

The Self-Managing Database: Proactive Space & Schema Object Management

An Oracle White Paper
Nov. 2003

The Self-Managing Database: Proactive Space & Schema Object Management

| | |
|--|----|
| Introduction..... | 3 |
| Proactive Tablespace Management | 3 |
| Overview..... | 3 |
| Tablespace Alerts..... | 4 |
| Special Tablespaces | 6 |
| Efficient, Real Time Tablespace Monitoring | 7 |
| Querying and Setting Thresholds using Enterprise Manager | 8 |
| Default User Tablespace | 10 |
| Segment Management..... | 10 |
| Create Table Estimator – how many disks would I need?..... | 11 |
| Growth Trend Report – how many more disks will I need for the Christmas rush?..... | 12 |
| Segment Advisor – Can I reclaim some space?..... | 13 |
| Segment Shrink? – Let's reclaim free space!! | 14 |
| Undo Management..... | 16 |
| Auto-sizing Undo Tablespace..... | 16 |
| Auto Tuning of Undo Retention | 17 |
| Undo Advisor..... | 18 |
| Proactive Undo Tablespace Monitoring | 19 |
| Fast Ramp-up | 19 |
| Guaranteed Availability of Undo..... | 19 |
| Conclusion | 20 |

The Self-Managing Database: Proactive Space & Schema Object Management

INTRODUCTION

Space management is one of the most time consuming tasks for database administrators. According to a survey conducted by Oracle, DBAs currently spend approximately 20% of their time in performing space management operations. Fortunately, the Oracle Database 10g automatically manages its space consumption, alerts administrators on potential space problems, and recommends possible solutions. This paper details the solutions that enable administrators to maintain optimal space distribution and, address space related problems in an easy and timely manner. The paper covers the various components of Tablespace Management, Segment Management and Undo Management that make the Oracle Database 10g a self-managing database.

PROACTIVE TABLESPACE MANAGEMENT

Overview

The Proactive Tablespace Management (PTM) capability in the Oracle Database 10g brings efficient and powerful space monitoring, notification and space trending to the Oracle Database. Prior to Oracle Database 10g, the tools available for monitoring and setting up notifications regularly polled the database to monitor its space usage. Querying space usage information requires collecting data about the state of the database – state that is constantly changing in a production system. Because such queries are inherently expensive, the space monitoring tools typically run them infrequently, once a day or once every couple of hours. When they are run, the queries steal CPU, IO and memory (especially the buffer cache) resources away from critical business activity in the production system. It's a health check that is either late or hurts the health of the system or worse, both!

With PTM, Oracle Database 10g introduces a *non-intrusive and timely health check* of space in the database server. And best of all, PTM is available *by default, causes no measurable performance impact, and is uniformly available across all tablespace types*. Also, the same functionality is available both through Enterprise Manager (EM) as well as SQL. PTM is available for use with locally managed tablespaces. The health check is performed incrementally as

space is allocated and freed up in the database server. This guarantees immediate availability of space usage information whenever the user needs it. In addition to computing it on-the-fly, space usage information is assembled (across nodes in RAC) and alerts are pushed *proactively* every 10 minutes.

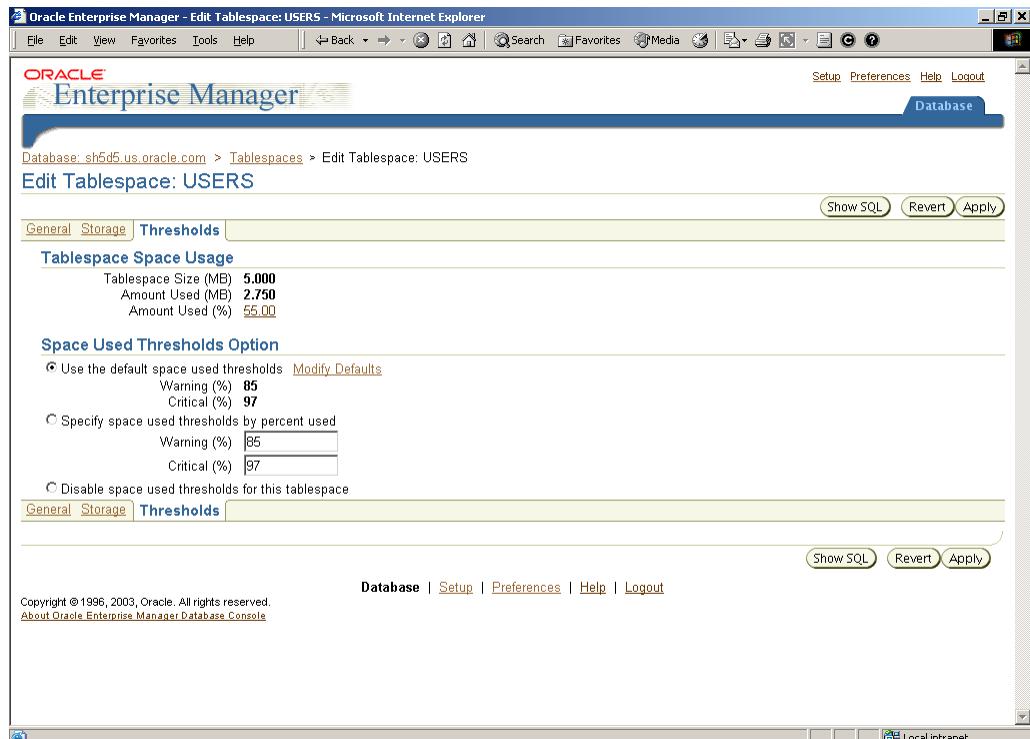


Fig 1: Viewing and Setting Alert Thresholds Using Enterprise Manager

Tablespace Alerts

The Oracle Database 10g provides the complete lifecycle around Tablespace Alerts right into the database kernel. Alert conditions are automatically computed and a notification is sent to the administrator if required. Once the underlying problem is fixed, any outstanding alerts are automatically cleared and moved to alert history.

Alerts are pushed into the server side table that holds alerts (available through DBA_OUTSTANDING_ALERTS) and are also made available through EM. When an alert is cleared, it is archived in the alerts history table available through DBA_ALERT_HISTORY.

Notification is performed using server generated alerts mechanism, which is yet another new functionality in the Oracle Database 10g. The alerts are triggered when certain space related events occur in the database. For example when the space usage threshold of a tablespace is crossed, an alert is raised. Another example of an alert is when a *Resumable* session encounters an out of space situation. An alert is sent instantaneously to the DBA to take corrective measures. The DBA may choose to get paged with the alert information and add

space to the tablespace to allow the *suspended* operation to continue from where it left off.

The database comes with a default set of alert thresholds. The DBA may override the default for a given tablespace or set a new default for the entire database through EM. The DBA may invoke the same functionality directly using DBMS_SERVER_ALERT.SET_THRESHOLD procedure. Also, current thresholds can be viewed using EM or through the DBMS_SERVER_ALERT.GET_THRESHOLD procedure.

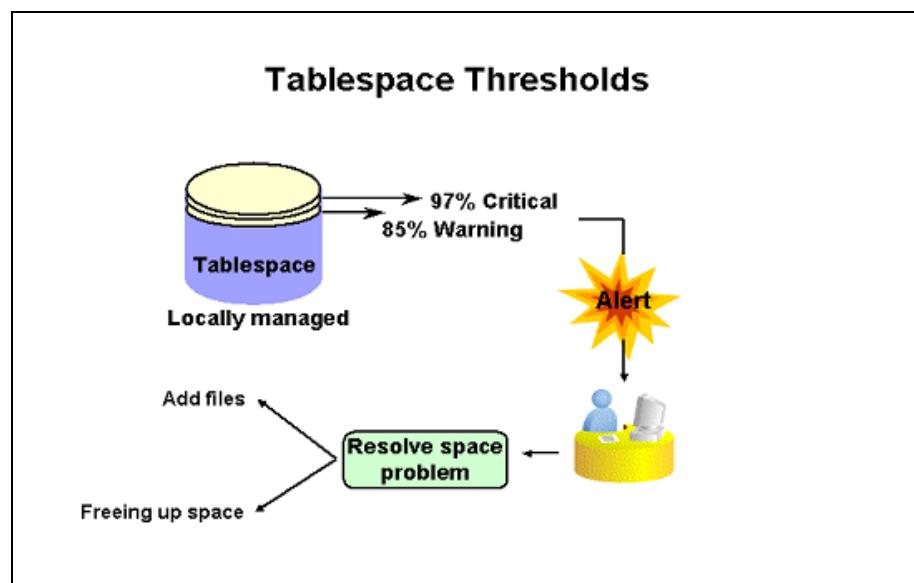


Fig 2: Tablespace threshold levels

Tablespace thresholds are defined in terms of the fullness of the tablespace as a percentage. Critical and warning thresholds are two thresholds that apply to tablespaces. The threshold values can be viewed and modified using Enterprise Manager or the DBMS_SERVER_ALERTS package. When the tablespace space utilization crosses either of these two limits, appropriate alerts are raised. Conversely, when the space utilization falls below these thresholds, any corresponding outstanding alert is cleared.

The user can specify the critical or warning threshold values. If the user does not specify a value, then a default of 85% and 97% are chosen for warning and threshold values respectively. For databases migrating from previous releases, the threshold values will be NULL – this indicates that tablespace threshold alerts will not be raised until the DBA explicitly enables them. The database wide default can be reset or modified when desired.

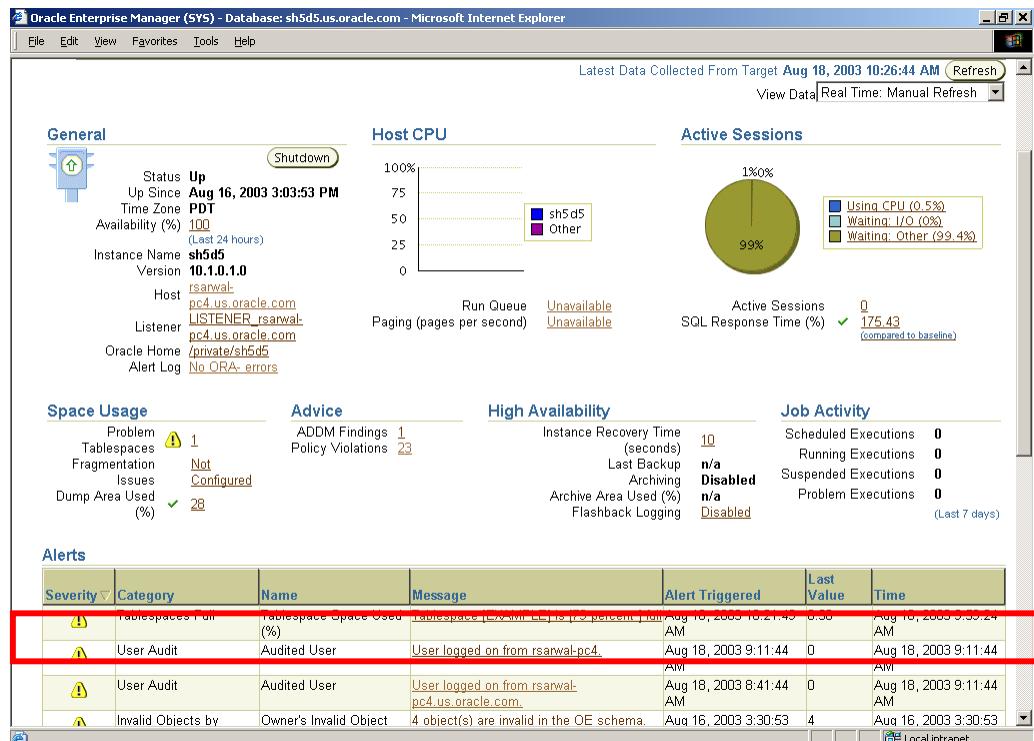


Fig 3: Tablespace Alerts Displayed on Enterprise Manager Database Home Page

Special Tablespaces

Space in an Undo tablespace may be in one of four states:

- Not allocated to segments
- Allocated space being used by active transactions
- Allocated space being retained for queries and Flashback
- Allocated space not being retained

Allocated space being used by active transactions and the allocated space being retained for queries/Flashback is considered as the “used space” for the purpose of comparing it against the threshold. This is a key distinction from user tablespaces.

For temporary tablespaces, “used space” is the space currently allocated to sessions and being actively used. For temporary tablespace groups, thresholds are set and evaluated on individual tablespaces.

For read-only and offline tablespaces, alerts are disabled.

For tablespaces with autoextensible files, the maximum tablespace size is computed by the maximum file size declared in the create/alter statement.

Efficient, Real Time Tablespace Monitoring

The key aspect of tablespace space monitoring is that the database server tracks space utilization while performing regular space management operations, e.g. allocating a new extent. When alert thresholds have been specified, every 10 minutes, a background process compares the state of the tablespaces against the thresholds specified to determine if an alert should be raised or cleared. If a state transition has occurred, the first instance (in a RAC environment) that detects this triggers an appropriate action.

At the core of the alert infrastructure is the active space monitoring built into the database server. Every instance in RAC maintains its own local information of the state of various tablespaces. This information is aggregated every 10 minutes to compute the global picture of the state of the database. In a non-RAC environment this step is simply skipped. The aggregate information is incrementally computed every 10 minutes to minimize the cost of this operation – only data for tablespaces that have changed their space allocation profile is considered. This is accomplished by using System Commit Numbers (SCNs) to track changes to the state of a tablespace. It is important to note here that even though the alert condition evaluation is done every 10 minutes, monitoring of space is done in the real time whenever any space management operation is performed. In other words, unlike external polling based management tools, the Oracle Database 10g collects data about its space utilization simultaneously with regular space management tasks and does not impose any additional overhead. This also ensures that information is collected just in time and only if there is any change in the database state as against an external tool which must query the database periodically regardless of whether there is any activity or not.

These inter-node optimizations combined with a very lightweight implementation of active space monitoring provide an efficient and scalable technology to maintain accurate and up to the minute state of the system.

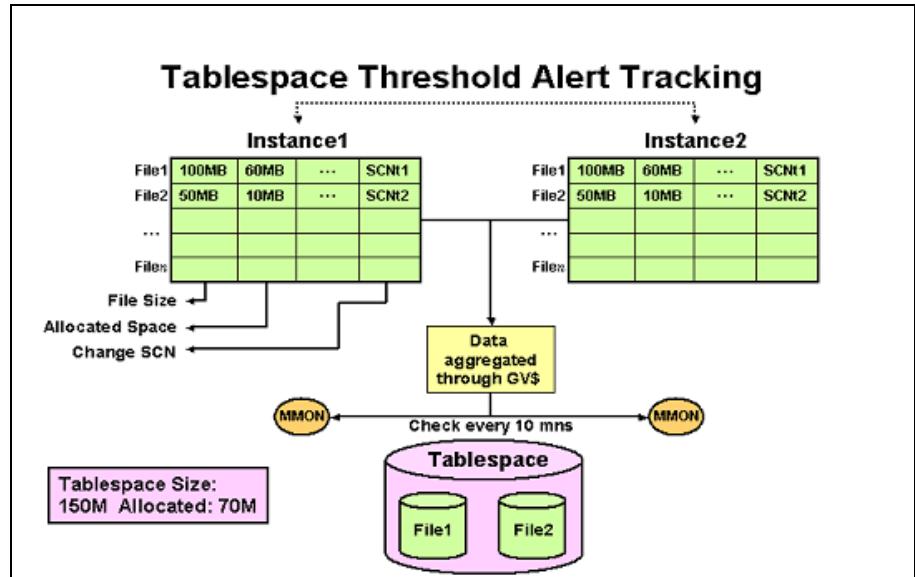


Fig 4: Tablespace Space Threshold Evaluation

Querying and Setting Thresholds using Enterprise Manager

EM provides an easy but powerful set of UI to manage database thresholds. Database wide defaults for the thresholds may be set from the Edit Metric Thresholds page as seen in the graphic below. This page can be launched from the database home page by selecting Edit Metric Thresholds under Related Links. This link will allow users to set the threshold values for all tablespaces.

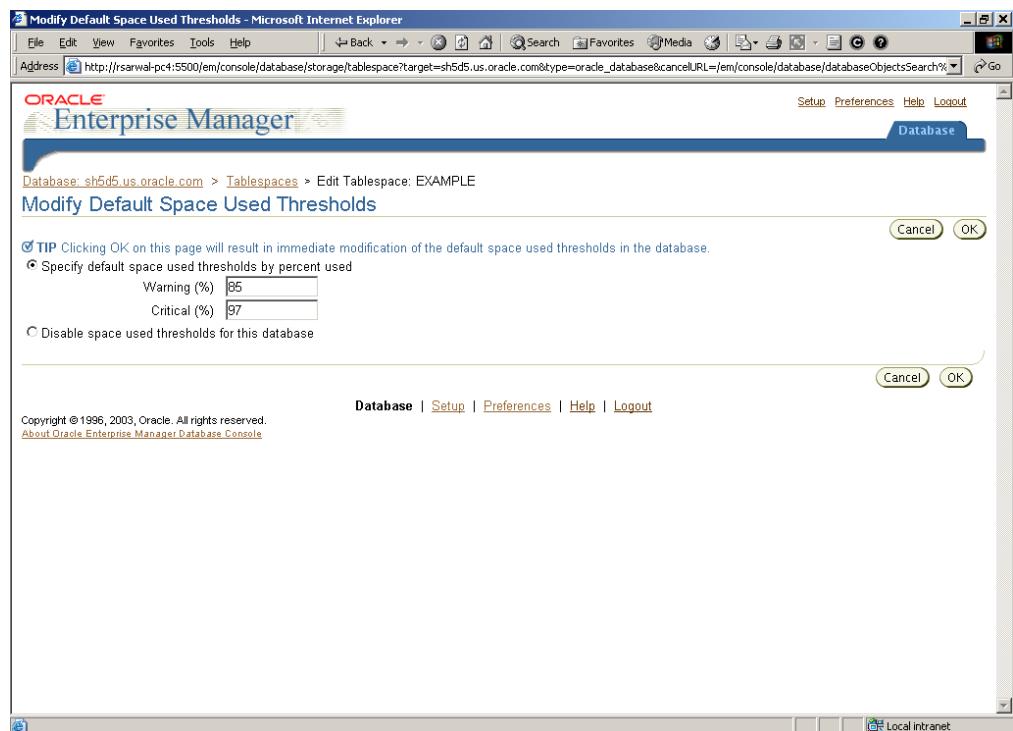


Fig 5: Viewing and Modifying Default Database Level Thresholds using Enterprise Manager

Warning and Critical tablespace thresholds may also be set on a specific tablespace directly from the Tablespaces page. To navigate to this page, start from the Tablespaces page, then select the tablespace, then click Thresholds.

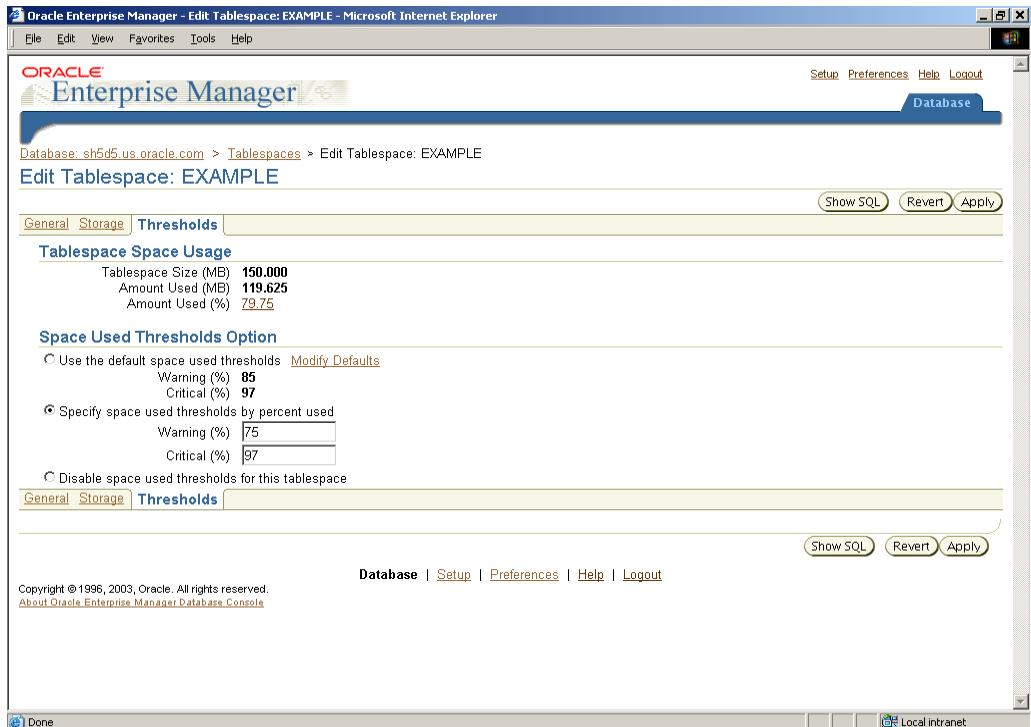


Fig 6: Viewing and Modifying Thresholds for A Given Tablespace

Default User Tablespace

Oracle Database 10g also provides the ability to define a given tablespace as the default user tablespace. When objects are created without the tablespace clause, they are automatically placed in the default permanent tablespace (not in SYSTEM tablespace, as previously). This simplifies manageability by making sure that objects don't get accidentally created in the SYSTEM tablespace. This feature is similar to the default temporary tablespace functionality introduced in Oracle9*i*.

SEGMENT MANAGEMENT

DBAs face a lot of challenges in managing space allocated to segments. Space may get over-allocated because of the difficulty to predict the space requirement of an object or because of the inability to predict the growth trend of an object. On tables that are heavily updated, the resulting segment may have a lot of internal fragmentation and maybe even row chaining. These issues may result in a wide variety of problem symptoms in the database, from poor performance to space wastage. This section describes the new functionality introduced in the Oracle Database 10g to address these challenges.

Create Table Estimator – how many disks would I need?

The Oracle Database 10g can predict the size of a given table based on its stricture and estimated number of rows. This enables “what if” analysis that allows estimation of the size of an object before it is created or rebuilt. If tablespaces have different extent management policies then the tool will help decide the tablespace that will cause least internal fragmentation. The Object Space Advisor works for tables with indexes on them as well as Index Organized Tables. The inputs to the cost estimator can be

Average row length, number of rows, PCTFREE, tablespace OR

Column description (data type, size), number of rows, PCTFREE, tablespace PCTFREE is optional.

This functionality can be used either through EM or command line interface. In Enterprise Manager, a new “Estimate Size” option is available while creating new tables or indexes. The same operation can be performed from command line using the DBMS_SPACE package.

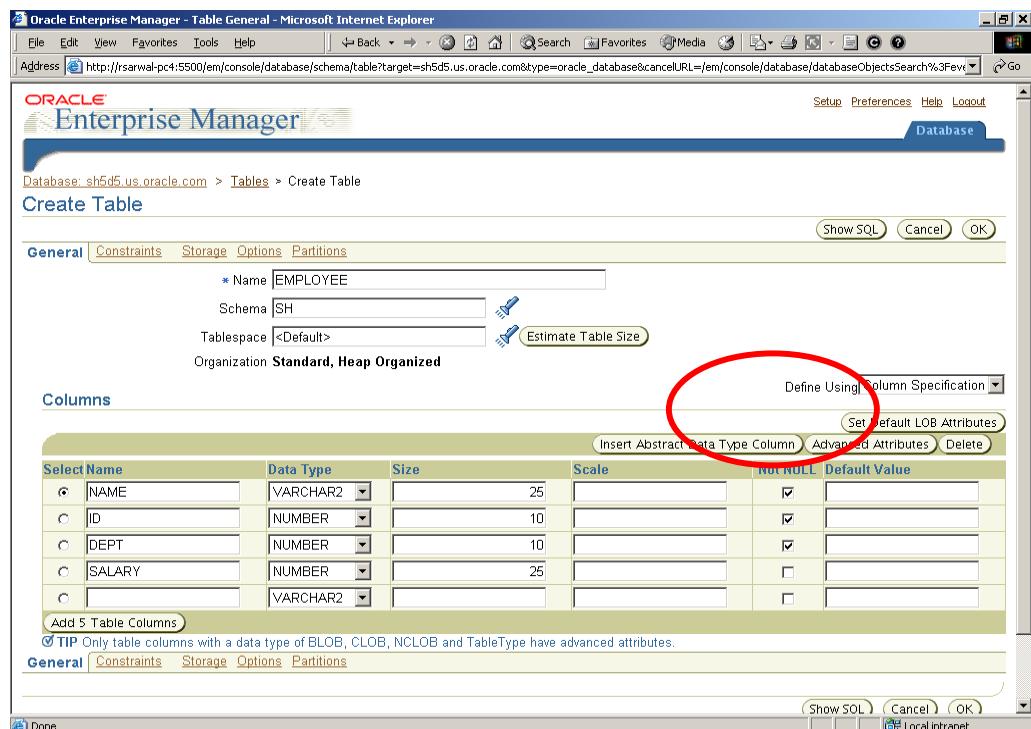


Fig 7: Estimating the size of a table before its creation

The computation of the estimated size takes into account all the internal structures that occupy space. For example, bitmap blocks, extent map blocks, index branch blocks (in case of Index Organized Tables) are accounted in the

space estimate. Oracle takes care of the internal metadata overhead to give the user the correct answer.

By predicting the size of schema objects even before they have been created, the Oracle Database 10g helps administrator plan up front and avoid any capacity issues subsequently.

Growth Trend Report – how many more disks will I need for the Christmas rush?

The growth trend report takes the DBA to the next step of capacity planning – planning for growth. Most database systems grow over time. Planning for growth is an important aspect of provisioning resources. If the DBA plans for growth well in advance, out of space conditions can be avoided from occurring. Of course, if and when they do occur, the Tablespace Alerts will provide you with an early warning of the impending problem. At that time the user may want to identify what objects are growing and predict the space utilization of the future.

At the heart of both, the growth trend reporting as well as the Segment Advisor (described later), is a built in mechanism to incrementally track space utilization in every segment in the database. This is done incrementally and with a high performance, low impact design along the lines of active space monitoring for Tablespace Alerts as described previously. The active space monitoring of individual segments in the database makes available to the database, the up to the minute status of individual segments in the system. This information is periodically written to the automatic workload repository. This provides sufficient information over time to perform growth trending of individual objects in the database as well as the database as a whole. The graphic below shows the EM page that shows the graphical representation of table “COUNTRIES” in the schema “HR”.

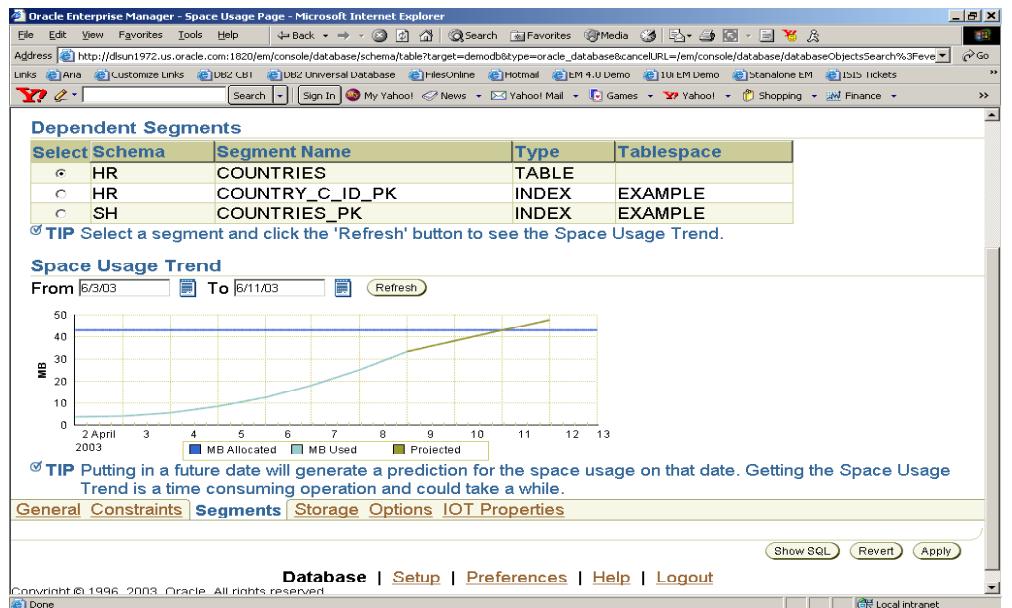


Fig 8: Viewing object growth trend

The Growth Trend Report comprises of two components: 1. Growth history 2. Growth forecast. Growth history is simply an interpolation of space usage data collected by the database server on a continuous basis. The Growth forecast is computed using straight-line projection. The slope of the line is determined using linear regression (LR). LR determines the slope of the line that best fits the data points. The slope is determined by finding the line that minimizes the sum of squares of the vertical distance of the data points. This technique will cause the spikes to be eliminated.

Segment Advisor – Can I reclaim some space?

The Oracle Database 10g introduces the ability to reclaim space from a segment by shrinking it. Shrinking of a segment will make unused space available to other segments in the tablespace and may improve the performance of queries and DML operations. To go with Segment Shrink, is the Segment Advisor. The Segment Advisor analyzes the specified tablespace or a schema for objects that may be candidates for shrinking.

The Segment Advisor performs growth trend analysis on individual objects to determine if there will be any additional space left in the object in 7 days. It then uses the reclaim space target to select candidate objects to shrink. The Segment Advisor can be invoked in the *comprehensive* mode. In this mode, in addition to using the pre-computed statistics in the workload repository, the Segment Advisor performs sampling of the objects under consideration to refine the statistics for the objects. Although this operation is more resource intensive, it may be used to perform a more accurate analysis, when desired.

In the Enterprise Manager, the Segment Advisor may be navigated to from one of three locations: from the tablespace page by selecting the Segment Advisor menu item, Schema Object page by selecting the Segment Advisor menu item or from the Advisor Central. The Segment Advisor may also be scheduled to run at certain time (like a batch window) or even created as a repeating task (to be run once a week on the OLTP tablespace, for example).

The analysis may also be limited to run within a certain interval. The Segment Advisor returns within the time limited limit. The recommendation from the Segment Advisor can be retained in the system for a specified period of time.

Segment Shrink? – Let's reclaim free space!!

The Oracle Database 10g provides the ability to perform *an in place reorganization* of data for optimal space utilization by shrinking it. As the name suggests, it also provides the ability to both compact the space used in a segment and then deallocate it from the segment. The deallocated space is returned to the tablespace and is available to other objects in the tablespace. Sparsely populated tables may cause a performance problem for full table scans. By performing shrink, data in the table is compacted and the high water mark of the segment is pushed down. This makes full table scans read less blocks and hence, run faster. Also, during compaction, row chaining is eliminated whenever possible.

Segment Shrink is an *online* operation – the table being shrunk is open to queries and DML while the segment is being shrunk. Additionally, segment shrink is performed *in place*. This is a key advantage over performing Online Table Redefinition for compaction and reclaiming space. The DBA may schedule segment shrink for one or all the objects in the database as nightly jobs without requiring any additional space to be provided to the database.

Segment Shrink works on heaps, IOTs, LOBs, Materialized Views and Indexes with row movement enabled in tablespaces with Automatic Segment Space Management. When Segment Shrink is performed on tables with indexes on them, the indexes are automatically maintained when rows are moved around for compaction. User defined triggers are not fired, however, because compaction is a purely physical operation and does not impact the application. Note that Segment Shrink can be performed only on tables with row movement enabled. Only applications that explicitly track rowids of objects will not be able to be shrunk because the application tracks the physical location of rows in the objects.

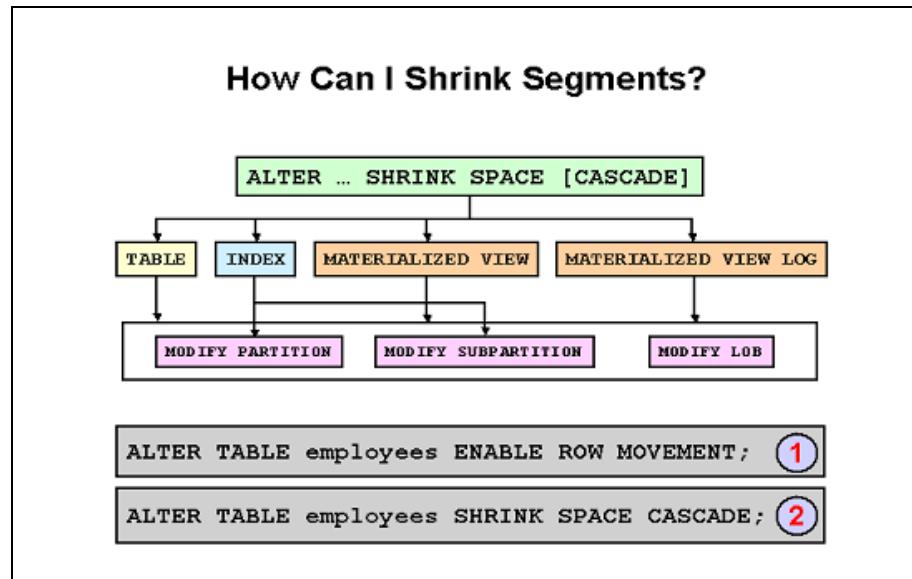


Fig 9: How Can I Shrink Segments?

As shown above, the alter table/index/... syntax is modified to contain the additional SHRINK SPACE [CASCADE] [COMPACT]. When the alter command is invoked without COMPACT, the default behavior of the Shrink command is to both perform the compaction and release the excess space back to the tablespace. When the command is invoked with the COMPACT option, only the compaction phase is performed. At the end of this phase the data has been moved to the beginning section of the segment, but the excess space has not been deallocated. The state of the the segment is remembered persistently. When an ALTER TABLE/INDEX/... SHRINK SPACE follows this operation, the database detects the fact that space has already been compacted and skips this phase (with the exception of concurrent DML activity that may fragment the segment after the compaction). Also, if the user interrupts the COMPACT operation, whatever compaction has been performed so far is saved in the database. The next invocation of COMPACT will pick up from where it left off. Thus Segment Shrink is a *restartable* operation.

Enterprise Manager fully supports the shrink functionality. Using EM, one can shrink a segment either as a response to Segment Advisor recommendation or directly by selecting a table or index.

The screenshot shows the Oracle Enterprise Manager interface for managing tables. The user is connected to the 'sh5d5.us.oracle.com' database. In the 'Tables' section, a table named 'CAL_MONTH_SALES_MV' from the 'SH' schema is selected. A context menu is open over this table, with the 'Shrink Segment' option highlighted.

| Select | Schema | Table Name | Tablespace | Partition Type | Partitions | Actions |
|----------------------------------|--------|----------------------|------------|----------------|------------|--|
| <input checked="" type="radio"/> | SH | CAL_MONTH_SALES_MV | EXAMPLE | | 0 | Edit View Delete Actions |
| <input type="radio"/> | SH | CHANNELS | EXAMPLE | | 0 | Create Like Create Index Create Synonym Create Trigger Gather Statistics Generate DDL Grant Privileges Reorganize Segment Advisor Shrink Segment Show Dependencies |
| <input type="radio"/> | SH | COSTS | | RANGE | 28 | |
| <input type="radio"/> | SH | COUNTRIES | EXAMPLE | | 0 | |
| <input type="radio"/> | SH | CUSTOMERS | EXAMPLE | | 0 | |
| <input type="radio"/> | SH | FWEEK_PSCAT_SALES_MV | EXAMPLE | | 0 | |
| <input type="radio"/> | SH | MVIEW\$_EXCEPTIONS | EXAMPLE | | 0 | |
| <input type="radio"/> | SH | ODM_BUILD | EXAMPLE | | 0 | |
| <input type="radio"/> | SH | ODM_TEST | EXAMPLE | | 0 | |
| <input type="radio"/> | SH | PRODUCTS | EXAMPLE | | 0 | |
| <input type="radio"/> | SH | PROMOTIONS | EXAMPLE | | 0 | |

Fig 10: Initiating a Segment Shrink using Enterprise Manager

UNDO MANAGEMENT

A lot of new enhancements to undo management make it easier to use than ever. The enhancements are based on a self-learning system that can automatically size an undo tablespace, dynamically tune undo retention, and provide fast ramp-up during sudden bursts of activity. This section covers key features in this area – Auto-tuning of undo retention, Auto-sizing Undo Tablespace, Undo Advisor, Proactive Undo Tablespace Monitoring and Alerts, Fast Ramp-up and Guaranteed Undo Retention

Auto-sizing Undo Tablespace

DBAs are often confronted with the task of selecting the *correct size* for a given tablespace. The Oracle Database 10g brings a welcome relief by introducing a number of easy to use capacity planning tools. As described above, the Create Table Estimator provided estimates on the sizing requirements of application data. The Oracle Database 10g provides similar help in sizing Undo tablespaces. There are two ways of doing it. You could either size a system using the Undo Advisor. It provides guidance on sizing undo tablespaces for the desired undo retention. Or you could leave the system to automatically size the undo tablespace by simply running the workload.

In the Oracle Database 10g, a DBA can provide one or more *autoextensible* files to the undo tablespace and the database server will automatically grow the files to the size required by the application. While this ability to auto extend a file has

been available for a while now, in Oracle Database 10g, *a self-learning system* is built into undo management. When you run your application against the Oracle Database 10g, the system automatically *learns* how long the application requires the undo data to be retained, either for read consistency or flashback usage. This, in turn, is used to control how big the undo tablespace should grow. This makes the Automatic Undo Management functionality fully self-contained – as long as sufficient disk space is provided to the tablespace, the self learning system automatically manages the undo tablespace.

The self-learning system also reacts to changing application needs – it is *constantly learning* the application running on the database. For example, if during daytime, a payroll system is running short transactions that last less than 10 minutes, the undo tablespace will grow to accommodate for the OLTP activity. Then, during off peak hours, the OLTP activity dies down and long running queries are run for payroll reporting purposes. This requires `undo_retention` of a high value. The self-learning system automatically detects the change in the DML activity and the change in the duration of the application queries and adapts the undo tablespace to this new environment. Note that in the absence of the self-learning system, if `undo_retention` was set to 10 minutes, the report generation queries will fail. If `undo_retention` is set to be longer, then that will cause excessive undo to be retained (and require more disk space) during the OLTP workload. The self-learning system automatically adjusts `undo_retention` to changing workload to adjust the resource configuration and usage for efficient execution of the application.

Auto Tuning of Undo Retention

As described above, in Oracle Database 10g, `undo_retention` need not be manually specified any longer. The self- learning component of undo management learns the required undo retention from the workload running on the database and dynamically adapts the `undo_retention` to that workload. If the DBA does specify a value for `undo_retention`, then the self-learning component still tunes the `undo_retention`, but does not allow it to fall below the user supplied `undo_retention`. `Undo_retention` may be supplied, for example, when using Flashback features. Because the system does not tune `undo_retention` below the user supplied value, Flashback is available at least for that amount of time.

There are two components to auto tuning `undo_retention`:

- Proactive Tuning. Undo retention is tuned to remain ahead of the longest running query in the system. Query duration information is collected every 30 seconds.
- Reactive Tuning. Undo retention is gradually lowered when the system is running under space pressure – that is the system does not have sufficient disk space available (if autoextensible files are present, they

cannot be extended further) to accommodate the undo_retention for the given undo generation rate of the system.

Auto tuning undo_retention allows for efficient use of the space in the undo_tablespace by dynamically adapting to the workload. In addition, it simplifies the job of a database administrator significantly by freeing them for having to determine and set the proper value of undo retention manually.

Undo Advisor

The Oracle Database 10g includes the Undo advisor to help DBAs optimize the undo configuration. It can provide answer to commonly asked questions such as:

- Is the size of my undo tablespace adequate for the application workload?
- If I wish to set the undo retention to a given value, how big an undo tablespace do I need?
- Given the size of my current undo tablespace, what is the maximum undo retention database can support?

At the heart of the undo advisor is the continuous statistics gathering in the database. The Oracle Database 10g continuously gathers statistics on the health of the undo tablespace and stores it in the workload repository. These historical statistics are then used to compute the space requirement if the undo retention were to be changed or vice versa.

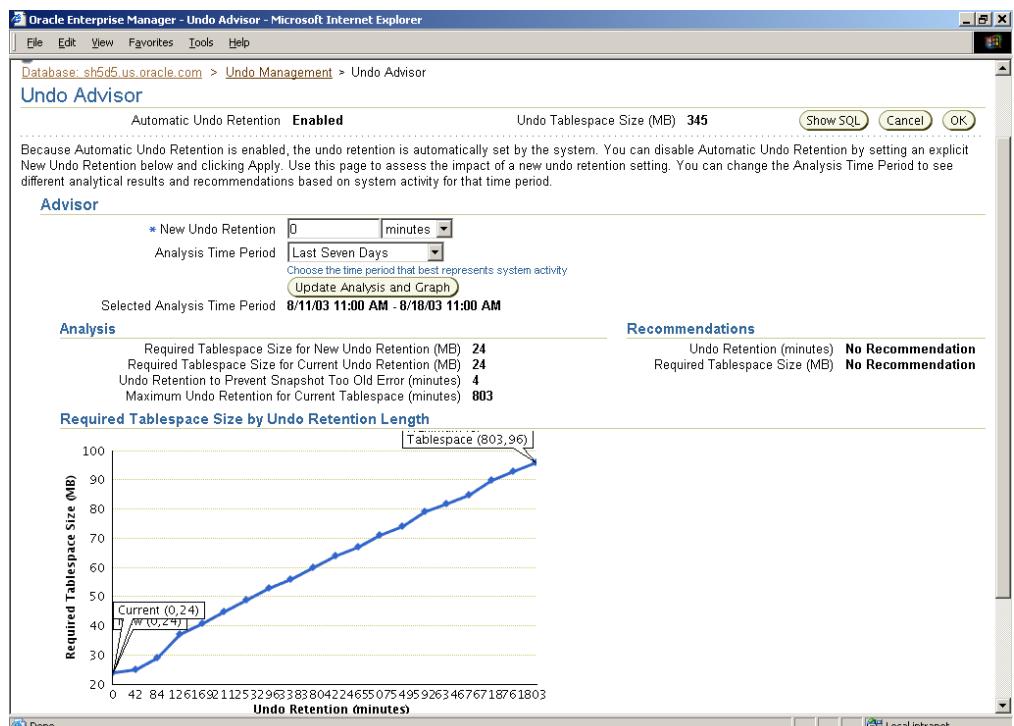


Fig 11: The Undo Advisor

Proactive Undo Tablespace Monitoring

As described in the Tablespace Alert section, the database actively monitors space usage in the undo tablespace. If the space usage crosses the warning or critical thresholds, alerts are raised or cleared as the case may be.

Undo tablespaces may also trigger the *Long Query Warning Alert* – this alert alerts the DBA of long running queries that may fail because the system may be running low on free space in the undo tablespace.

Fast Ramp-up

The self-learning system that controls auto undo tuning preserves it's learning across instance shutdown. Through the workload repository, it keeps track of the appropriate resource configuration that the system needs for optimal performance. So instead of having to tune the system up and then quiesce down during low activity, the self-learning system keeps the system in a state of readiness. This enables the system to provide high performance and low response time when the application load suddenly spikes up.

Guaranteed Availability of Undo

While using Automatic Undo Management, transactions take precedence over queries in space consumption by default. For example, booking a sales order is more important than querying the current sales results. There are times when the success of queries is critical to the administrator as well as the application. For example, generating reports for printing paychecks may be a critical operation towards the end of a pay period in a HRMS system. Also, the DBA may choose the use the Flashback features as an insurance to recover from application/human errors. And this insurance may be critical part of the failure recovery strategy. In such an environment, the DBA may wish to *guarantee* the availability of undo in the undo tablespace for the specified undo_retention duration.

The Oracle Database 10g allows the administrator to choose between the default behavior of compromising the undo retention during space pressure or enforcing a given undo retention. It may be noted that if the system does run into space pressure, DML activity may get out of space errors in order to satisfy the guarantee commitment. Of course, Undo Tablespace Alerts will fire long before the system runs into space pressure.

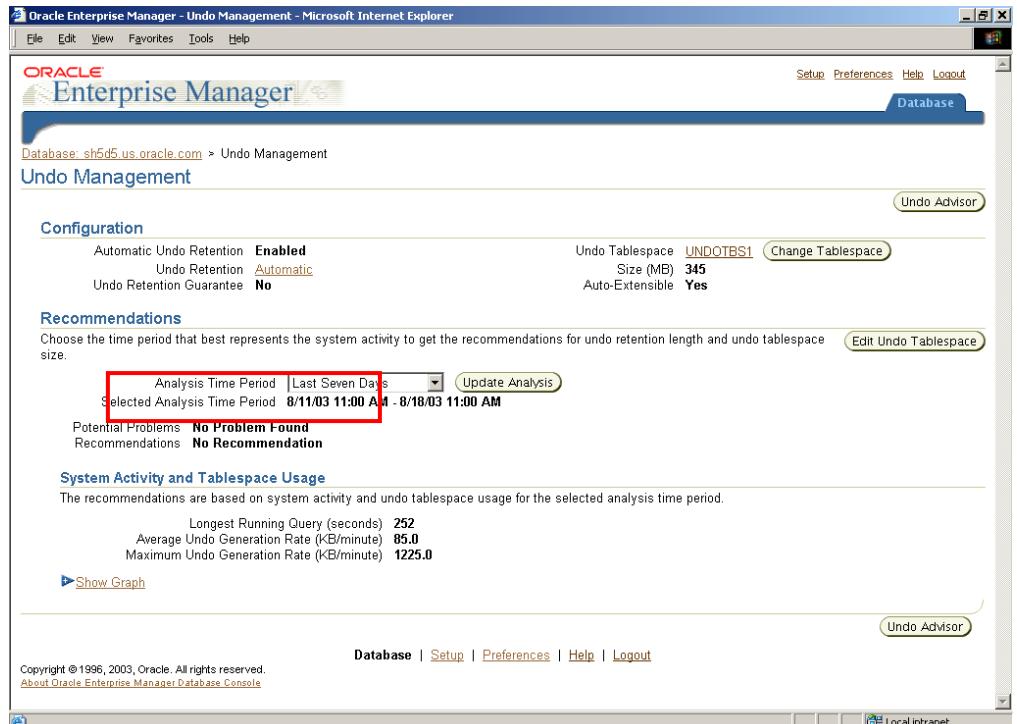


Fig 12: Guaranteeing Undo Retention

CONCLUSION

Space Management functionality in the Oracle Database 10g is a key contributor in making it a self-managing database. It is designed to have self-awareness and self-learning built into the core database engine. That, combined with intuitive, integrated, uniform and easy to use Enterprise Manager user interface, makes the task of space management in the Oracle Database 10g, significantly easier and more powerful than ever.



White Paper Title: The Self-Managing Database: Proactive Space & Schema Object Management

Nov 2003

Author:Amit Ganesh

Contributing Authors: Sushil Kumar

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

www.oracle.com

**Oracle Corporation provides the software
that powers the internet.**

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2002 Oracle Corporation
All rights reserved.