daily change in size in MB
# Segments	Number of segments
Chg. Segm/day	Average daily change in number of segments
# Tables	Number of tables
Chg. Tables/day	Average daily change in number of tables
# Indexes	Number of indexes
Chg. Indexes/day	Average daily change in number of indexes
# Extents	Number of extents

Column	Description
Chg. Extents/day	Average daily change in number of extents

Users Detailed Analysis

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.

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
User name	User name
User id	User ID
Password	Password
Account status	Status of the user account
Def. tablespace	Default tablespace name for user
Temp. tablespace	Temporary tablespace name for user
Profile	Profile name of user
Resource consumer gr.	Resource consumption group of user
External name	External name of user
Creation date	User creation date
Creation time	User creation time

Column	Description
Acc. lock date	Date user account was locked
Acc. expiry date	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*):

  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.

  End of the note.

Column	Description
Date	Analysis date
User name	User name
User id	User ID
Size (MB)	Size in MB
Chg. Size/day	Average daily change in size in MB
# Segments	Number of segments
Chg. Segm/day	Average daily change in number of segments
# Tables	Number of tables
Chg. Tables/day	Average daily change in number of tables
# Indexes	Number of indexes
Chg. Indexes/day	Average daily change in number of indexes
# Extents	Number of extents
Chg. Extents/day	Average daily change in number of extents

Tablespaces Overview

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.

You choose ▶ *Space* ▶ *Tablespaces* ▶ *Overview* ◀ in the DBA Cockpit.

Structure

This screen contains the following information:

- Tab *Main Data*:

Column	Description
Tablespace name	Name of tablespace
Size (MB)	Size of tablespace in MB
Free (MB)	Size of free space in the tablespace in MB
Used (%)	Percentage of available space used in the tablespace
Autoextend	Autoextend option
Total size (MB)	Total size of tablespace in MB
Sum of all free fragments	Total size of all free fragments in MB
Total used (%)	Percentage of the total space used in tablespace
# Files	Number of files in tablespace
# Segments	Number of segments in tablespace
# Extents	Number of extents in tablespace
Status	Status of tablespace
Contents	Type of tablespace data

- Tab *Storage Management*:

Column	Description
Tablespace name	Tablespace name
Extent management	Extent management tracking, *DICTIONARY* or *LOCAL*
Allocation type	Type of extent allocation for tablespace
SSM	Segment space management tracking, *AUTO* or *MANUAL*
Block size (KB)	Tablespace block size in KB
Init. extent (MB)	Default initial extent size of tablespace in MB
Next extent (MB)	Next extent size of tablespace in MB

Column	Description
Min. extents	Minimum number of extents for tablespace
Max. extents	Maximum number of extents for tablespace
PCT increase (%)	Percentage increase of extents for tablespace

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

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

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

Column	Description
Tablespace name	Tablespace name
Max free (MB)	Largest free fragment in tablespace in MB
Total free (MB)	Total size of all free fragments in tablespace in MB
# Fragments	Number of fragments in tablespace

Column	Description
Max. next extent	Maximum number of next extents in tablespace
Critical objects 1	Number of segments in tablespace that cannot allocate more than P extents
Critical objects 2	Number of segments in tablespace that cannot allocate more than P' but that can allocate more than P extents

- 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.
- End of the note.
- Sub-tab *Free fragments*:

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

- Tab *Summary history*:

Column	Description
Begin date	Start date for analysis
End date	End date for analysis
Tablespace name	Tablespace name
Size (MB)	Size of tablespace in MB
Chg. size/day	Change in used size of tablespace per day

Column	Description
Free (MB)	Free size in MB
Chg. free/day	Change in free space in tablespace per day
Used (%)	Percentage used space in tablespace
Chg. Used/day	Change in used space in tablespace per day
Autoextend	Autoextend flag, *YES* or *NO*
Total size (MB)	Total size of tablespace in MB
Change of total size per day (MB/day)	Change in total size of tablespace per day
Sum of all free fragments (MB)	Total of all free fragments in tablespace in MB
Chg. free/day (MB/day)	Change in free size in tablespace per day
Total used (%)	Percentage of total space used in tablespace
Chg. Total used /day (%/day)	Change in percentage of total space used in tablespace per day
# Files	Number of files in tablespace
Chg. #files/day	Change in number of files in tablespace per day
# Segments	Number of segments in tablespace
Chg. #segments/day	Change in number of segments in tablespace per day
# Extents	Number of extents
Chg. #extents/day	Change in number of extents in tablespace per day

Tablespaces Detailed Analysis

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.

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
Tablespace name	Tablespace name
Status	Tablespace status, *ONLINE*, *OFFLINE*, or *READ*.
Contents	Tablespace contents, *PERMANENT* or *TEMPORARY*
Logging	Default logging attribute for tablespace
Forced logging	Forced logging mode for tablespace
Extent management	Extent management tracking, *DICTIONARY* or *LOCAL*
Allocation type	Type of extent allocation for tablespace
Plugged in	Specifies whether a tablespace is transported to an instance
Segment space mgment	Segment space management tracking, *AUTO* or *MANUAL*
Def. table compression	Default table compression mode, *DISABLED* or *ENABLED*
Size (MB)	Size of tablespace in MB
Free (MB)	Size of free space in tablespace in MB
Block size (KB)	Tablespace in KB
Initial extent (MB)	Default initial extent size of tablespace in MB
Next extent (MB)	Default next extent size of tablespace in MB
Min. extents	Default minimum number of extents for tablespace
Max. extents	Default maximum number of extents for tablespace
PCT increase (%)	Default percent increase of extent size for tablespace
Min. Extent length (MB)	Minimum extent size for tablespace in MB

- The tab *Files*:

Column	Description
File name	File name

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

- The tab *Freespace*:

Column	Description
Tablespace name	Tablespace name to which segment belongs
File id	ID of database data file containing the fragment
Block Id	Starting block number of file containing the fragment
Free (MB)	Size of fragment in MB
# Blocks	Size of fragment in Oracle blocks
Relative file number	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*):

  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.

  End of the note.

Column	Description

Column	Description
Date	Date of analysis
Tablespace name	Name of tablespace
Size (MB)	Tablespace size in MB
Maxsize (MB)	Maximum size of tablespace in MB
Free (MB)	Size of free space in tablespace in MB
Maxblocks	Maximum size of tablespace in Oracle blocks
Increment by	Default increment for autoextension
User size (MB)	Total space available for data in the tablespace or datafile in MB
Autoextend	Autoextensibe indicator for tablespace
User blocks	Number of blocks available for data in the tablespace
Block id	Starting block number of file containing tablespace
Total free (MB)	Total size of all free fragments in tablespace in MB
Chg. free/day (MB/day)	Change in free size in tablespace per day
Total used (%)	Percentage of total space used in tablespace
Chg. Total used/day (%/day)	Change in percentage of total space used in tablespace per day
# Files	Number of files in tablespace
Chg. #files/day	Change in number of files in tablespace per day
# Segments	Number of segments in tablespace
Chg. #segments/day	Change in number of segments in tablespace per day
# Extents	Number of extents
Chg. #extents/day	Change in number of extents in tablespace per day

Segments Overview

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.

You choose ▶ *Space* ▶*Segments* ▶*Overview* ◀in the DBA Cockpit.

Structure

This screen contains the following information:

- *Main Data* shows all segments

Column	Description
Type	Segment type
Size (MB)	Segment size in MB
Avg. Size (MB)	Average segment size in MB
# Segments	Number of segments
# Extents	Number of extents
More 100 Extents	Number of segments of the specified segment type with more than 100 extents
Compressed	Number of segments of the specified segment type that are compressed to reduce disk use
No Logging	Number of segments of the specified segment type that do not log creation or modification in the redo log file

- *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
Owner	Segment owner
Name	Segment name
Partition	Partition name
Type	Segment type
Tablespace	Tablespace name of segment
Size (MB)	Segment size in MB

Column	Description
# Extents	Number of extents
# Blocks	Number of blocks
Next Extent (MB)	Size of next extent in MB

- *Top Growth* shows growth data for segments – the sub-tab *Days* is shown below (the columns are analogous for *Weeks* and *Months*)

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

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

Column	Description
Owner	Segment owner
Name	Segment name
Partition	Partition name
Type	Segment type
Tablespace	Tablespace name of segment
Size (MB)	Segment size in MB
Chg. Size/month	Change in size per month
# Extents	Number of extents

Column	Description
# Blocks	Number of blocks
Next Extent (MB)	Size of next extent in MB

- *Extents Critical Objects*

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

Column	Description
Begin date	Start date for analysis
End date	End date for analysis
Type	Segment type
Size (MB)	Segment size in MB
Chg. Size/day	Change in segment size per day
Avg. Size (MB)	Average segment size in MB
Chg. Avg. Size/Day	Change in average segment size per day
# Segments	Number of segments
Chg. Segments/Day	Change in number of segments per day
#Extents	Number of extents

Column	Description
Chg. Extents/day	Change in number of extents per day
More 100 Extents	Number of segments of the specified segment type that have more than 100 extents
Chg. More 100 Extents/day	Change in number of segments of the specified segment type that have more than 100 extents per day
Compressed	Number of segments of the specified segment type that are compressed to reduce disk use
Chg. Compressed/Day	Change in number of segments of the specified segment type that are compressed to reduce disk use per day
Nologging	Number of segments of the specified segment type that do not log creation or modification in the redo log file
Chg. Nologging/day	Change in number of segments of the specified segment type that do not log creation or modification in the redo log file per day

Segments Detailed Analysis

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.

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
Owner	Owner name
Segment Name	Segment name
Partition Name	Partition name

Column	Description
Type	Segment type
Tablespace	Tablespace for segment
Header File	Number of header files
Header Block	Number of header blocks
Freelists	Number of process freelists allocated in this segment
Freelist Group	Number of freelist groups allocated in this segment
Relative FNO	Relative number of the file containing the segment header
Buffer Pool	Buffer pool
Size [MB]	Size in MB
Extents	Number of extents
Blocks	Number of blocks
Initial Extents [MB]	Size in MB of the initial extent of the segment
Next Extents [MB]	Size in MB of the next extent to be allocated to the segment
Min Extents	Number of minimum extents allowed in the segment
Max Extents	Number of maximum extents allowed in the segment
PCT increase	Percent increase in the size of the next extent to be allocated
Compressed	Number of segments of the specified segment type that are compressed to reduce disk use
NoLogging	Number of segments of the specified segment type that do not log creation or modification in the redo log file

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

Column	Description
Owner	Owner name
Segment	Segment name
Partition	Partition name
Type	Segment type

Column	Description
Tablespace	Tablespace for segment
Size (MB)	Segment size in MB
# Extents	Number of extents
# Blocks	Number of blocks
Initial Extents [MB]	Size in MB of the initial extent of the segment
Next Extents [MB]	Size in MB of the next extent to be allocated to the segment
Min. Extents	Minimum number of extents allowed in the segment
Max. Extents	Maximum number of extents allowed in the segment
Pct. Increase	Percent increase in the size of the next extent to be allocated
Buffer Pool	The default buffer pool to be used for segments blocks

- The tab *Extents*:

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

- The tab *Storage*:

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

Column	Description

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

Column	Description
Avg. initial length (Bytes)	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*):

  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.

  End of the note.

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

Segments Detailed Analysis (Aggregated)

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.

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
Owner	Name of segment owner
Object name	Object name
Object type	Object type
Tablespace	Tablespace name
Partitioned	Whether segment is partitioned
Lobs	Whether segment has lobs
Size [MB]	Size in MB
Extents	Number of extents
Blocks	Number of blocks
# Partitions	Number of partitions
# Lobs	Number of lobs

- *Tables + Indexes*

Column	Description
Owner	Name of object owner
Object	Object name
Object type	Object type
Tablespace	Tablespace name

Column	Description
Size [MB]	Size of object in MB
Extents	Number of extents
Blocks	Number of blocks
Partitioned	Whether object is partitioned
# Partitions	Number of partitions
Lobs	Whether object has lobs
# Lobs	Number of lobs

- *Partitions*

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

- *Lobs*

  The information shown here is the same as the table above.

- *History*

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.

End of the note.

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

Collector Logs

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

You choose ▶ *Space* ▶ *Additional Functions* ▶ *Collector Logs* ◀ in the DBA Cockpit.

Structure

This screen contains the following information:

- The tab *Overview*:

Column	Description

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

- The tab *Modules* with more statistical information and logs for each collector module:

Column	Description
Module ID	Collector module ID
Module name	Collector module name
Status	Current status
Last col. date	Date of last collection
Last col. time	Time of last collection
Last upload (KB)	Size of last collection in KB
Last duration	Duration of last collection
Avg. upload (KB)	Average upload within the previous `LOG_LIFE` days
Avg. duration	Average duration within the previous `LOG_LIFE` days

Column	Description
Max. upload (KB)	Largest upload within the previous `LOG_LIFE` days
Date max. upload	Date of largest upload
Max. duration	Longest runtime within the previous `LOG_LIFE` days
Date max. duration	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.

BW Analysis

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
BW Area	BW area
BW Object Type	BW object type
Details	Details

Column	Description
# BW Objects	Number of BW objects
# BW Root Objects	Number of BW root objects
# Partitions	Number of partitions
Size [MB]	Size in MB
Extents	Number of extents
Blocks	Number of blocks

- *Main Data*
- *Partitions* and *LOBs*

Column	Description
Owner	Name of partition or lob owner
Root name	Root name of partition or lob
Sub component name	Sub-component name of partition or lob
Sub component type	Sub-component type of partition or lob
Tablespace	Tablespace name of partition or lob
Size [MB]	Size of partition or lob in MB
# Extents	Number of extents for partition or lob
# Blocks	Number of blocks for partition or lob
Initial extents	Number of initial extents for partition or lob
Next extents	Number of next extents for partition or lob
Minimum extents	Minimum number of next extents for partition or lob
Maximum extents	Maximum number of extents for partition or lob
Percentage increase	Percentage increase in number of extents allocated
Buffer pool	Buffer pool of the partition

Jobs (Oracle)

You can use the DBA Cockpit for the following jobs:

- DBA Planning Calendar
- DBA logs
- Backup logs

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.

> 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:

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

  3 FUD 2

  On system FUD for Thursday 8th 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*.

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 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*:

3	*FUD*	*2*
4	*FIB*	*1*

   On system FUD for Thursday 8th 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:

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

   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.

DBA Planning Calendar (Oracle)

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.

Note

The old DBA Planning Calendar, which runs outside the DBA Cockpit, is still available but is no longer documented here. For more information, see earlier releases of the online documentation in SAP Library, which you can find at:

https://help.sap.com ▶ *SAP NetWeaver*

You can start the old DBA Planning Calender with transaction DB13OLD.

The documentation here concerns *only* the new DBA Planning Calendar, which runs in the DBA Cockpit.

End of the note.

Caution

Do *not* run the old and the new DBA Planning Calendar together as this causes problems.

End of the caution.

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.

  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:

  End of the note.

Color	Meaning
Light blue	The action has not yet started.
Dark blue	The action has not yet finished.
Green	The action has run successfully.
Yellow	The action has finished with a warning. Check the job log for details.
Red	An error has occurred and the action was interrupted. Check the job log for details and reschedule the action.

Color	Meaning
Dark yellow	No more information is available.
Dark red	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.

More Information

Setting Up the DBA Planning Calendar

Setting Up the DBA Planning Calendar (Oracle)

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.

End of the example.

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

End of the caution.

You can schedule the following actions with the DBA Planning Calendar:

Tasks	Actions
Database backup and recovery	*Whole database offline + redo log backup*
	Whole database online backup
	Whole database offline backup
	Whole database online + redo log backup
	Full database offline + redo log backup
	Full database online backup
	Full database offline backup
	Full database online + redo log backup
	Incremental database offline + redo log backup
	Incremental database online backup
	Incremental database offline backup
	Incremental database online + redo log backup
	Partial database offline backup
	Partial database online
	Prepare for RMAN Backup
	Redo log backup
Space Management	*Adapt next extents*
	Compress database
Other	*Check database*
	Check and update optimizer statistics
	Cleanup logs
	Initialize tape
	Validate structure
	Verify database

Tasks	Actions
	Central Calendar Log Mirror

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 initialization profile 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 Calender in production operation, make sure that you check the results daily to make sure that actions have been executed correctly.

End of the caution.

Configuring the Back End of the DBA Planning Calender (Oracle)

You can configure the back end of the DBA Planning Calender 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. From the DBA Planning Calendar, choose ▶ *Goto* ▶ *Backend Configuration* ◀.
2. In *Selected Scope*, choose the scope of the configuration entry.
3. In *Background Dispatching*, choose appropriate values as follows:

Field	Meaning
Background Server	Specifies the server, where scheduled jobs are executed If no server is specified, the background job dispatcher dynamically selects the server.
Job Priority	Specifies the priority of the job If no priority is specified, jobs run with default priority (medium).
User	Name of the user to execute the job If no user is specified, the dialog user is used.

4. Save your changes.

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.

End of the note.

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.

Parameter	**Description**
Recurrence Pattern	Interval for the action in weeks, days or hours Depending on the selected recurrence pattern, you need to specify the pattern in more detail, that is the days of the week for weeks and the hours of the day for a daily period. The action is repeated at the interval you enter. If you select *Once only*, the action is executed only once.
Recurrence Range	Range of time where the action recurs, that is for a specific time interval or for a limited number of occurrences

5. Caution

6. The system warns you if there is a conflict with an existing action, but it does *not* prevent you from inserting the new action.

7. 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.

8. End of the caution.

9. To schedule the action, choose *Add*.

Changing an Action (Oracle)

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).

Note

If an action has already been executed, you can only display it.

End of the note.

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*.

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).

Note

If an action has already been executed, you can only display it.

End of the note.

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*.

Re-Executing an Action (Oracle)

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.

End of the example.

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*.

   End of the note.

The action is scheduled for immediate re-execution.

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.

   End of the note.

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*.

Displaying Scheduled Actions (Oracle)

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.

   End of the caution.

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.

Troubleshooting (Oracle)

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.

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.

End of the note.

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.

Displaying DBA Logs (Oracle)

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 *Jobs* *DBA Logs* .

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
Operation start	Date and time for the start of the operation
Operation end	Date and time for the end of the operation
FID	Function ID indicating type of operation
RC	Return code For more information, see Messages and Return Codes for the BR Tools.
Database Operation	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.

Display of the Backup Logs (Oracle)

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).

- *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).

- *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).

Displaying Logs for Database Backups (Oracle)

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:

   o 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

Heading	Description
Tape	Tape name
Position	Position of the file on the tape
Backup time	Time the backup of the file ended
Compressn	Compression rate (if compression was used)
DF ID	Oracle data file ID
Backint ID	Backup ID of Backint interface (only if external backup program was used)
Redo No.	Log sequence number of the online redo log file
Tablespace name	Tablespace name
TS Status	Oracle tablespace status
DF Status	Oracle data file status
File Name	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 BRBACKUP Detail Log.

- To see an overview of all existing backup logs, choose *Overview of database backups* (backups with errors are highlighted in red):

Heading	Description
Backup function	Type of backup
Start of backup	Time the backup started
End of backup	Time the backup ended
RC	BRBACKUP return code For more information, see Messages and Return Codes for the BR Tools.
Action ID/Log	File system log name `<Action ID>.<Function ID>` For more information, see Names of the BRBACKUP Detail Logs.

- 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 BRBACKUP Detail Log.

Displaying Logs for Redo Log Backups (Oracle)

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
Redo No	Log sequence number
Archiving Time	Time the offline redo log file was created
Tape	Tape name
Position	Position of the file on the tape
Backup time	Time the offline redo log file was backed up
Backint ID	Backup ID of Backint interface (only if external backup program was used)
Offline Redo Log Name	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
Function	Type of backup
Operation Start	Time redo log backup started
Operation End	Time redo log backup ended
RC	BRARCHIVE return code

Heading	Description
	For more information, see Messages and Return Codes for the BR Tools.
Action ID/Log	File system log name `<Action ID>.<Function ID>` For more information, see Names of the BRARCHIVE Detail Logs.

- To see more information about a redo log backup, double-click the appropriate line to display the BRARCHIVE action log:

Heading	Description
Function	Backup functions: ▪ *Save* Creates first copy of the offline redo log file ▪ *Copy* Creates second copy of the offline redo log file ▪ *Del* Deletes offline redo log file from the archiving directory
RedoNo.	Log sequence number
Archiving time	Time the offline redo log file was created (that is, archived)
Tape name	Tape name
Pos.	Position of the file on the tape
Backup time	Time the offline redo log file was backed up
Compr	Compression rate (if compression was used)
Backint ID	Backup ID of Backint interface (only if external backup program was used)
Offline redo log file name	Directory and name of the 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 BRARCHIVE Detail Log.

Displaying the Recovery Report (Oracle)

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 display 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.

More Information

Restore and Recovery

Restore and Recovery with BR*Tools