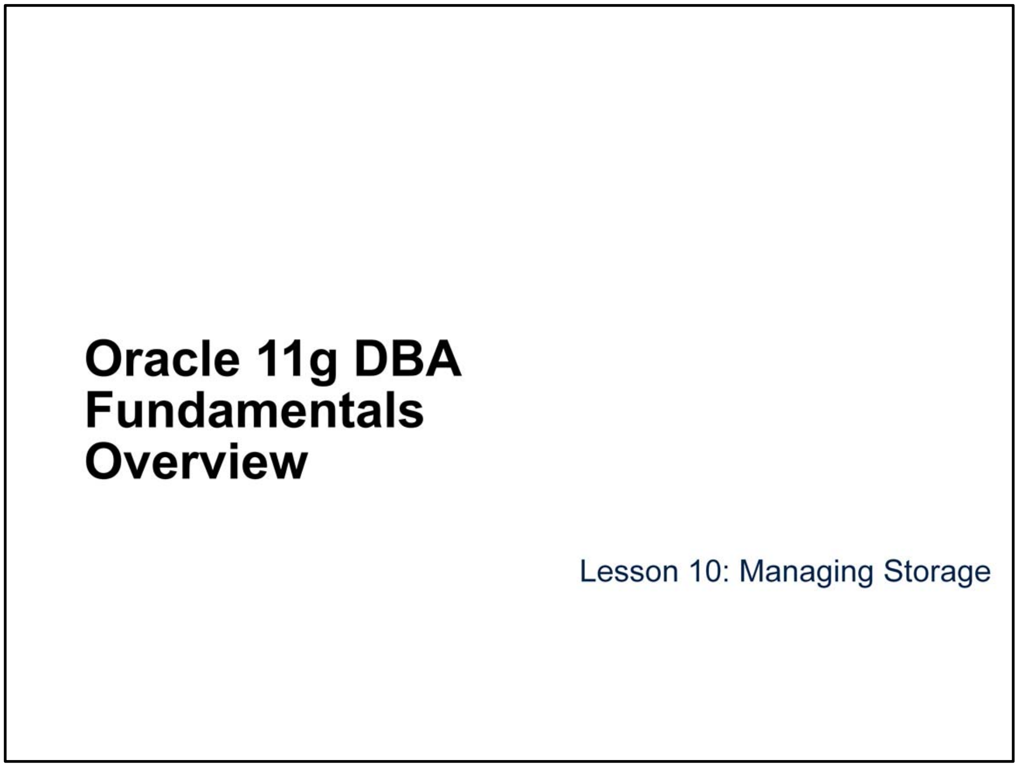
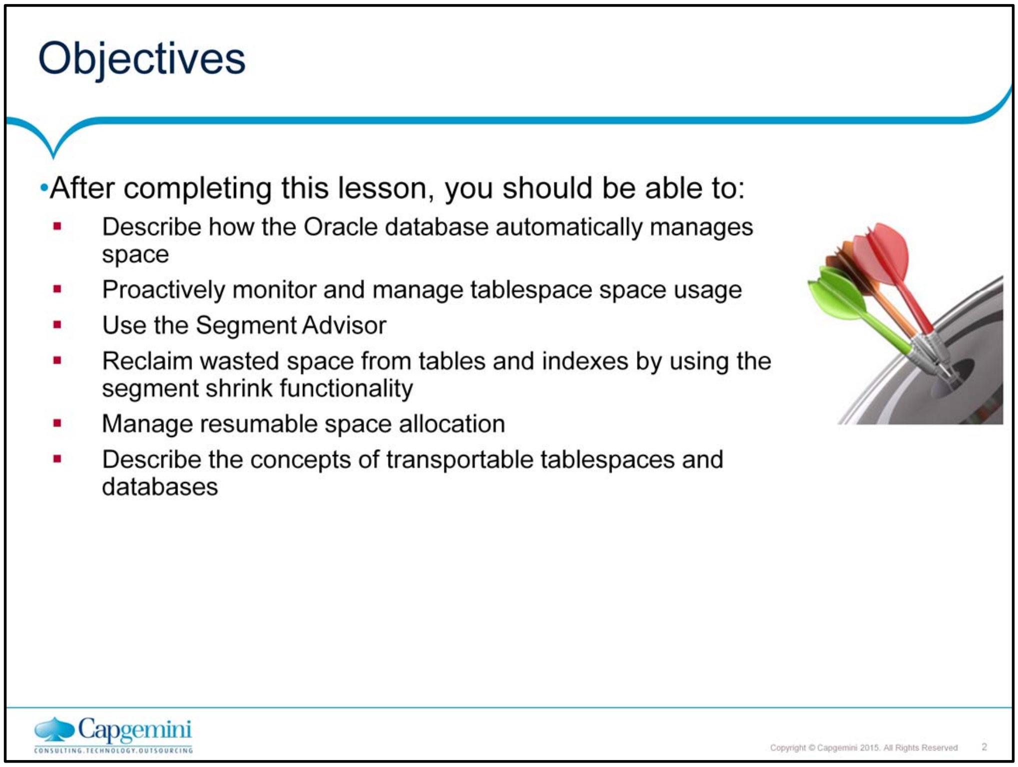


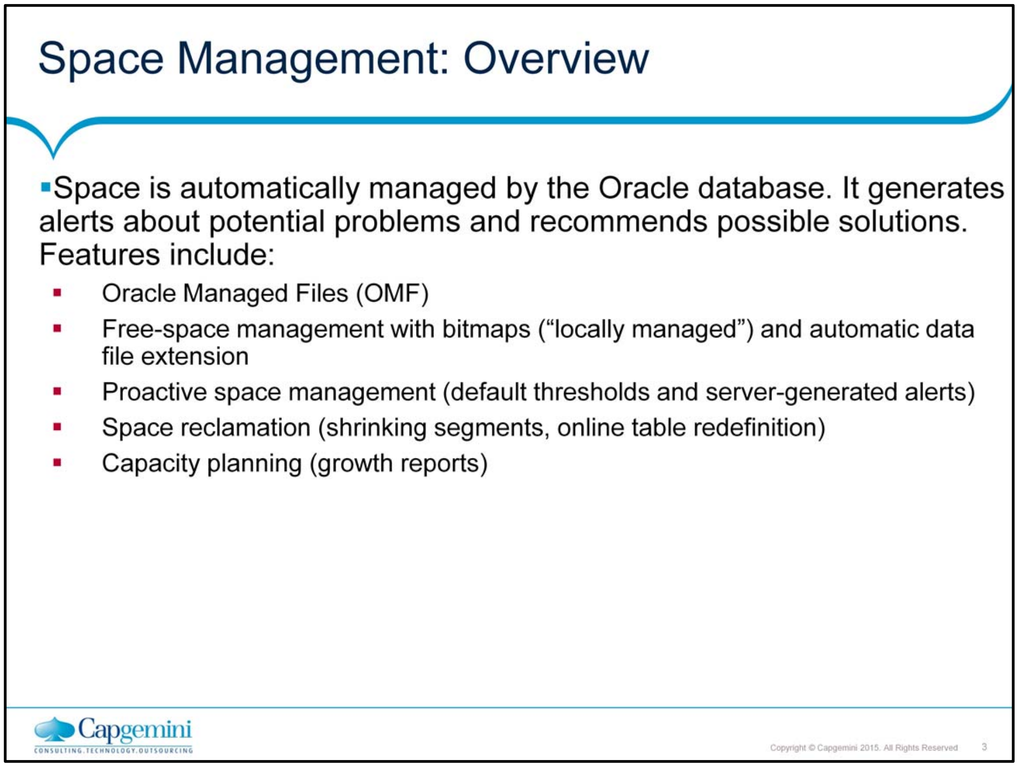
Oracle 11g DBA Fundamentals Overview

Lesson 10: Managing Storage



Objectives

- After completing this lesson, you should be able to:
  - Describe how the Oracle database automatically manages space
  - Proactively monitor and manage tablespace space usage
  - Use the Segment Advisor
  - Reclaim wasted space from tables and indexes by using the segment shrink functionality
  - Manage resumable space allocation
  - Describe the concepts of transportable tablespaces and databases



Space Management: Overview

- Space is automatically managed by the Oracle database. It generates alerts about potential problems and recommends possible solutions. Features include:
  - Oracle Managed Files (OMF)
  - Free-space management with bitmaps (“locally managed”) and automatic data file extension
  - Proactive space management (default thresholds and server-generated alerts)
  - Space reclamation (shrinking segments, online table redefinition)
  - Capacity planning (growth reports)

Space Management: Overview

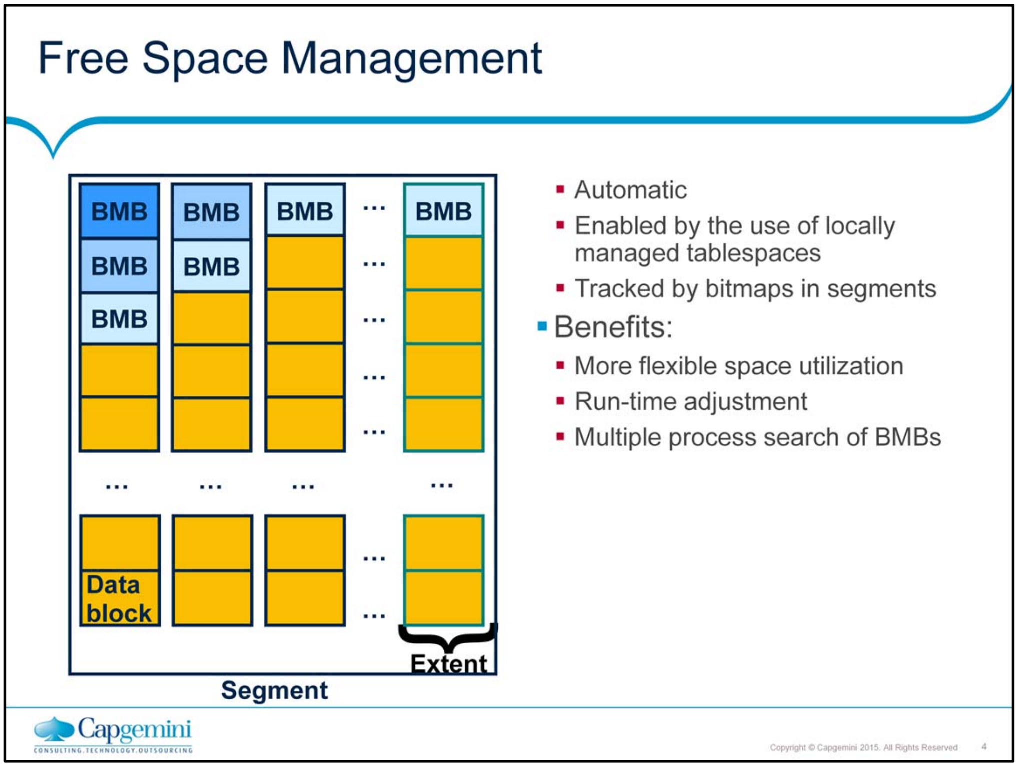
With Oracle Managed Files (OMF), you can specify operations in terms of database objects rather than file names. For more details, see the lesson titled “Introduction.”

The Oracle database manages free space within a table with bitmaps. This is known as a “locally managed” tablespace. (Dictionary-managed tablespaces are supported only for backward compatibility.) The bitmapped implementation eliminates much space-related tuning of tables, while providing improved performance during peak loads. Additionally, the Oracle database provides automatic extension of data files, so the files can grow automatically based on the amount of data in the files.

When you create a database, proactive space monitoring is enabled by default. (This causes no performance impact.) The Oracle database monitors space utilization during normal space allocation and deallocation operations and alerts you if the free space availability falls below the predefined thresholds (which you can override). Advisors and wizards assist you with space reclamation.

For capacity planning, the Oracle database provides space estimates based on table structure and number of rows and a growth trend report based on historical space utilization stored in the Automatic Workload Repository (AWR).

The Oracle Database 11g: Administration Workshop I course provides an introduction to space and storage concepts, related utilities, and DBA tasks. Through this or other means, you should be familiar with the basic concepts and storage features.



Free Space Management

- Automatic
- Enabled by the use of locally managed tablespaces
- Tracked by bitmaps in segments

Benefits:

- More flexible space utilization
- Run-time adjustment
- Multiple process search of BMBs

Copyright © Capgemini 2015. All Rights Reserved 4

Free Space Management

Free space can be managed automatically inside database segments. The in-segment free or used space is tracked with bitmaps. You specify Automatic Segment Space Management, when you create a locally managed tablespace. Your specification then applies to all segments subsequently created in this tablespace.

Automatic space management segments have a set of bitmap blocks (BMBs) describing the space utilization of the data blocks in that segment. BMBs are organized in a tree hierarchy. The root level of the hierarchy, which contains the references to all intermediate BMBs, is stored in the segment header. The leaves of this hierarchy represent the space information for a set of contiguous data blocks that belong to the segment. The maximum number of levels inside this hierarchy is three.

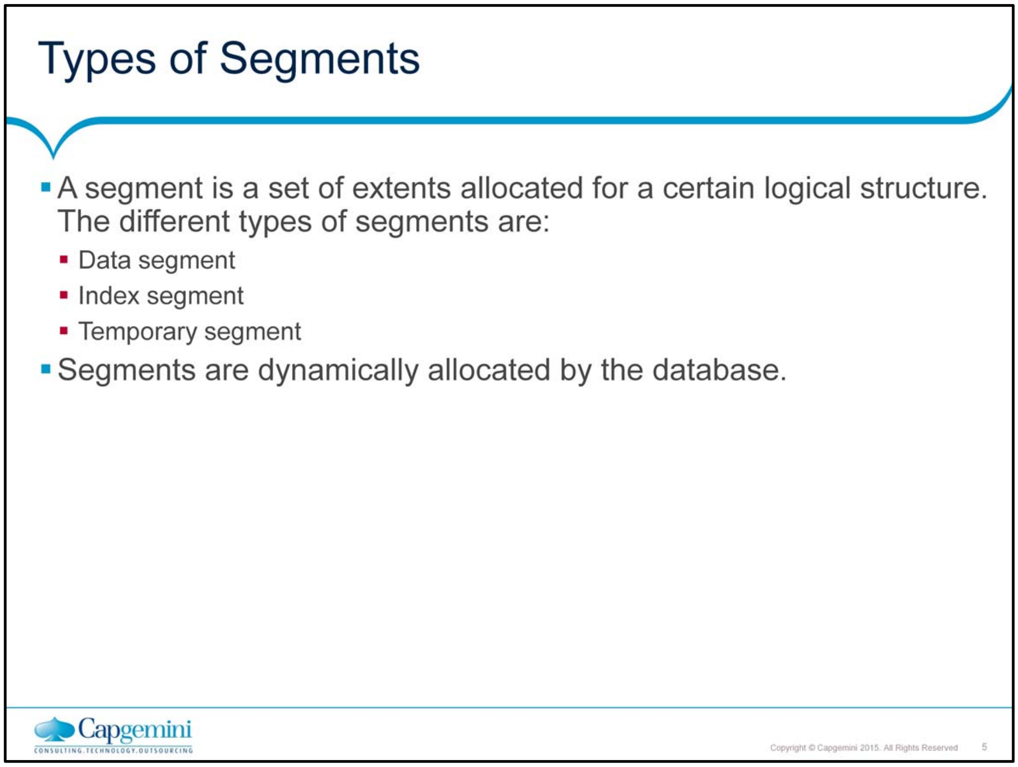
Benefits of using automatic space management (compared to manual space management, which uses “freelist” data structures and is synonymous with “dictionary-managed” tablespaces):

Better space utilization, especially for the objects with highly varying row sizes

Better run-time adjustment to variations in concurrent access

Better multi-instance behavior in terms of performance or space utilization

Therefore, less work for you, the DBA.



Types of Segments

- A segment is a set of extents allocated for a certain logical structure. The different types of segments are:
  - Data segment
  - Index segment
  - Temporary segment
- Segments are dynamically allocated by the database.

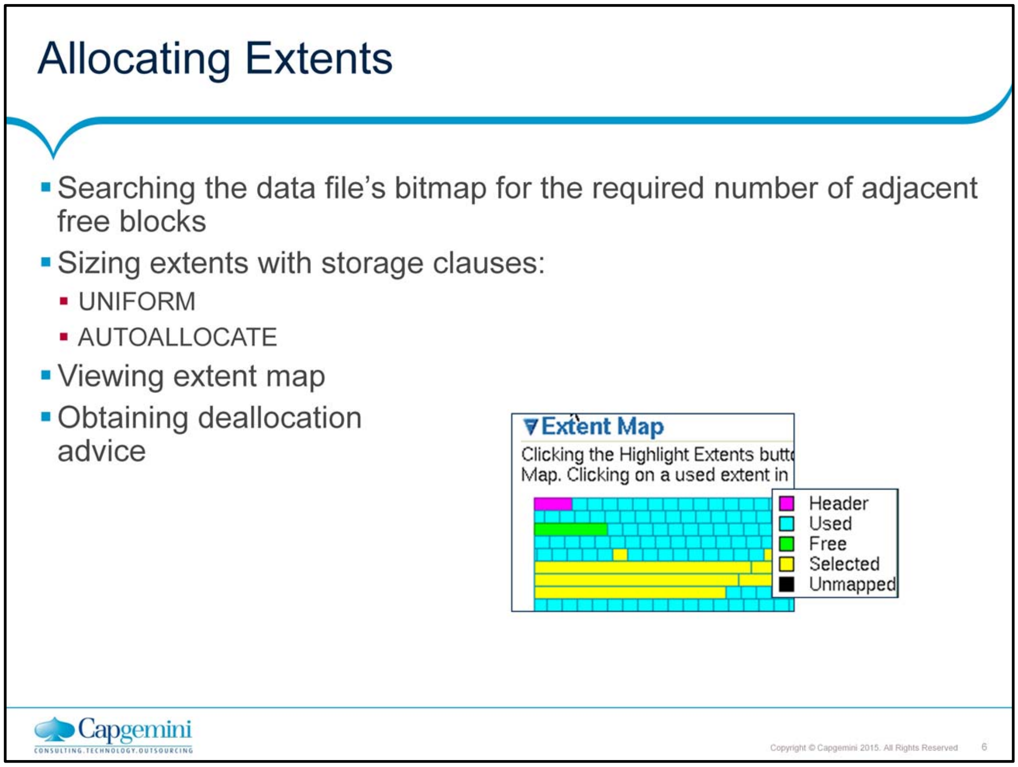
Types of Segments

Data segment: Each nonclustered table has a data segment. All table data is stored in the extents of the data segment. For a partitioned table, each partition has a data segment. Each cluster has a data segment. The data of every table in the cluster is stored in the cluster’s data segment.

Index segment: Each index has an index segment that stores all of its data. For a partitioned index, each partition has an index segment.

Temporary segment: A temporary segment is created by the Oracle database when a SQL statement needs a temporary database area to complete execution. When the statement finishes execution, the extents in the temporary segment are returned to the system for future use.

The Oracle database dynamically allocates space when the existing extents of a segment become full. Because extents are allocated as needed, the extents of a segment may or may not be contiguous on disk.



Allocating Extents

- Searching the data file's bitmap for the required number of adjacent free blocks
- Sizing extents with storage clauses:
  - UNIFORM
  - AUTOALLOCATE
- Viewing extent map
- Obtaining deallocation advice

▼Extent Map

Clicking the Highlight Extents butt
Map. Clicking on a used extent in

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

6

Allocating Extents

With locally managed tablespaces, the Oracle database looks for free space to allocate to a new extent by first determining a candidate data file in the tablespace and then searching the data file's bitmap for the required number of adjacent free blocks. If that data file does not have enough adjacent free space, then the Oracle database looks in another data file.

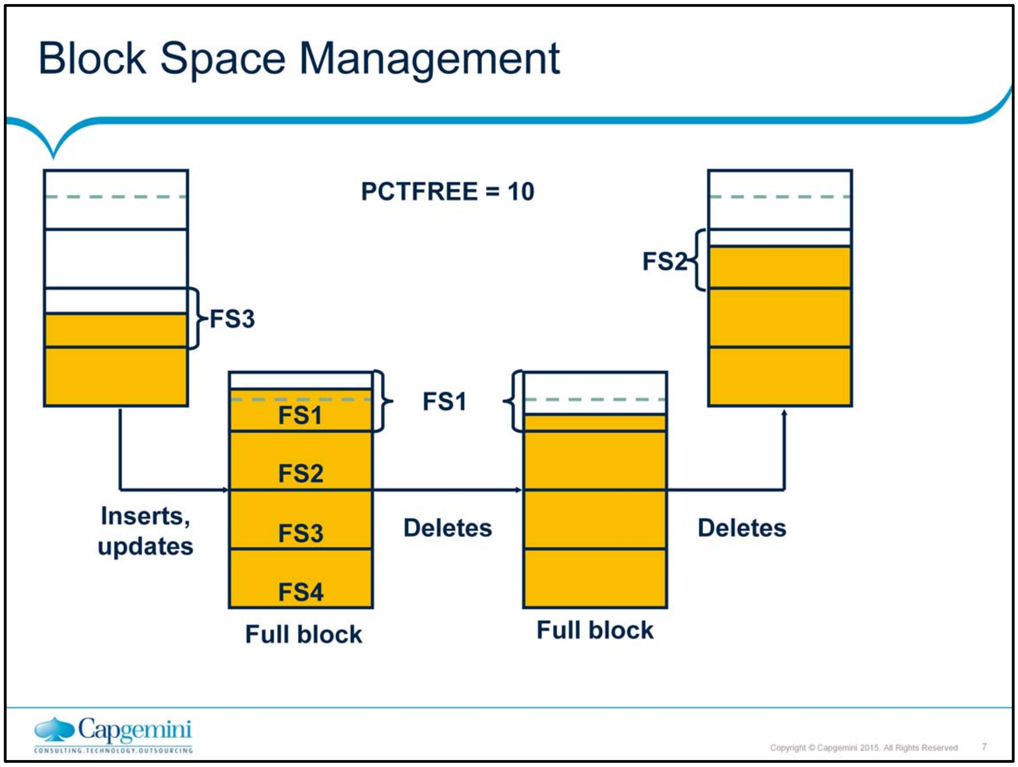
Two clauses affect the sizing of extents:

With the UNIFORM clause, the database creates all extents of a uniform size that you specified (or a default size) for any objects created in the tablespace.

With the AUTOALLOCATE clause, the database determines the extent-sizing policy for the tablespace.

To view the extent map in Enterprise Manager, choose Administration > Tablespaces > View Tablespace > Show Tablespace Contents.

The Oracle database provides a Segment Advisor that helps you determine whether an object has space available for reclamation on the basis of the level of space fragmentation within the object.



Block Space Management

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

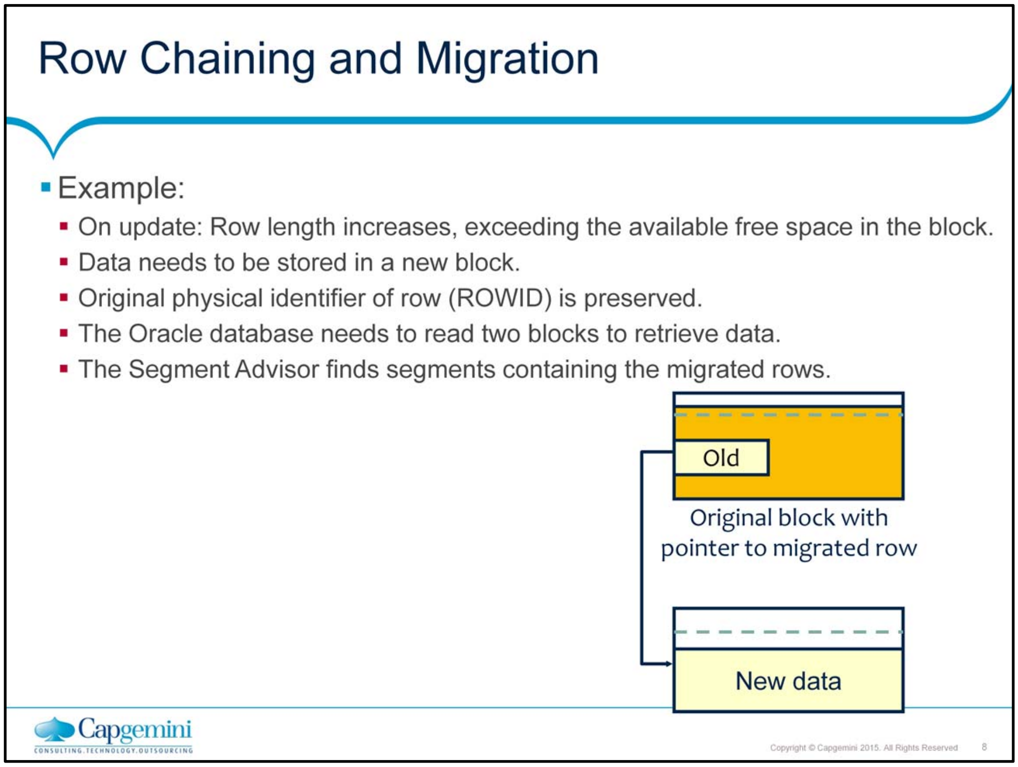
7

Block Space Management

Space management involves the management of free space at the block level. With Automatic Segment Space Management, each block is divided into four sections, named FS1 (between 0 and 25% of free space), FS2 (25% to50% free), FS3 (50% to 75% free), and FS4 (75% to 100% free). Depending on the level of free space in the block, its status is automatically updated. That way, depending on the length of an inserted row, you can tell whether a particular block can be used to satisfy an insert operation. Note that a “full” status means that a block is no longer available for inserts.

In the slide example, the block on the left is an FS3 block because it has between 50% and 75% free space. After some insert and update statements, PCTFREE is reached (the dashed line) and it is no longer possible to insert new rows in that block. The block is now considered as a “full” or FS1 block. The block is considered for insertion again, as soon as its free space level drops below the next section. In the above case, it gets status FS2 as soon as the free space is more than 25%.

Note: Large object (LOB) data types (BLOB, CLOB, NCLOB, and BFILE) do not use the PCTFREE storage parameter. For more information, see the Oracle Database Application Developer’s Guide - Large Objects 11g Release 2 (10.2).



Row Chaining and Migration

- Example:
  - On update: Row length increases, exceeding the available free space in the block.
  - Data needs to be stored in a new block.
  - Original physical identifier of row (ROWID) is preserved.
  - The Oracle database needs to read two blocks to retrieve data.
  - The Segment Advisor finds segments containing the migrated rows.

Old

Original block with pointer to migrated row

New data

Row Chaining and Migrating

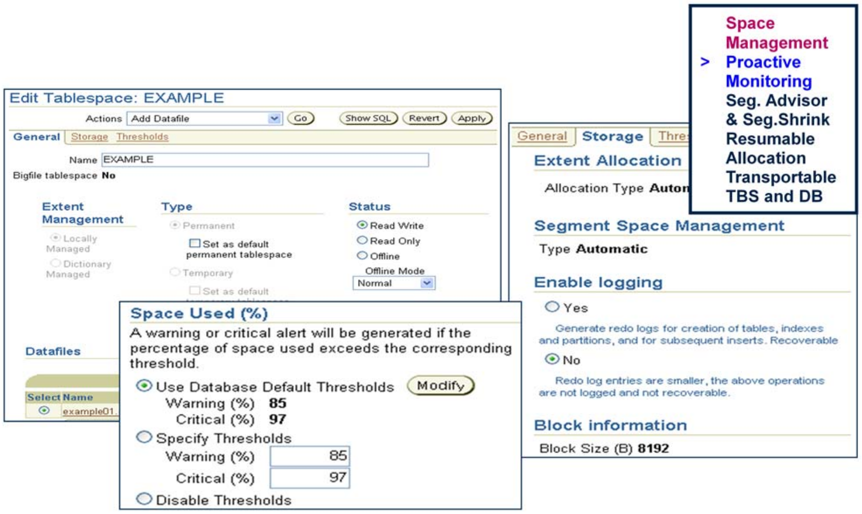
In two circumstances, the data for a row in a table may be too large to fit into a single data block. In the first case, the row is too large to fit into one data block when it is first inserted. In this case, the Oracle database stores the data for the row in a chain of data blocks (one or more) reserved for that segment. Row chaining most often occurs with large rows, such as rows that contain a column of data type LONG or LONG RAW. Row chaining in these cases is unavoidable.

However, in the second case, a row that originally fit into one data block is updated, so that the overall row length increases, and the block's free space is already completely filled. In this case, the Oracle database migrates the data for the entire row to a new data block, assuming that the entire row can fit in a new block. The database preserves the original row piece of a migrated row to point to the new block containing the migrated row. The ROWID of a migrated row does not change.

When a row is chained or migrated, I/O performance associated with this row decreases because the Oracle database must scan more than one data block to retrieve the information for the row.

The Segment Advisor finds the segments containing migrated rows that result from an UPDATE.

Proactive Tablespace Monitoring



Space Management
> Proactive Monitoring
Seg. Advisor & Seg.Shrink
Resumable Allocation
Transportable TBS and DB

Edit Tablespace: EXAMPLE

Actions: Add Datafile (Go) (Show SQL) (Revert) (Apply)

General | Storage | Thresholds

Name: EXAMPLE
Bigfile tablespace: No

Extent Management: ● Locally Managed ○ Dictionary Managed

Type: ● Permanent ☐ Set as default permanent tablespace ○ Temporary ☐ Set as default

Status: ● Read Write ○ Read Only ○ Offline
Offline Mode: Normal

Datafiles

Select Name
● example01

General | Storage | Thresholds

Extent Allocation
Allocation Type: Autom

Segment Space Management
Type: Automatic

Enable logging
○ Yes
Generate redo logs for creation of tables, indexes and partitions, and for subsequent inserts. Recoverable
● No
Redo log entries are smaller, the above operations are not logged and not recoverable.

Block information
Block Size (B): 8192

Space Used (%)
A warning or critical alert will be generated if the percentage of space used exceeds the corresponding threshold.

● Use Database Default Thresholds (Modify)
Warning (%) 85
Critical (%) 97
○ Specify Thresholds
Warning (%) 85
Critical (%) 97
○ Disable Thresholds

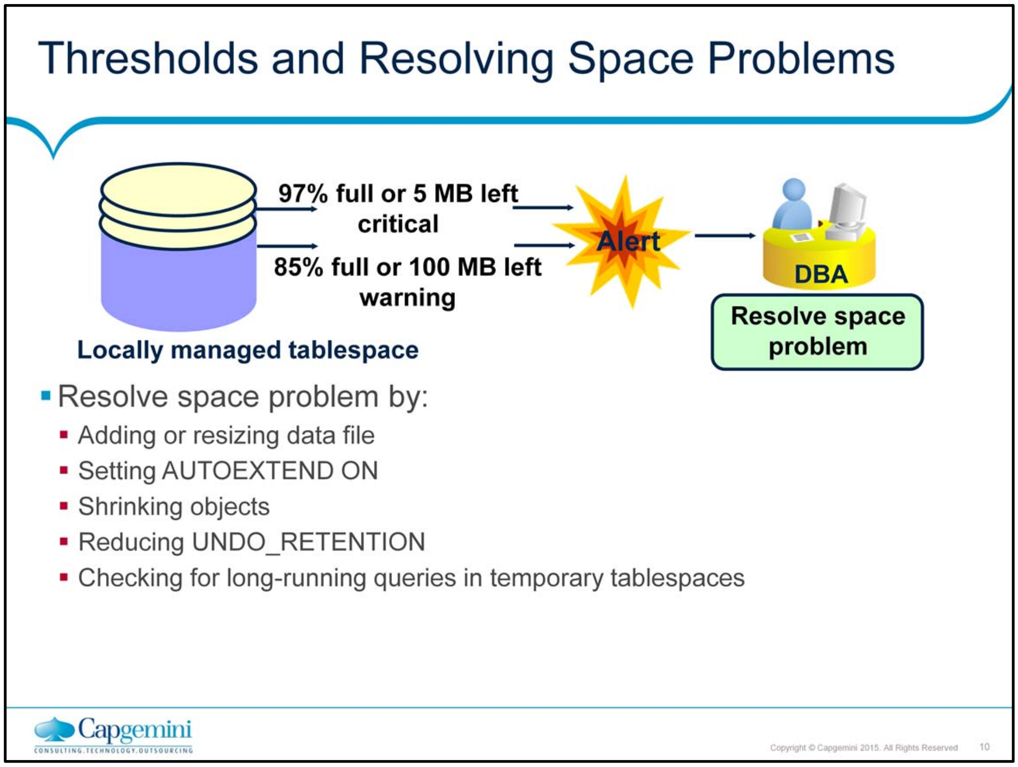
Proactive Tablespace Monitoring

Tablespace disk space usage is proactively managed by the database in the following ways:

Through the use of database alerts, you are informed when a tablespace runs low on available disk space as well as when particular segments are running out of space. You can then provide the tablespace with more disk space, thus avoiding out-of-space conditions.

Information gathered is stored in the Automatic Workload Repository (AWR) and is used to perform growth trend analysis and capacity planning of the database.

To view and modify tablespace information in Enterprise Manager, select Administration > Tablespaces. Select the tablespace of your choice and click the Edit button.



Thresholds and Resolving Space Problems

97% full or 5 MB left
critical

85% full or 100 MB left
warning

Alert

DBA

Resolve space problem

Locally managed tablespace

- Resolve space problem by:
  - Adding or resizing data file
  - Setting AUTOEXTEND ON
  - Shrinking objects
  - Reducing UNDO_RETENTION
  - Checking for long-running queries in temporary tablespaces

Thresholds and Resolving Space Problems

Tablespace thresholds are defined either as full or as available space in the tablespace. Critical and warning thresholds are the two thresholds that apply to a tablespace. The DBMS_SERVER_ALERT package contains procedures to set and get the threshold values. When the tablespace limits are reached, an appropriate alert is raised. The threshold is expressed in terms of a percentage of the tablespace size or in remaining bytes free. It is calculated in memory. You can have both a percentage and a byte-based threshold defined for a tablespace. Either or both of them may generate an alert.

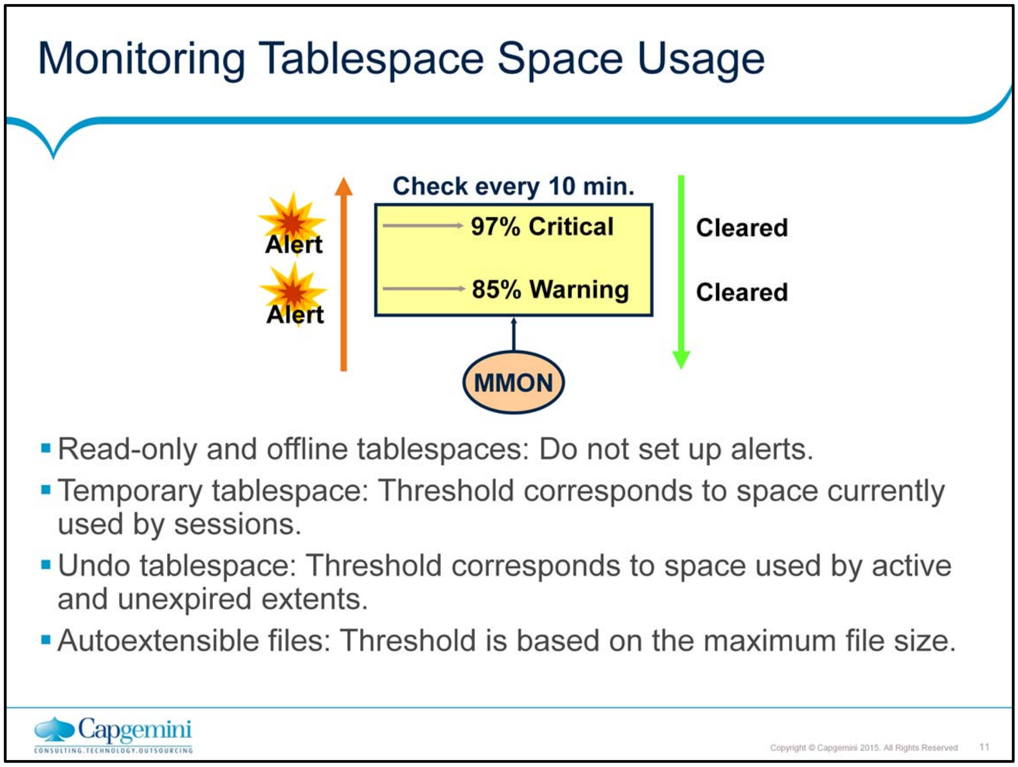
The ideal setting for the warning threshold trigger value results in an alert that is early enough to ensure that there is enough time to resolve the problem before it becomes critical, but late enough so that you are not bothered when space is not a problem.

The alert indicates that the problem can be resolved by doing one or more of the following:

Adding more space to the tablespace by adding a file or resizing existing files, or making an existing file autoextendable

Freeing up space on disks that contain any autoextendable files

Shrinking sparse objects in the tablespace



Monitoring Tablespace Space Usage

- Read-only and offline tablespaces: Do not set up alerts.
- Temporary tablespace: Threshold corresponds to space currently used by sessions.
- Undo tablespace: Threshold corresponds to space used by active and unexpired extents.
- Autoextensible files: Threshold is based on the maximum file size.

11

Monitoring Tablespace Space Usage

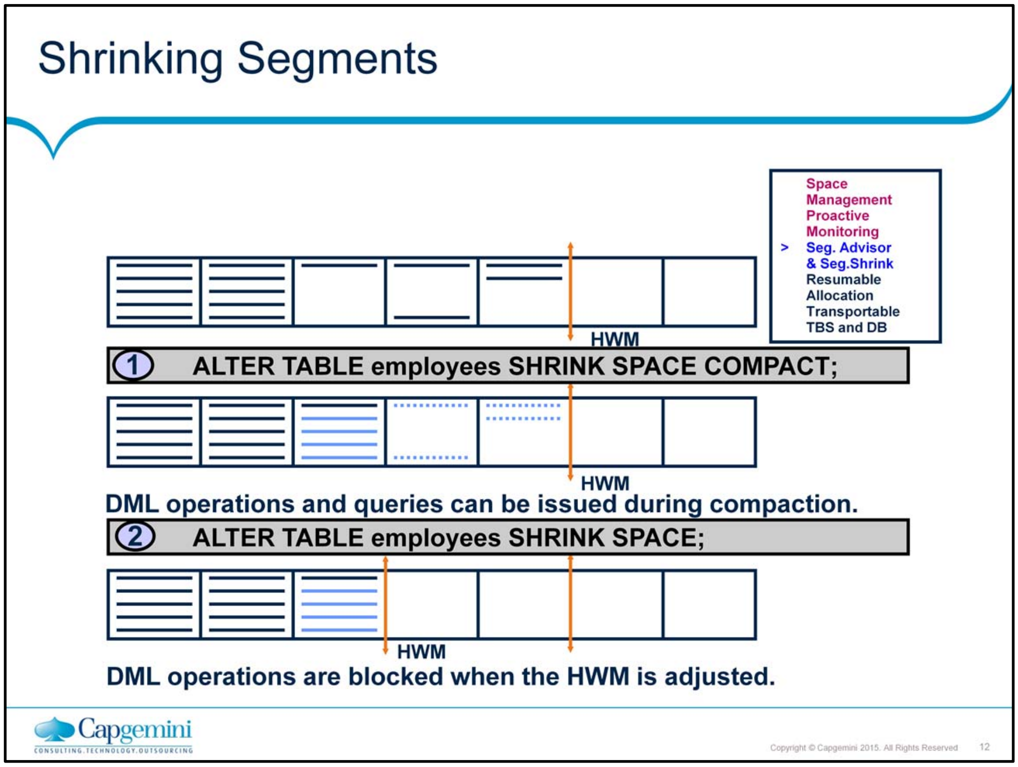
The database tracks space utilization while performing regular space management activities. This information is aggregated every 10 minutes by the MMON process. An alert is triggered when the threshold for a tablespace has been reached or cleared.

Alerts should not be flagged on tablespaces that are in read-only mode, or tablespaces that were taken offline, because there is not much to do for them.

In temporary tablespaces, the threshold value has to be defined as a limit on the used space in the tablespace.

For undo tablespaces, an extent is reusable if it does not contain active or unexpired undo. For the computation of threshold violation, the sum of active and unexpired extents is considered as used space.

For tablespaces with autoextensible files, the thresholds are computed according to the maximum file size you specified, or the maximum OS file size.



Shrinking Segments

Space
Management
Proactive
Monitoring
> Seg. Advisor
& Seg.Shrink
Resumable
Allocation
Transportable
TBS and DB

1 ALTER TABLE employees SHRINK SPACE COMPACT;

DML operations and queries can be issued during compaction.

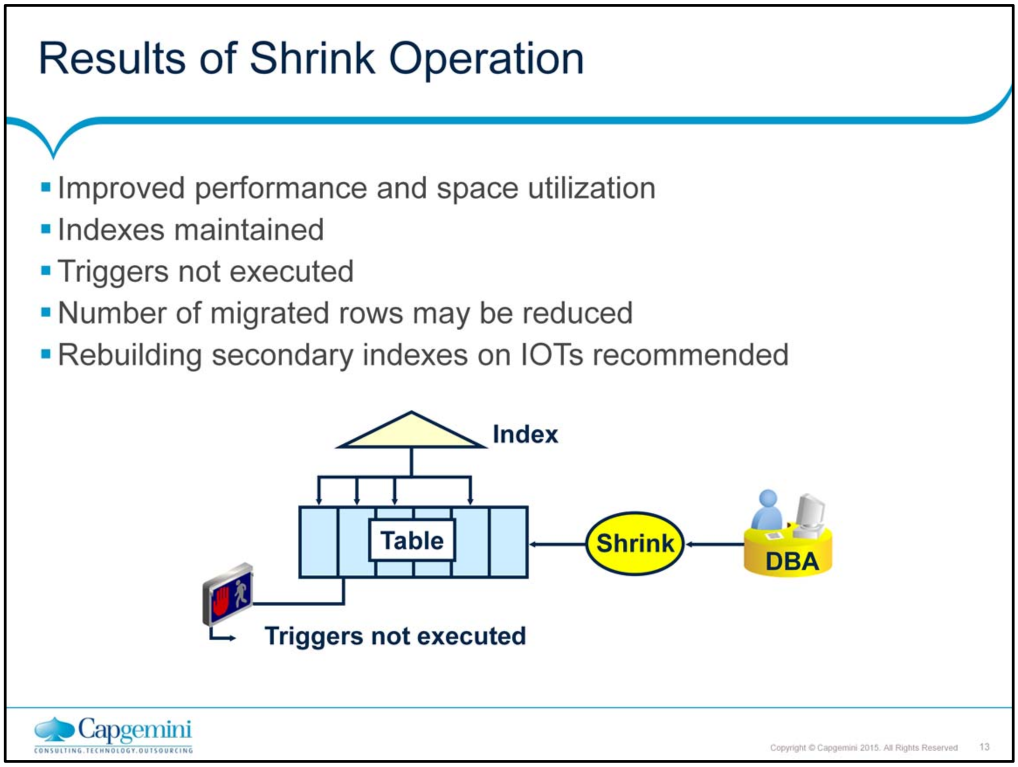
2 ALTER TABLE employees SHRINK SPACE;

DML operations are blocked when the HWM is adjusted.

Shrinking Segments

The diagram in the slide describes the two phases of a table shrink operation. The first phase does the compaction. During this phase, rows are moved to the left part of the segment as much as possible. Internally, rows are moved by packets to avoid locking issues. After the rows have been moved, the second phase of the shrink operation is started. During this phase, the high-water mark (HWM) is adjusted and the unused space is released.

The COMPACT clause is useful if you have long-running queries that might span the shrink operation and attempt to read from blocks that have been reclaimed. When you specify the SHRINK SPACE COMPACT clause, the progress of the shrink operation is saved in the bitmap blocks of the corresponding segment. This means that the next time a shrink operation is executed on the same segment, the Oracle database remembers what has been done already. You can then reissue the SHRINK SPACE clause without the COMPACT clause during off-peak hours to complete the second phase.



Results of Shrink Operation

- Improved performance and space utilization
- Indexes maintained
- Triggers not executed
- Number of migrated rows may be reduced
- Rebuilding secondary indexes on IOTs recommended

Index
Table
Shrink
DBA
Triggers not executed

Results of Shrink Operation

Shrinking a sparsely populated segment improves the performance of scan and DML operations on that segment. This is because there are fewer blocks to look at after the segment has been shrunk. This is especially true for:

Full table scans (fewer and denser blocks)

Better index access (fewer I/Os on range ROWID scans due to a more compact tree)

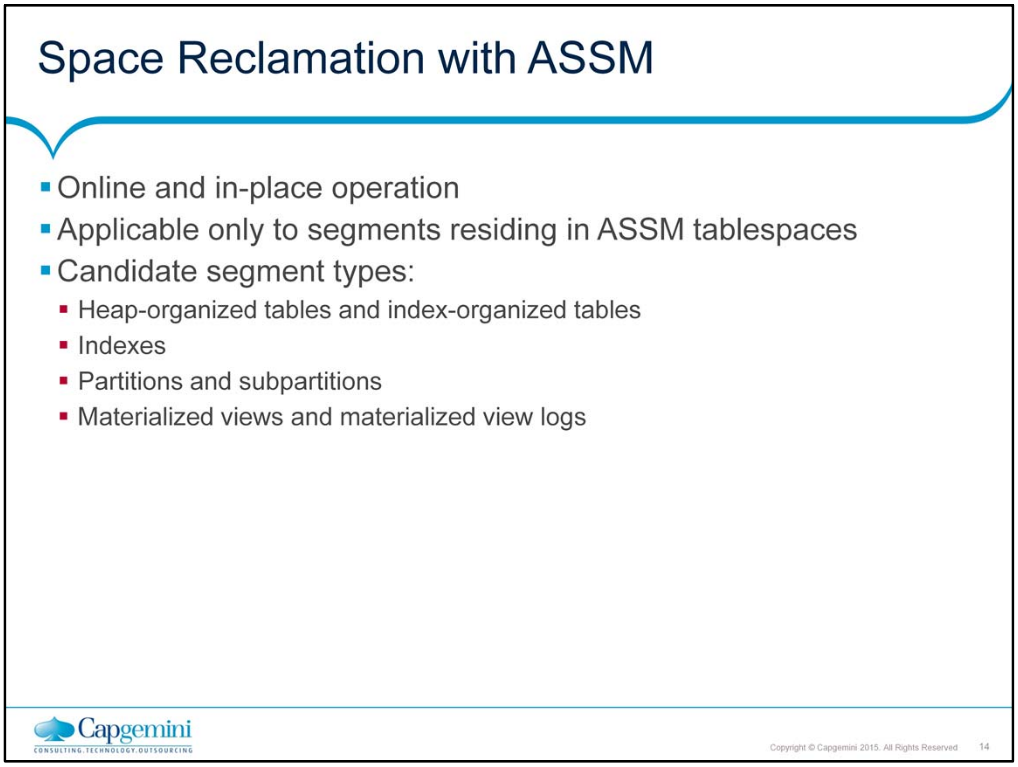
Also, by shrinking sparsely populated segments, you enhance the efficiency of space utilization inside your database because more free space is made available for objects in need.

Index dependency is taken care of during the segment shrink operation. The indexes are in a usable state after shrinking the corresponding table. Therefore, no further maintenance is needed.

The actual shrink operation is handled internally as an INSERT/DELETE operation. However, any DML triggers are not executed because the data itself is not changed.

As a result of a segment shrink operation, it is possible that the number of migrated rows is reduced. However, you should not always depend on reducing the number of migrated rows after a segment has been shrunk. This is because a segment shrink operation may not touch all the blocks in the segment. Therefore, it is not guaranteed that all the migrated rows are handled.

Note: It is recommended to rebuild secondary indexes on an index-organized table (IOT) after a shrink operation.



Space Reclamation with ASSM

- Online and in-place operation
- Applicable only to segments residing in ASSM tablespaces
- Candidate segment types:
  - Heap-organized tables and index-organized tables
  - Indexes
  - Partitions and subpartitions
  - Materialized views and materialized view logs

Space Reclamation with ASSM

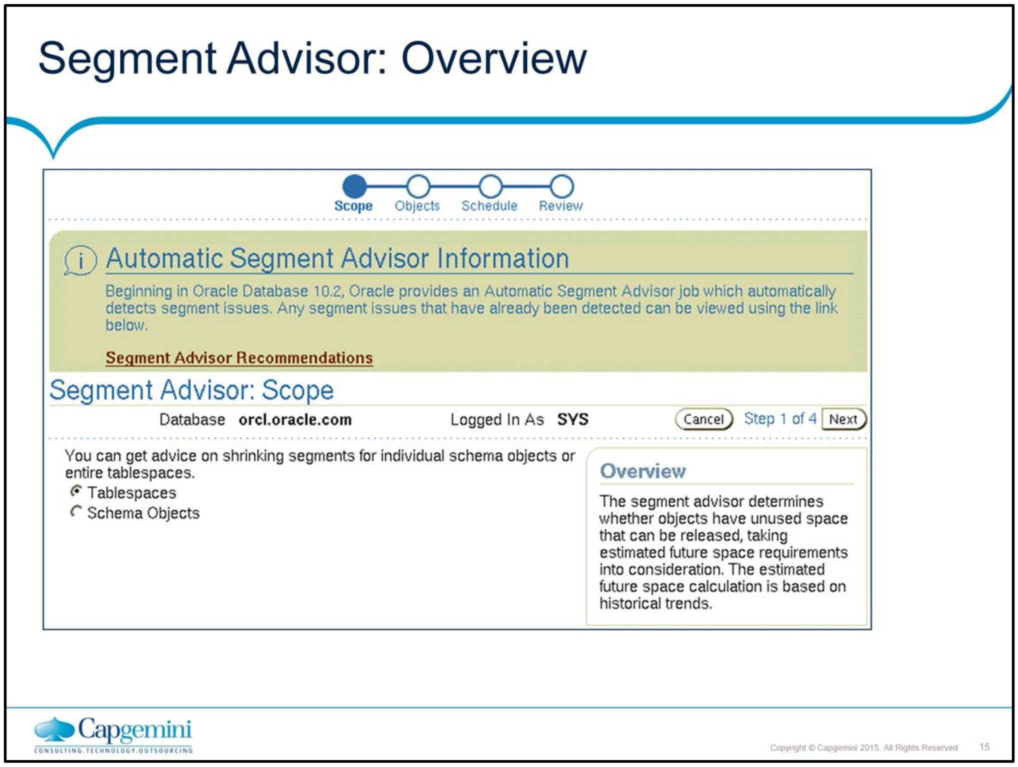
A shrink operation is an online and in-place operation because it does not need extra database space to be executed.

You cannot execute a shrink operation on segments managed by free lists. Segments in automatic segment-space managed tablespaces can be shrunk. However, the following objects stored in ASSM tablespaces cannot be shrunk:

Tables in clusters
Tables with LONG columns
Tables with on-commit materialized views
Tables with ROWID-based materialized views
IOT mapping tables
Tables with function-based indexes

ROW MOVEMENT must be enabled for heap-organized segments.

Note: Automatic Segment Space Management (ASSM) is the default type of segment space management for all new, permanent, locally managed tablespaces in Oracle Database 11g Release 2.



Segment Advisor: Overview

Automatic Segment Advisor Information

Beginning in Oracle Database 10.2, Oracle provides an Automatic Segment Advisor job which automatically detects segment issues. Any segment issues that have already been detected can be viewed using the link below.

Segment Advisor Recommendations

Segment Advisor: Scope

Database **orcl.oracle.com** Logged In As **SYS**

You can get advice on shrinking segments for individual schema objects or entire tablespaces.

● Tablespaces

○ Schema Objects

Overview

The segment advisor determines whether objects have unused space that can be released, taking estimated future space requirements into consideration. The estimated future space calculation is based on historical trends.

15

Segment Advisor: Overview

The Segment Advisor determines whether an object is a good candidate for a shrink operation. The advisor also finds the segments containing migrated rows that result from an UPDATE. (Beginning with Oracle Database 10.2, the Segment Advisor jobs are automatically run for you.) The advisor makes recommendations based on the amount of unused space that can be released, and takes into consideration estimated future space requirements by using criteria from the gathered information about segment growth trends.

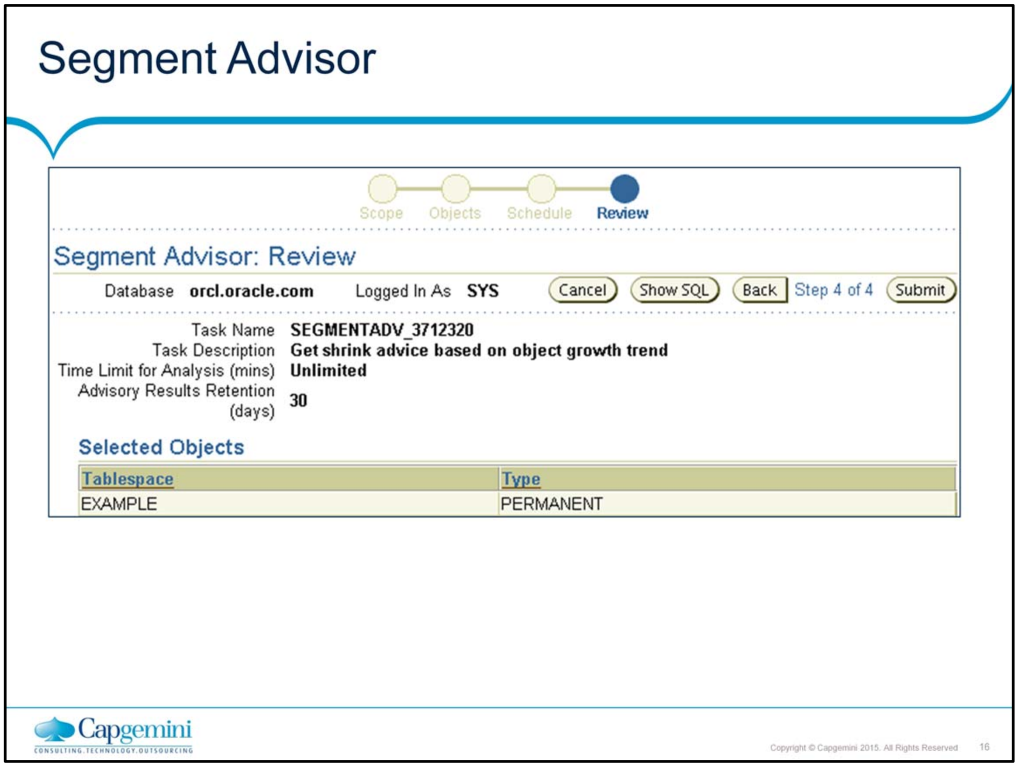
After the recommendations are made, you can choose to implement the recommendations. The shrink advisor can be invoked at the segment or tablespace level.

The EM Database Control Console is the interface to the Segment Advisor. You can access the Segment Advisor from several places within EM:

- Advisor Central page
- Tablespaces page
- Schema object pages

The Database Control Console provides the option to select various inputs and schedule a job that calls the Segment Advisor to get shrink advice. The Segment Advisor wizard can be invoked with no context, in the context of a tablespace, or in the context of a schema object.

The Segment Advisor makes recommendation on the basis of sampled analysis, historical information, and future growth trends.



Segment Advisor

Segment Advisor: Review

Database **orcl.oracle.com** Logged In As **SYS**

Task Name **SEGMENTADV_3712320**
Task Description **Get shrink advice based on object growth trend**
Time Limit for Analysis (mins) **Unlimited**
Advisory Results Retention (days) **30**

Selected Objects

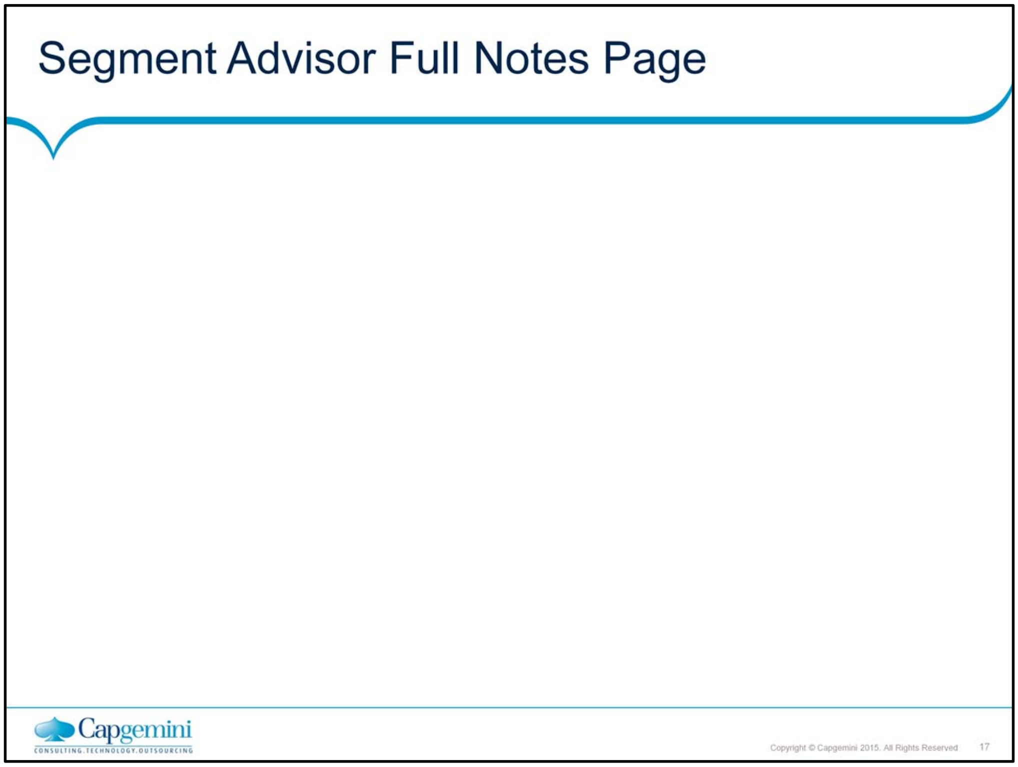
Tablespace	Type
EXAMPLE	PERMANENT

Segment Advisor

From the Administration page, select Tablespaces in the Storage section. On the Tablespaces page, select the tablespace on which you want to perform the shrink analysis, and then select Run Segment Advisor in the Actions drop-down list. Click Go to open the Segment Advisor initial page. You must choose “comprehensive” or “limited’ analysis mode. In comprehensive mode, the analysis is longer because the advisor is sampling the segments to identify the right targets.

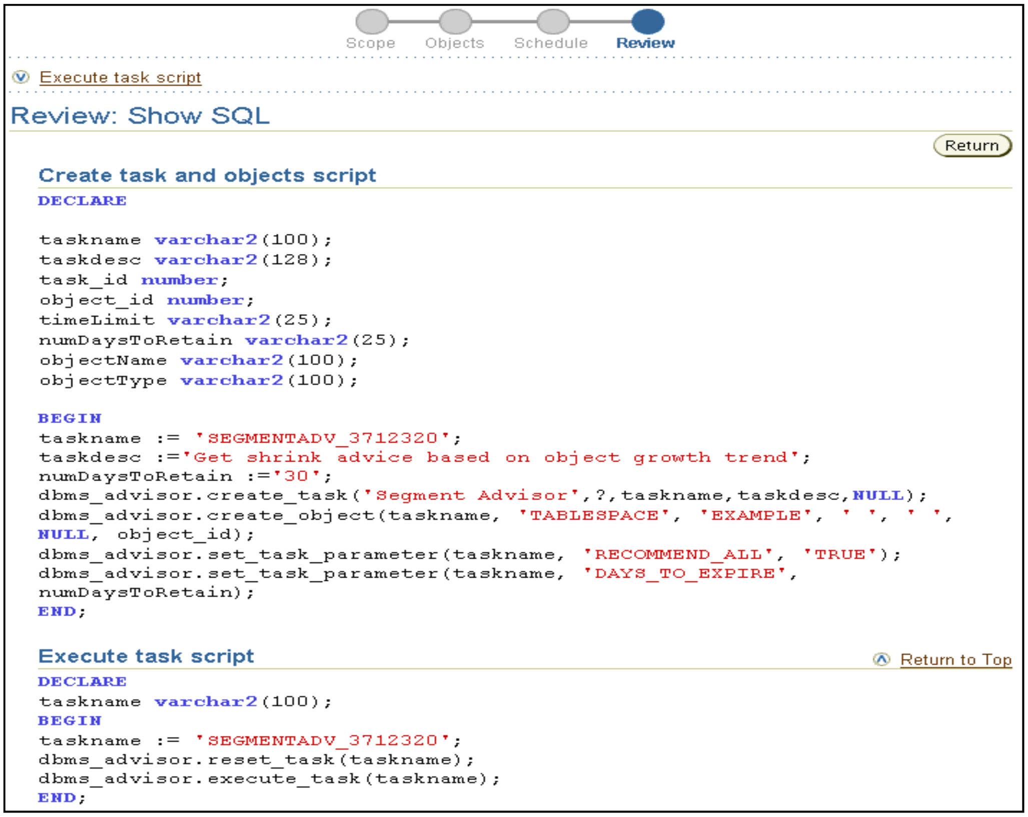
Keep clicking Continue to answer the various questions of the advisor. You end up on the Segment Advisor: Review page, where you can review the details of your analysis. The Segment Advisor analysis is run as a scheduled job, so you can review the scheduled task from the Advisor Central page. When completed, you can review the advisor’s recommendations.

Note: In the Segment Advisor, you can specify the duration of the analysis. This enables you to limit the time the advisor takes to produce recommendations. Generally speaking, a longer analysis period produces more comprehensive results. The results are stored in the AWR and can be viewed later. Use the “Number of days to retain” option to instruct the Oracle database how long these results should be preserved before being purged from the AWR.



Segment Advisor Full Notes Page

Segment Advisor (continued)



Execute task script

Review: Show SQL

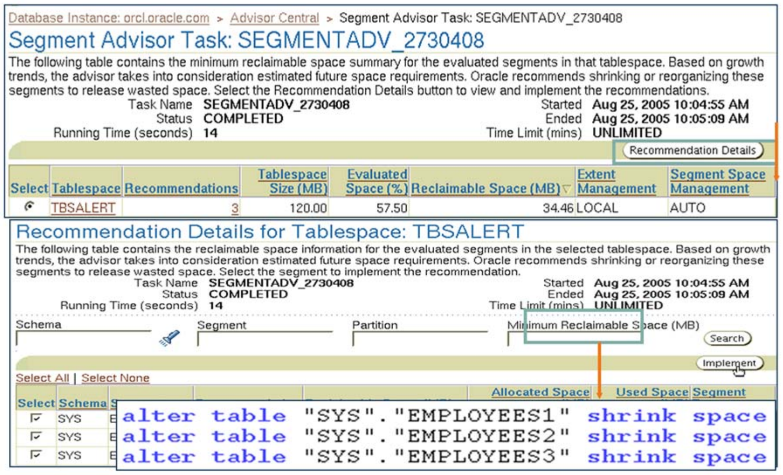
Create task and objects script

```
DECLARE

taskname varchar2(100);
taskdesc varchar2(128);
task_id number;
object_id number;
timeLimit varchar2(25);
numDaysToRetain varchar2(25);
objectName varchar2(100);
objectType varchar2(100);

BEGIN
taskname := 'SEGMENTADV_3712320';
taskdesc :='Get shrink advice based on object growth trend';
numDaysToRetain :='30';
dbms_advisor.create_task('Segment Advisor',?,taskname,taskdesc,NULL);
dbms_advisor.create_object(taskname, 'TABLESPACE', 'EXAMPLE', ' ', ' ',
NULL, object_id);
dbms_advisor.set_task_parameter(taskname, 'RECOMMEND_ALL', 'TRUE');
dbms_advisor.set_task_parameter(taskname, 'DAYS_TO_EXPIRE',
numDaysToRetain);
END;
```

Execute task script

```
DECLARE
taskname varchar2(100);
BEGIN
taskname := 'SEGMENTADV_3712320';
dbms_advisor.reset_task(taskname);
dbms_advisor.execute_task(taskname);
END;
```

Implementing Recommendations



Database Instance: orcl.oracle.com > Advisor Central > Segment Advisor Task: SEGMENTADV_2730408

Segment Advisor Task: SEGMENTADV_2730408

The following table contains the minimum reclaimable space summary for the evaluated segments in that tablespace. Based on growth trends, the advisor takes into consideration estimated future space requirements. Oracle recommends shrinking or reorganizing these segments to release wasted space. Select the Recommendation Details button to view and implement the recommendations.

Task Name	SEGMENTADV_2730408	Started	Aug 25, 2005 10:04:55 AM
Status	COMPLETED	Ended	Aug 25, 2005 10:05:09 AM
Running Time (seconds)	14	Time Limit (mins)	UNLIMITED

Select	Tablespace	Recommendations	Tablespace Size (MB)	Evaluated Space (%)	Reclaimable Space (MB) ▽	Extent Management	Segment Space Management
●	TBSALERT	3	120.00	57.50	34.46	LOCAL	AUTO

Recommendation Details for Tablespace: TBSALERT

The following table contains the reclaimable space information for the evaluated segments in the selected tablespace. Based on growth trends, the advisor takes into consideration estimated future space requirements. Oracle recommends shrinking or reorganizing these segments to release wasted space. Select the segment to implement the recommendation.

Task Name	SEGMENTADV_2730408	Started	Aug 25, 2005 10:04:55 AM
Status	COMPLETED	Ended	Aug 25, 2005 10:05:09 AM
Running Time (seconds)	14	Time Limit (mins)	UNLIMITED

Schema ____ Segment ____ Partition ____ Minimum Reclaimable Space (MB) ____

Select	Schema	S	Allocated Space	Used Space	Segment
☑	SYS	E			
☑	SYS	E			
☑	SYS	E			

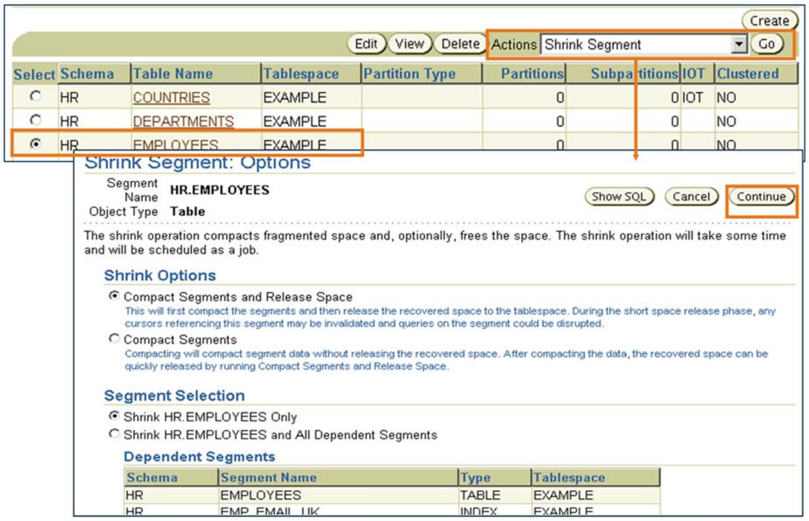
```
alter table "SYS"."EMPLOYEES1" shrink space
alter table "SYS"."EMPLOYEES2" shrink space
alter table "SYS"."EMPLOYEES3" shrink space
```

Implementing Recommendations

After the Segment Advisor completes its job, you can view the recommendation details and implement them directly.

Note: Before shrinking a heap-organized table, you must enable row movement on that table. You can do this with Database Control from the Options tab on the Edit Table page.

Database Control and Segment Shrink



Select	Schema	Table Name	Tablespace	Partition Type	Partitions	Subpartitions	IOT	Clustered
○	HR	COUNTRIES	EXAMPLE		0	0	IOT	NO
○	HR	DEPARTMENTS	EXAMPLE		0	0		NO
●	HR	EMPLOYEES	EXAMPLE		0	0		NO

Shrink Segment: Options

Segment Name **HR.EMPLOYEES**
Object Type **Table**

The shrink operation compacts fragmented space and, optionally, frees the space. The shrink operation will take some time and will be scheduled as a job.

Shrink Options

● Compact Segments and Release Space
This will first compact the segments and then release the recovered space to the tablespace. During the short space release phase, any cursors referencing this segment may be invalidated and queries on the segment could be disrupted.

○ Compact Segments
Compacting will compact segment data without releasing the recovered space. After compacting the data, the recovered space can be quickly released by running Compact Segments and Release Space.

Segment Selection

● Shrink HR.EMPLOYEES Only
○ Shrink HR.EMPLOYEES and All Dependent Segments

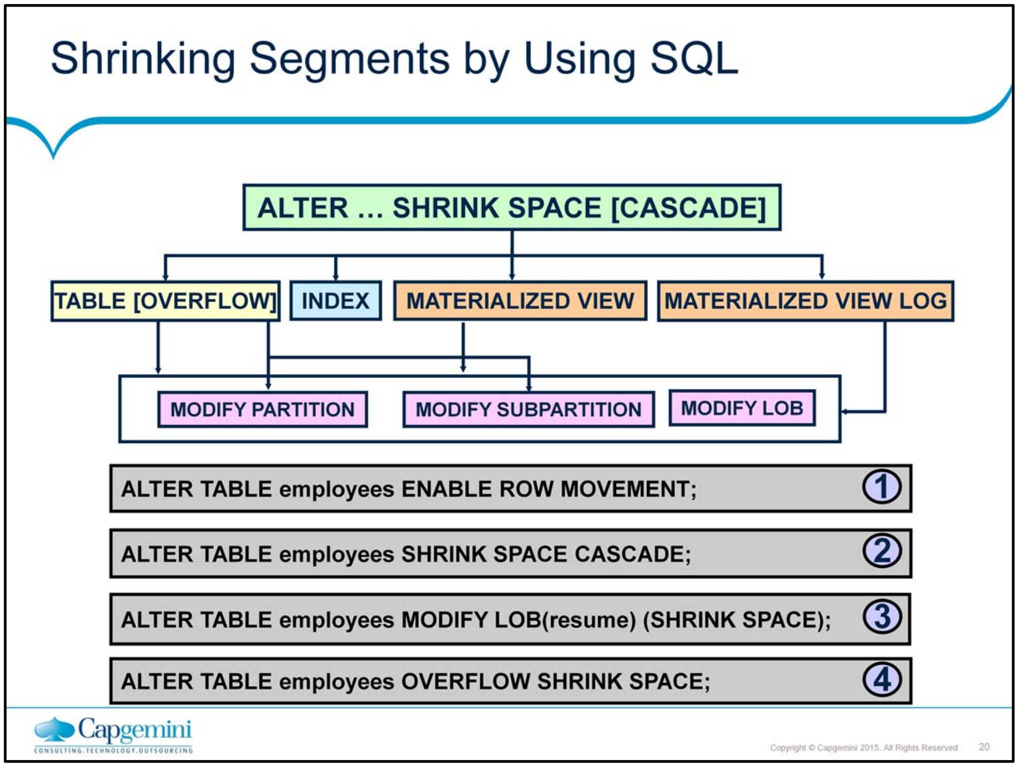
Dependent Segments

Schema	Segment Name	Type	Tablespace
HR	EMPLOYEES	TABLE	EXAMPLE
HR	EMP_EMAIL_UK	INDEX	EXAMPLE

Database Control and Segment Shrink

Alternatively (to implementing the Segment Advisor recommendations), you can shrink individual segments. For example, from the Database Control home page, click the Tables link in the Storage section. On the Tables page, select your table, and then select Shrink Segment in the Actions drop-down list. Then click the Go button. This brings you to the Shrink Segment page, where you can choose the dependent segments to shrink. You have the opportunity to compact only or to compact and release the space. You can also choose the CASCADE option.

When done, click the Continue link. This submits the shrink statements as a scheduled job.



Shrinking Segments by Using SQL

ALTER … SHRINK SPACE [CASCADE]

TABLE [OVERFLOW] | INDEX | MATERIALIZED VIEW | MATERIALIZED VIEW LOG

MODIFY PARTITION | MODIFY SUBPARTITION | MODIFY LOB

```
ALTER TABLE employees ENABLE ROW MOVEMENT;                   (1)
ALTER TABLE employees SHRINK SPACE CASCADE;                  (2)
ALTER TABLE employees MODIFY LOB(resume) (SHRINK SPACE);     (3)
ALTER TABLE employees OVERFLOW SHRINK SPACE;                  (4)
```

Shrinking Segments by Using SQL

Because a shrink operation may cause ROWIDs to change in heap-organized segments, you must enable row movement on the corresponding segment before executing a shrink operation on that segment. Row movement by default is disabled at segment level. To enable row movement, the ENABLE ROW MOVEMENT clause of the CREATE TABLE or ALTER TABLE command is used. This is illustrated in the first example in the slide.

Use the ALTER command to invoke segment shrink on an object. The object's type can be one of the following: table (heap- or index-organized), partition, subpartition, LOB (data and index segment), index, materialized view, or materialized view log.

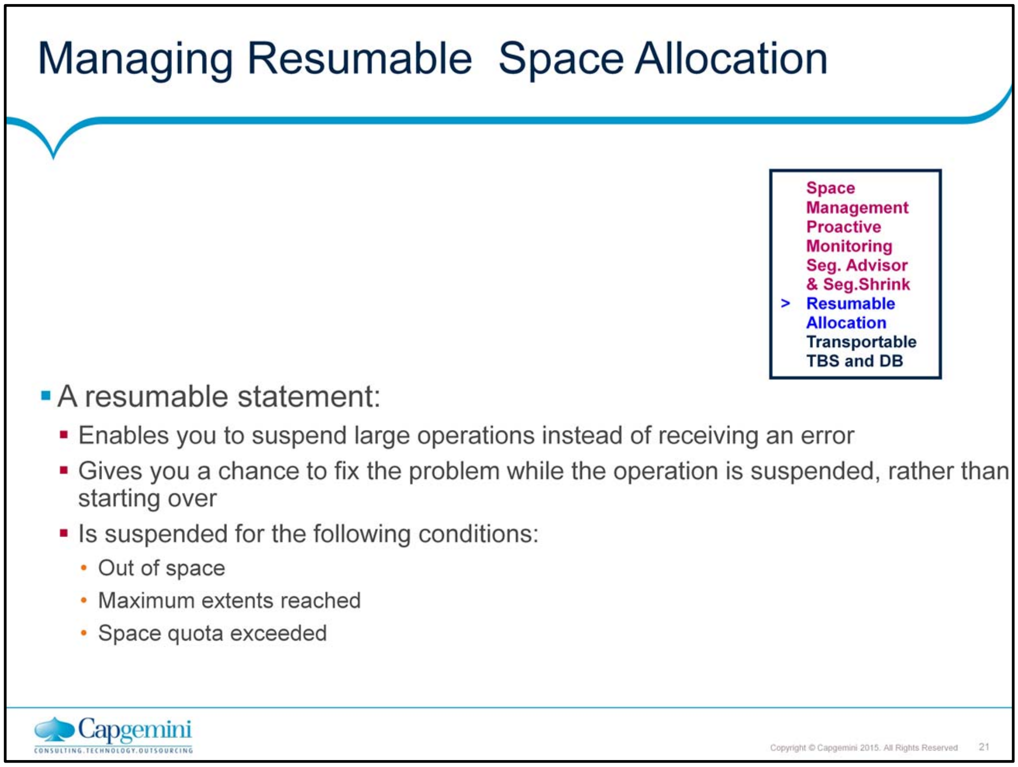
Use the SHRINK SPACE clause to shrink space in a segment. If CASCADE is specified, the shrink behavior is cascaded to all the dependent segments that support a shrink operation, except materialized views, LOB indexes, and IOT (index-organized tables) mapping tables. The SHRINK SPACE clause is illustrated in the second example.

In an index segment, the shrink operation coalesces the index before compacting the data.

Example 3 shows a command that shrinks a LOB segment, given that the RESUME column is a CLOB.

Example 4 shows a command that shrinks an IOT overflow segment belonging to the EMPLOYEES table.

Note: For more information, refer to the Oracle Database SQL Reference guide.



Managing Resumable Space Allocation

Space Management
Proactive Monitoring
Seg. Advisor & Seg.Shrink
> Resumable Allocation
Transportable TBS and DB

- A resumable statement:
  - Enables you to suspend large operations instead of receiving an error
  - Gives you a chance to fix the problem while the operation is suspended, rather than starting over
  - Is suspended for the following conditions:
    - Out of space
    - Maximum extents reached
    - Space quota exceeded

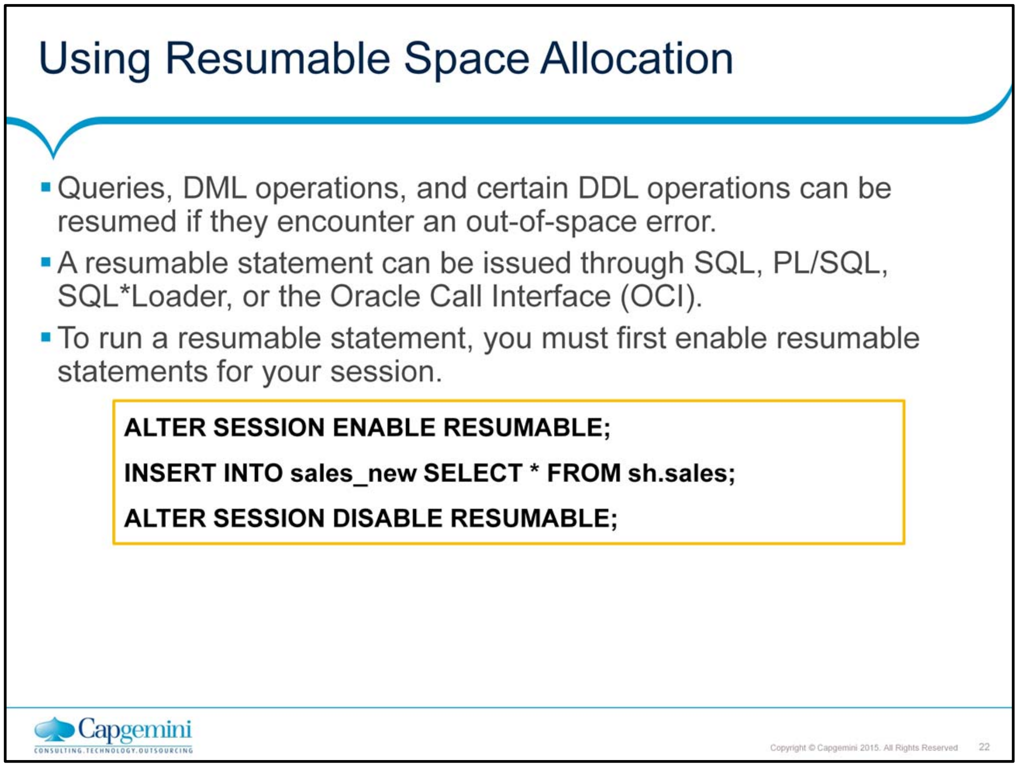
Managing Resumable Space Allocation

The Oracle database provides a means for suspending, and later resuming, the execution of large database operations in the event of space allocation failures. This enables you to take corrective action instead of the Oracle database server returning an error to the user. After the error condition is corrected, the suspended operation automatically resumes. This feature is called “resumable space allocation.” The statements that are affected are called “resumable statements.”

A statement executes in resumable mode only when the resumable statement feature has been enabled for the system or session.

Suspending a statement automatically results in suspending the transaction. Thus all transactional resources are held through the suspension and resuming of a SQL statement. When the error condition disappears (for example, as a result of user intervention or perhaps sort space released by other queries), the suspended statement automatically resumes execution.

A suspension time-out interval is associated with resumable statements. A resumable statement that is suspended for the time-out interval (the default is 7,200 seconds (2 hours)) reactivates itself and returns the exception to the user. A resumable statement can be suspended and resumed multiple times during execution.



Using Resumable Space Allocation

- Queries, DML operations, and certain DDL operations can be resumed if they encounter an out-of-space error.
- A resumable statement can be issued through SQL, PL/SQL, SQL*Loader, or the Oracle Call Interface (OCI).
- To run a resumable statement, you must first enable resumable statements for your session.

```
ALTER SESSION ENABLE RESUMABLE;
INSERT INTO sales_new SELECT * FROM sh.sales;
ALTER SESSION DISABLE RESUMABLE;
```

Using Resumable Space Allocation

Resumable space allocation is possible only when statements are executed within a session that has resumable mode enabled. There are two means of enabling and disabling resumable space allocation:

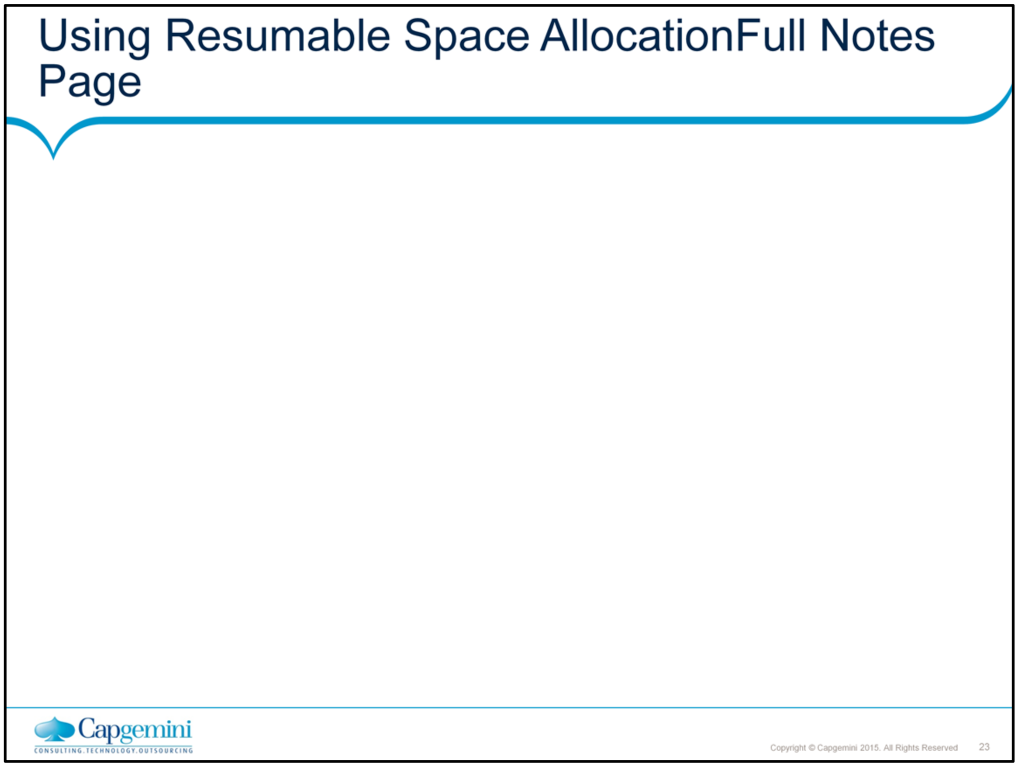
Issue the ALTER SESSION ENABLE RESUMABLE command.

Set the RESUMABLE_TIMEOUT initialization parameter to a nonzero value with an ALTER SESSION or ALTER SYSTEM statement.

When enabling resumable mode for a session or the database, you can specify a time-out period, after which a suspended statement errors out if no intervention has taken place. The RESUMABLE_TIMEOUT initialization parameter indicates the number of seconds before a time-out occurs. You can also specify the time-out period with the following command:

```
ALTER SESSION ENABLE RESUMABLE
TIMEOUT 3600;
```

The value of TIMEOUT remains in effect until it is changed by another ALTER SESSION ENABLE RESUMABLE statement, it is changed by another means, or the session ends. The default time-out interval when using the ENABLE RESUMABLE TIMEOUT clause to enable resumable mode is 7,200 seconds, or 2 hours.



Using Resumable Space AllocationFull Notes Page

Using Resumable Space Allocation (continued)

You can also give a name to resumable statements. For example:

```
ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600
NAME 'multitab insert';
```

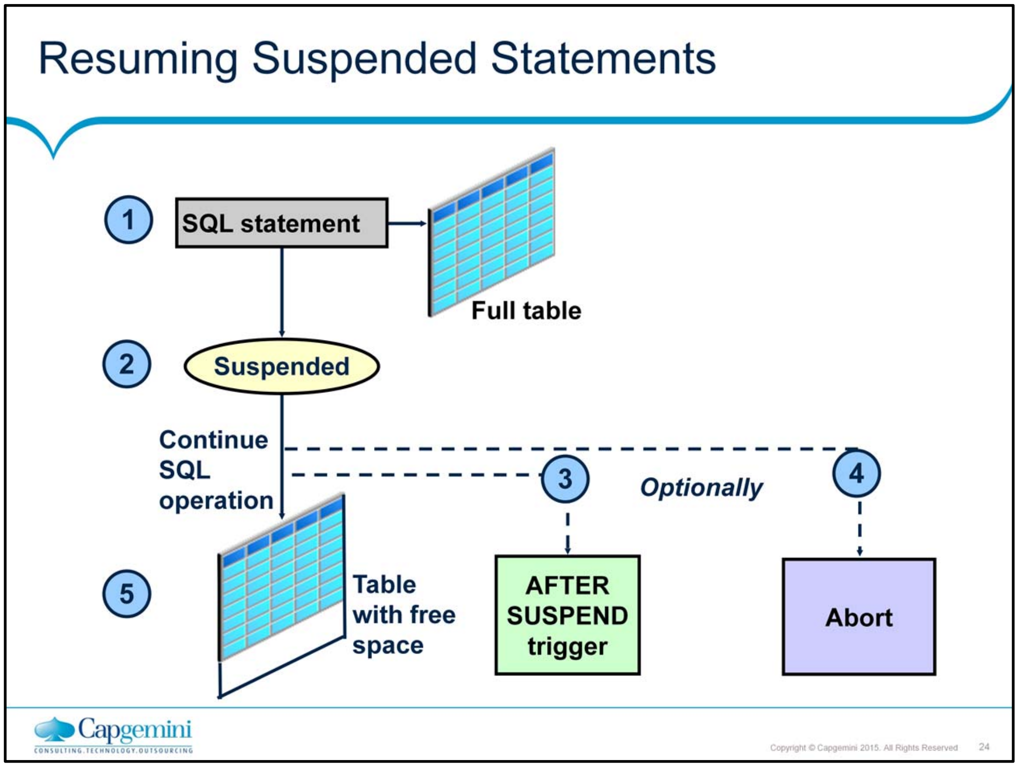
The name of the statement is used to identify the resumable statement in the DBA_RESUMABLE and USER_RESUMABLE views.

For example:

```
SELECT name, sql_text FROM user_resumable;

NAME           SQL_TEXT
--------------- --------------------------------------------
multitab insert INSERT INTO oldsales SELECT * FROM
sh.sales;
```

To automatically configure resumable statement settings for individual sessions, you can create and register a database-level LOGON trigger that alters a user’s session. The trigger issues commands to enable resumable statements for the session, specifies a time-out period, and associates a name with the resumable statements issued by the session.

Because suspended statements can hold up some system resources, users must be granted the RESUMABLE system privilege before they are allowed to enable resumable space allocation and execute resumable statements.



Resuming Suspended Statements

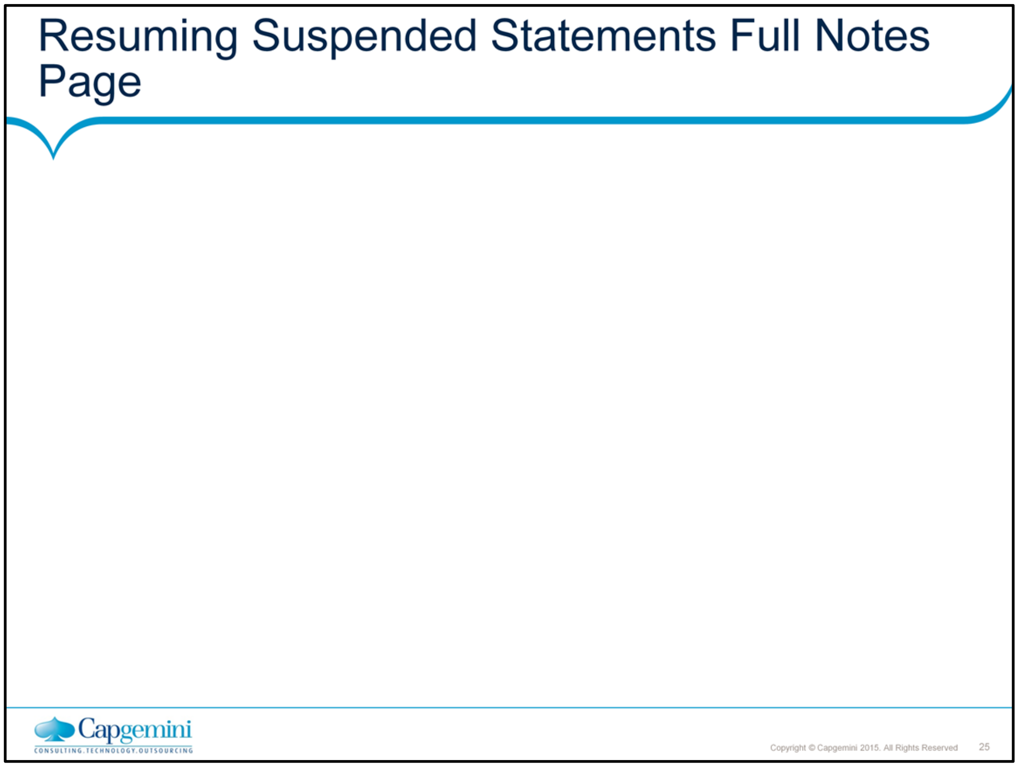
Resuming Suspending Statements

Example:

1. An INSERT statement encounters an error saying the table is full.
2. The INSERT statement is suspended, and no error is passed to client.
3. Optionally, an AFTER SUSPEND trigger is executed.
4. Optionally, the SQLERRROR exception is activated to abort the statement.
5. If the statement is not aborted and free space is successfully added to the table, the INSERT statement resumes execution.

Detecting a Suspended Statement

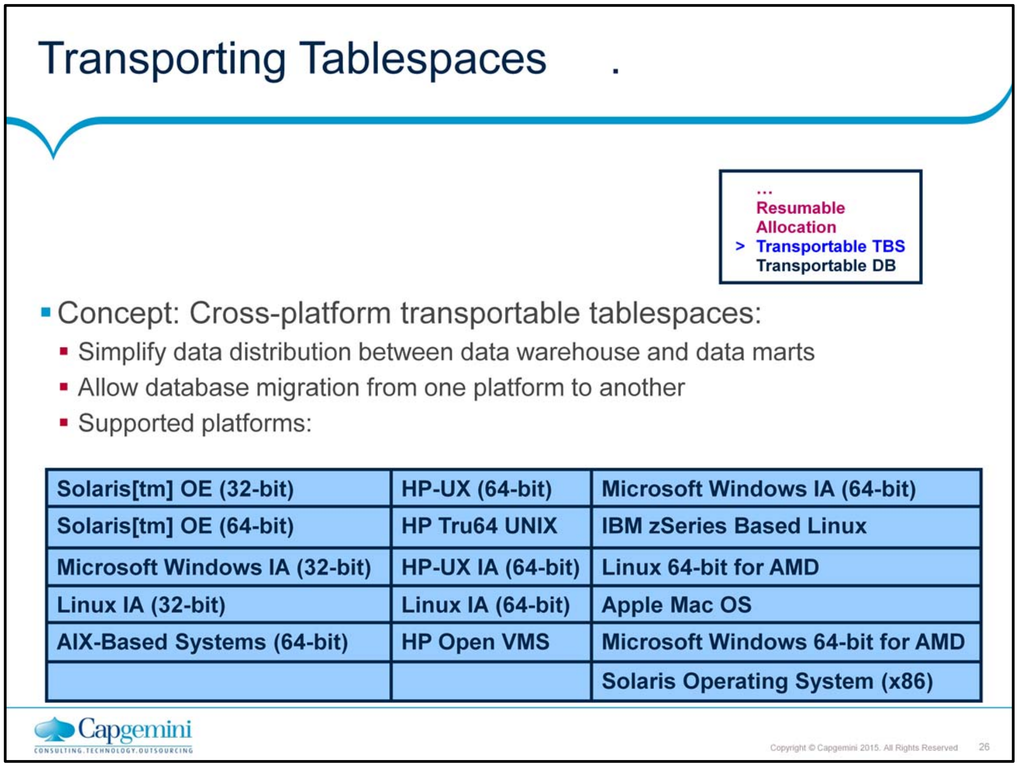
When a resumable statement is suspended, the error is not raised to the client. In order for corrective action to be taken, the Oracle database provides alternative methods for notifying users of the error and for providing information about the circumstances.



Resuming Suspended Statements Full Notes Page

Resuming Suspended Statements (continued)

- Possible Actions During Suspension
- When a resumable statement encounters a correctable error, the system internally generates the AFTER SUSPEND system event. Users can register triggers for this event at both the database and schema level. If a user registers a trigger to handle this system event, the trigger is executed after a SQL statement has been suspended. SQL statements executed within an AFTER SUSPEND trigger are always nonresumable and are always autonomous. Transactions started within the trigger use the SYSTEM rollback segment. These conditions are imposed to overcome deadlocks and reduce the chance of the trigger experiencing the same error condition as the statement.
- Within the trigger code, you can use the USER_RESUMABLE or DBA_RESUMABLE views, or the DBMS_RESUMABLE.SPACE_ERROR_INFO function to get information about the resumable statements.
- When a resumable statement is suspended:
  - The session invoking the statement is put into a wait state. A row is inserted into V$SESSION_WAIT for the session with the EVENT column containing “statement suspended, wait error to be cleared”.
  - An operation-suspended alert is issued on the object that needs additional resources for the suspended statement to complete.
- Ending a Suspended Statement
- When the error condition is resolved (for example, as a result of DBA intervention or perhaps sort space released by other queries), the suspended statement automatically resumes execution and the “resumable session suspended” alert is cleared.
- A suspended statement can be forced to activate the SERVERERROR exception by using the DBMS_RESUMABLE.ABORT() procedure. This procedure can be called by a DBA, or by the user who issued the statement. If the suspension time-out interval associated with the resumable statement is reached, the statement aborts automatically and an error is returned to the user.



Transporting Tablespaces .

...
Resumable
Allocation
> Transportable TBS
Transportable DB

- Concept: Cross-platform transportable tablespaces:
  - Simplify data distribution between data warehouse and data marts
  - Allow database migration from one platform to another
  - Supported platforms:

Solaris[tm] OE (32-bit)	HP-UX (64-bit)	Microsoft Windows IA (64-bit)
Solaris[tm] OE (64-bit)	HP Tru64 UNIX	IBM zSeries Based Linux
Microsoft Windows IA (32-bit)	HP-UX IA (64-bit)	Linux 64-bit for AMD
Linux IA (32-bit)	Linux IA (64-bit)	Apple Mac OS
AIX-Based Systems (64-bit)	HP Open VMS	Microsoft Windows 64-bit for AMD
		Solaris Operating System (x86)

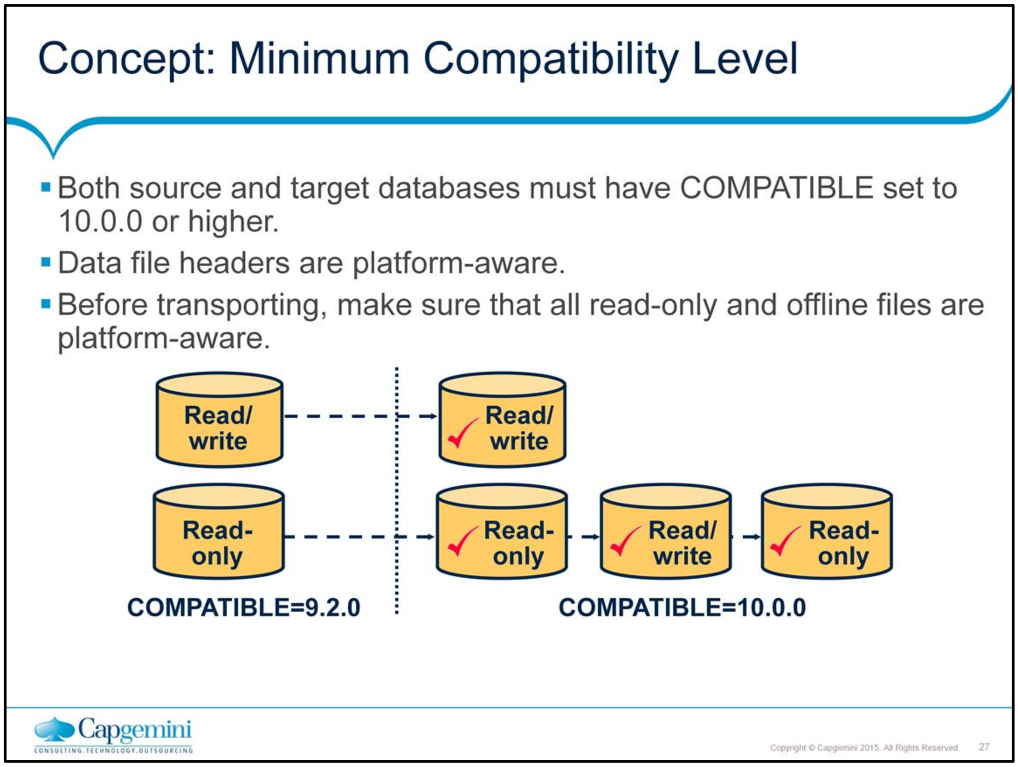
26

Transporting Tablespaces

You can use the transportable tablespace feature to move data across platform boundaries. This simplifies the distribution of data from a data warehouse environment to data marts, which often run on smaller platforms. It also allows a database to be migrated from one platform to another by rebuilding the dictionary and transporting the user tablespaces.

To be able to transport data files from one platform to another, you must ensure that both the source system and the target system are running on one of the supported platforms (see slide).

Note: The cross-platform transportable tablespace feature requires both platforms to be using the same character sets.



Concept: Minimum Compatibility Level

- Both source and target databases must have COMPATIBLE set to 10.0.0 or higher.
- Data file headers are platform-aware.
- Before transporting, make sure that all read-only and offline files are platform-aware.

Read/ write

Read/ write

Read- only

Read- only

Read/ write

Read- only

COMPATIBLE=9.2.0

COMPATIBLE=10.0.0

Capgemini

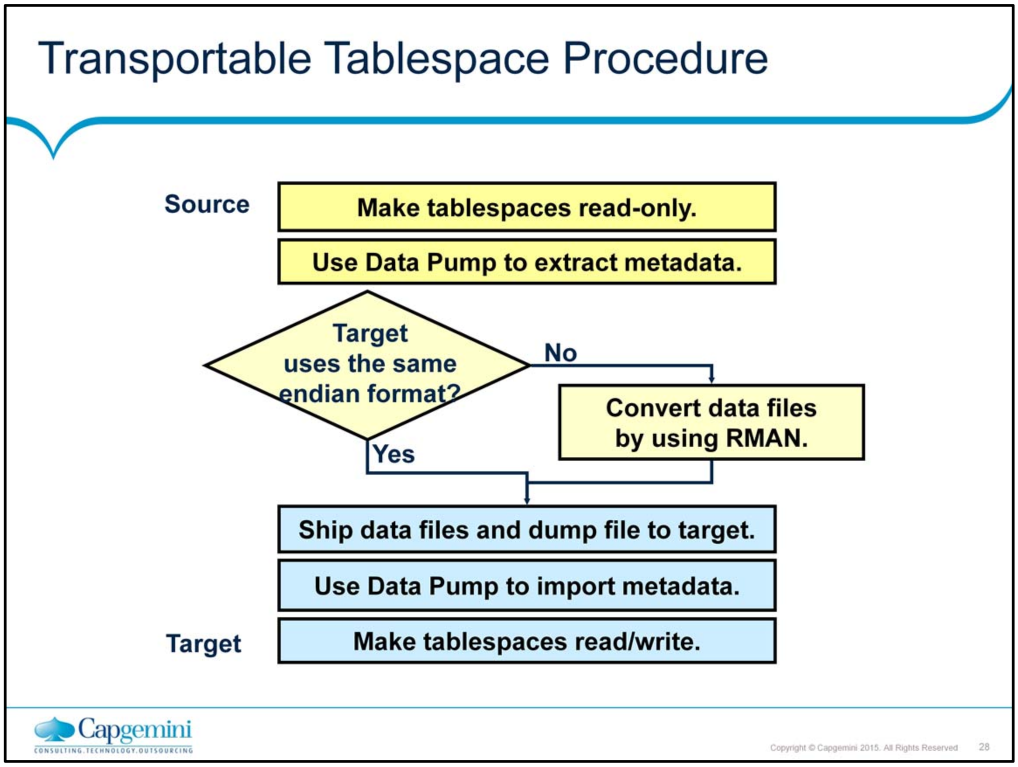
CONSULTING.TECHNOLOGY.OUTSOURCING

27

Concept: Minimum Compatibility Level

Both source and target databases need to advance their database COMPATIBLE initialization parameter to 10.0.0 or greater before they can use the cross-platform transportable tablespace feature.

When data files are first opened in Oracle Database 11g with COMPATIBLE set to 10.0.0 (or greater), the files are made platform-aware. This is represented by the check marks in the diagram. Each file identifies the platform that it belongs to. These files have identical on-disk formats for file header blocks that are used for file identification and verification. Read-only and offline files get the compatibility advanced only after they are made read/write or are brought online. This implies that tablespaces that are read-only in databases before Oracle Database 11g must be made read/write at least once before they can use the cross-platform transportable feature.



Transportable Tablespace Procedure

Source

Make tablespaces read-only.

Use Data Pump to extract metadata.

Target uses the same endian format?

No

Convert data files by using RMAN.

Yes

Ship data files and dump file to target.

Use Data Pump to import metadata.

Target

Make tablespaces read/write.

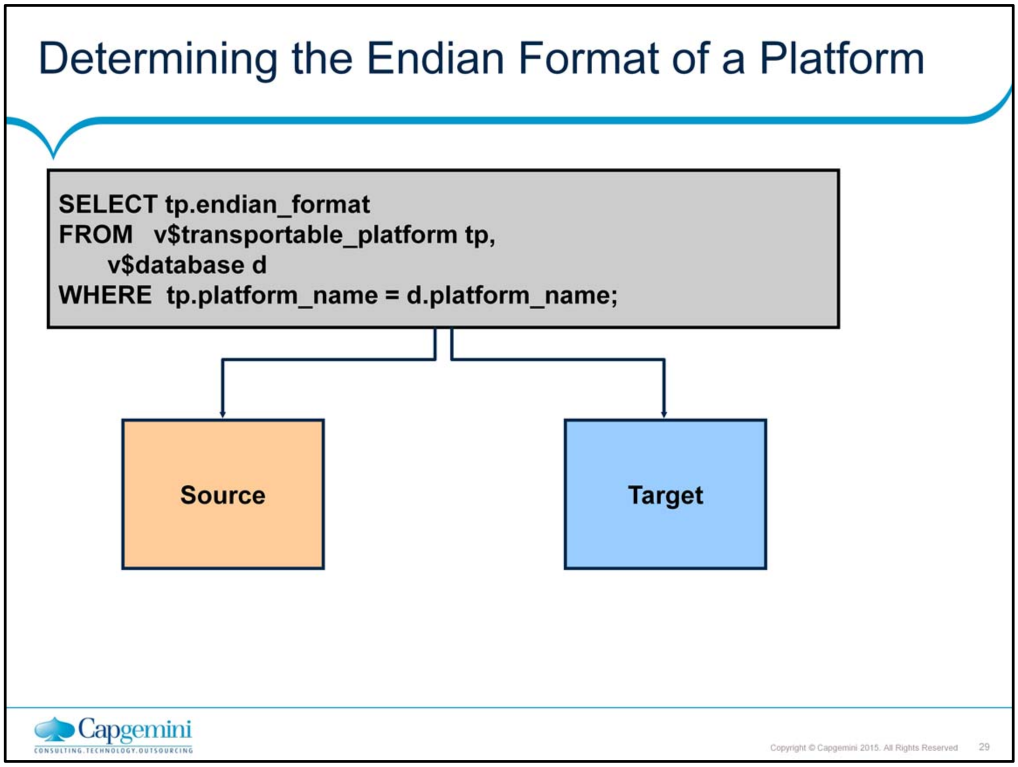
Transportable Tablespace Procedure

To transport a tablespace from one platform to another (source to target), data files belonging to the tablespace set must be converted to a format that can be understood by the target or destination database. Although with Oracle Database 11g, disk structures conform to a common format, it is possible for the source and target platforms to use different endian formats (byte ordering). When going to a different endian platform, you must use the CONVERT command of the RMAN utility to convert the byte ordering. This operation can be performed on either the source or the target platforms. For platforms that have the same endian format, no conversion is needed.

The slide graphic depicts the possible steps to transport tablespaces from a source platform to a target platform. However, it is possible to perform the conversion after shipping the files to the target platform. The last two steps must be executed on the target platform.

Basically, the procedure is the same as when using previous releases of the Oracle database except when both platforms use different endian formats. It is assumed that both platforms are cross-transportable compliant.

Note: Byte ordering can affect the results when data is written and read. For example, the 2-byte integer value 1 is written as 0x0001 on a big-endian system (such as Sun SPARC Solaris) and as 0x0100 on a little-endian system (such as an Intel-compatible PC).



Determining the Endian Format of a Platform

```
SELECT tp.endian_format
FROM   v$transportable_platform tp,
       v$database d
WHERE  tp.platform_name = d.platform_name;
```

Determining the Endian Format of a Platform

You can query V$TRANSPORTABLE_PLATFORM to determine whether the endian ordering is the same on both platforms. V$DATABASE has two columns that can be used to determine your own platform name and platform identifier.

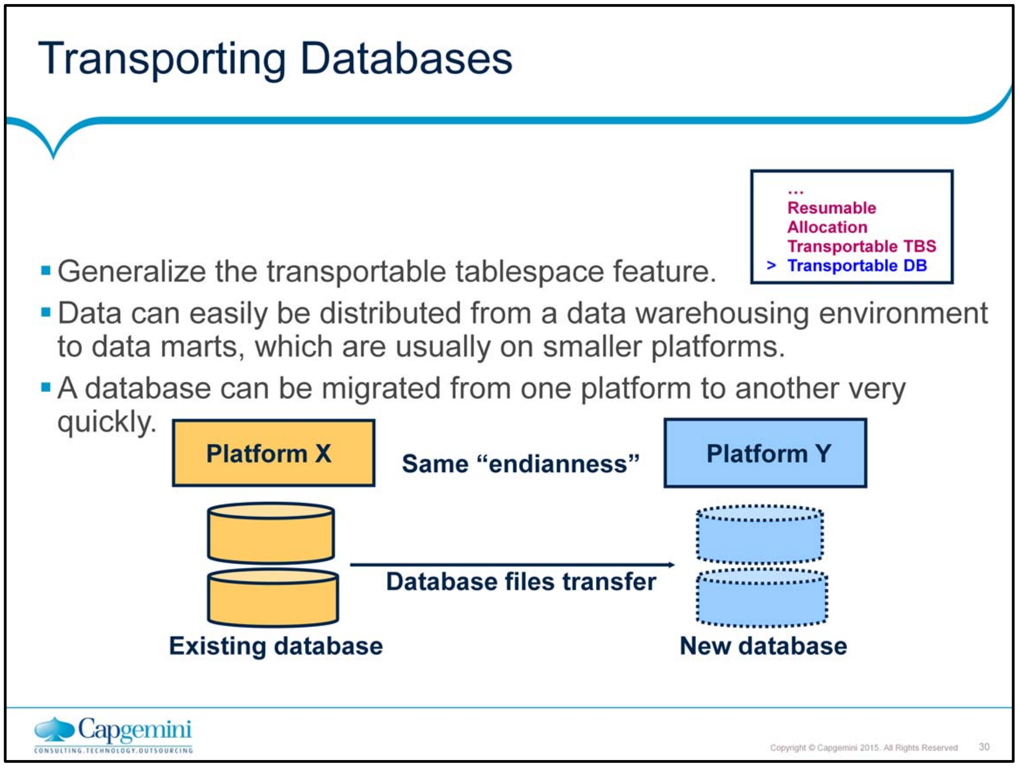
Use the query in the slide on both platforms, and then compare the results.

On a Sun SPARC Solaris system, the SELECT statement produces the following output:

```
ENDIAN_FORMAT
-------------
Big
```

On a Microsoft Windows Intel-based platform, the SELECT statement produces the following output:

```
ENDIAN_FORMAT
-------------
Little
```



Transporting Databases

…
Resumable
Allocation
Transportable TBS
> Transportable DB

- Generalize the transportable tablespace feature.
- Data can easily be distributed from a data warehousing environment to data marts, which are usually on smaller platforms.
- A database can be migrated from one platform to another very quickly.

Transporting Databases

With the transportable tablespace feature, moving data across different platforms becomes much faster. However, metadata still needs to be unloaded, because the system tablespace cannot be transported.

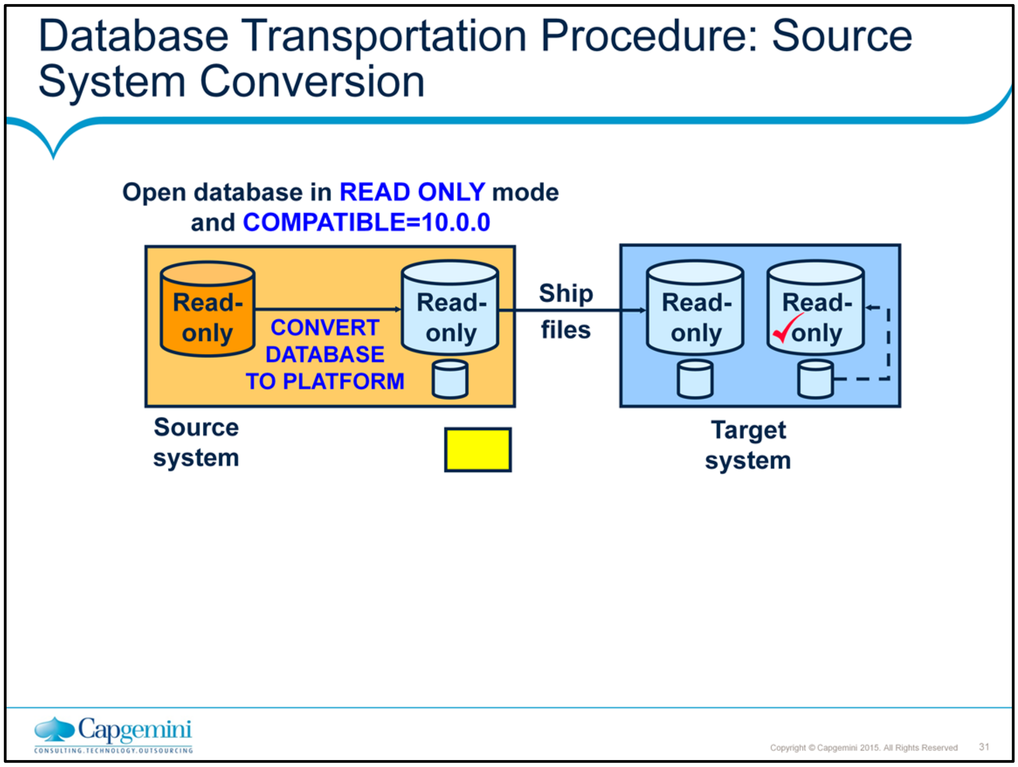
The purpose of the database transport feature is to provide a fast and easy way to transport a database across different platforms with the same endian format. However, the source platform and the target platform can have different disk alignments. For example, HP-UX and Solaris both have big endian, but the disk alignment is eight on HP-UX and four on Solaris.

To transport databases from one platform to another, you must ensure that both the source system and the target system are running on one of the platforms that are listed in V$TRANSPORTABLE_PLATFORM and that both have the same endian format. For example, you can transport a database running on Linux IA (32-bit) to one of the Windows platforms.

If one or both of the databases uses Automatic Storage Management (ASM), you may need to use the DBMS_FILE_TRANSFER package to ftp the files.

Unlike transportable tablespace, where there is a target database to plug data into, this feature creates a new database on the target platform. The newly created database contains the same data as the source database. Except for things such as database name, instance name, and location of files, the new database also has the same settings as the source database.

Note: Transporting database is faster than using Data Pump to move data.



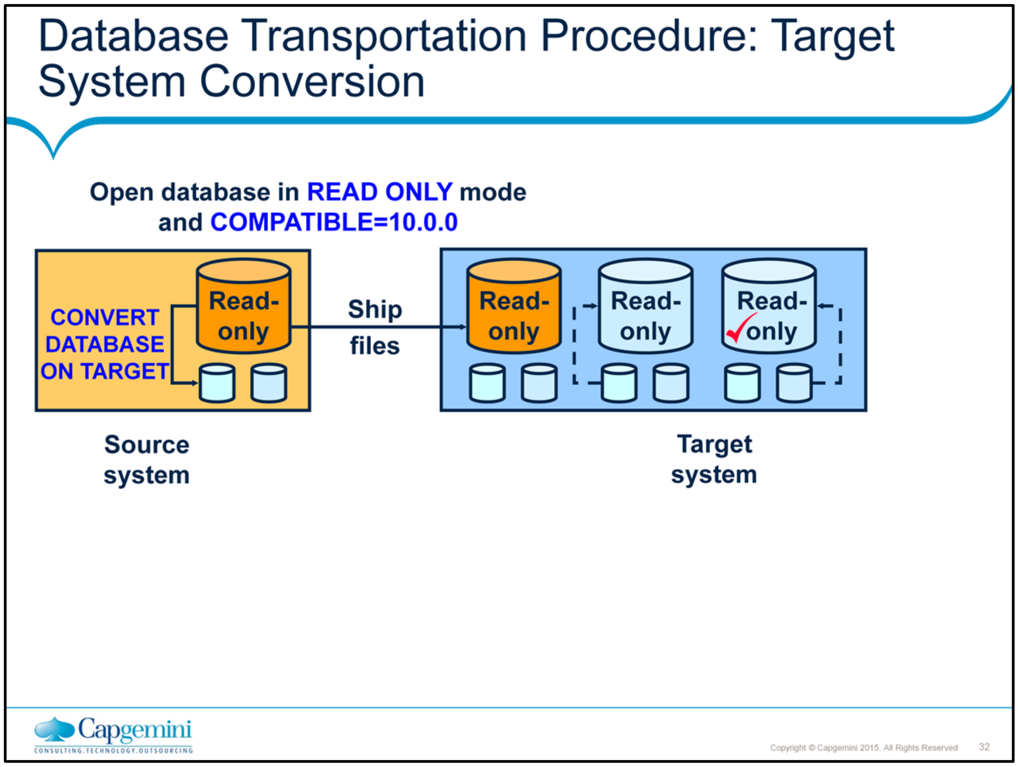
Database Transportation Procedure: Source System Conversion

Open database in READ ONLY mode
and COMPATIBLE=10.0.0

Database Transportation Procedure: Source System Conversion

Before you can transport your database, you must open it in READ ONLY mode. Then use RMAN to convert the necessary data files of the database. When you do the conversion on the source platform, the new RMAN command CONVERT DATABASE generates a script containing the correct CREATE CONTROLFILE RESETLOGS command that is used on the target system to create the new database. The CONVERT DATABASE command then converts all identified data files so that they can be used on the target system. You then ship the converted data files and the generated script to the target platform. By executing the generated script on the target platform, you create a new copy of your database.

Note: The source database must be running with the COMPATIBLE initialization parameter set to 10.0.0 or higher. All identified tablespaces must have been READ WRITE at least once since COMPATIBLE was set to 10.0.0 or higher.



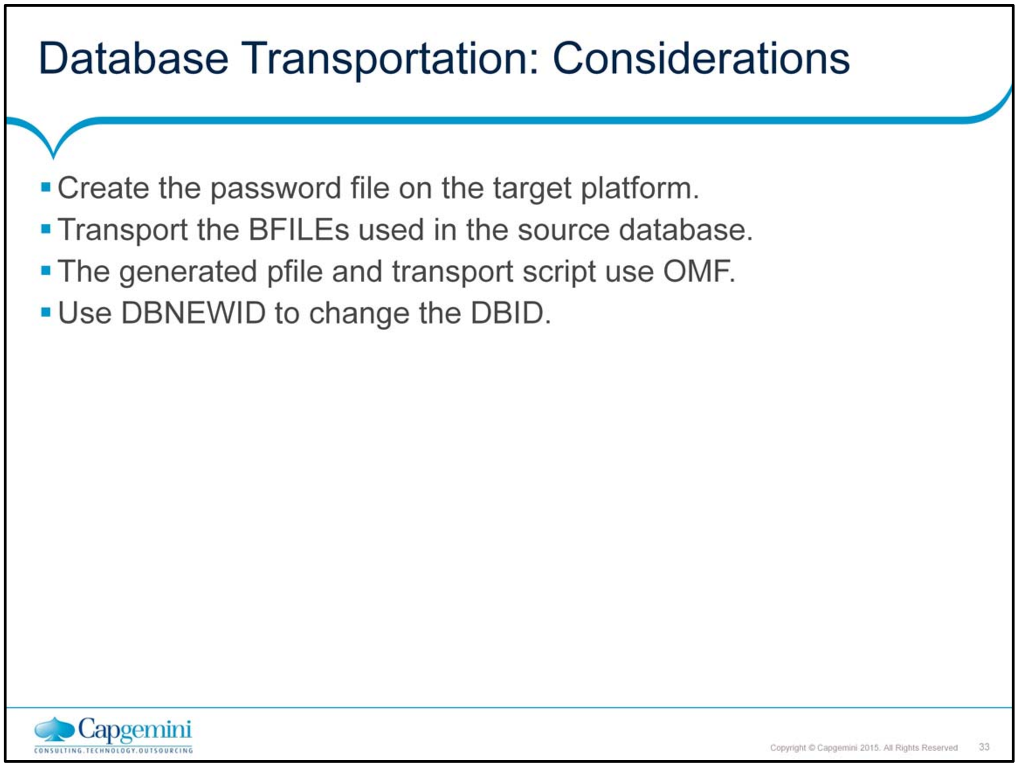
Database Transportation Procedure: Target System Conversion

Open database in READ ONLY mode and COMPATIBLE=10.0.0

Database Transportation Procedure: Target System Conversion

Before you can transport your database, you must open it in READ ONLY mode. Then use RMAN to convert the necessary data files of the database. When you do the conversion on the target platform, the CONVERT DATABASE command (which is executed on the source system) generates only two scripts used on the target system to convert the data files, and to re-create the control files for the new database. Then, you ship the identified data files and both scripts to the target platform. After this is done, execute both scripts in the right order. The first one uses the existing CONVERT DATAFILE RMAN command to do the conversion, and the second issues the CREATE CONTROLFILE RESETLOGS SQL command with the converted data files to create the new database.

Note: The source database must be running with the COMPATIBLE initialization parameter set to 10.0.0 or higher. All identified tablespaces must have been READ WRITE at least once since COMPATIBLE was set to 10.0.0 or higher.



Database Transportation: Considerations

- Create the password file on the target platform.
- Transport the BFILEs used in the source database.
- The generated pfile and transport script use OMF.
- Use DBNEWID to change the DBID.

Database Transportation: Considerations

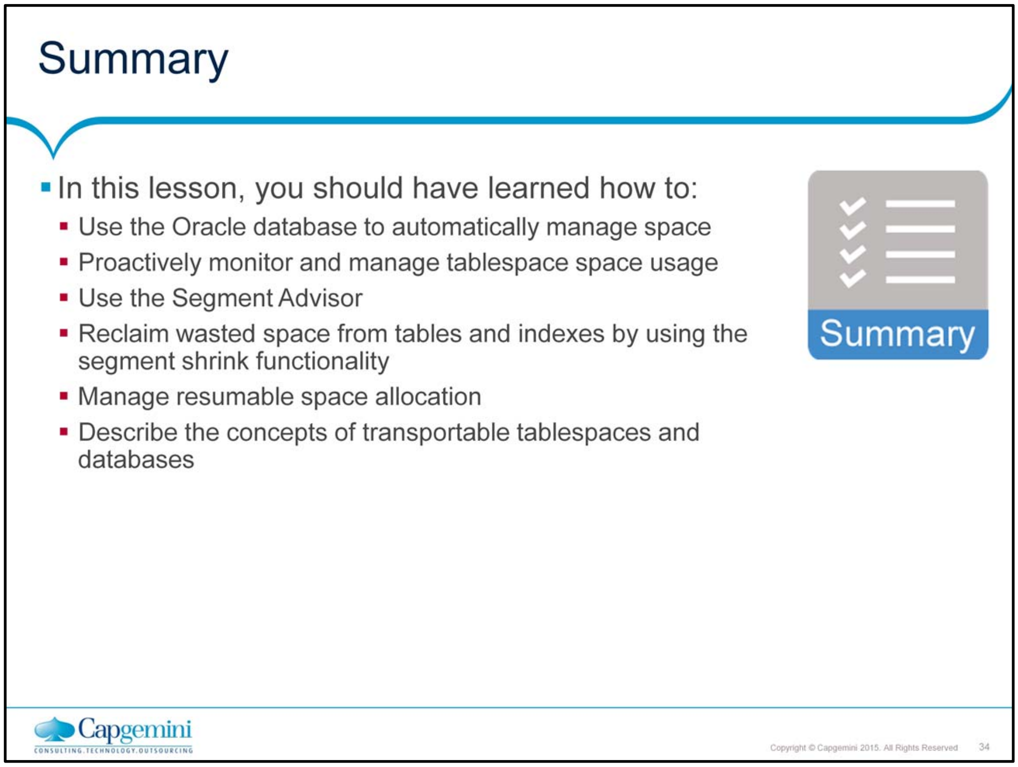
Redo logs, control files, and tempfiles are not transported. They are re-created for the new database on the target platform. As a result, the new database on the target platform must be opened with the RESETLOGS option.

If a password file is used, it is not transported and you need to create it on the target platform. This is because the types of file names allowed for the password file are OS specific. However, the output of the CONVERT DATABASE command lists all the usernames and their system privileges, and advises to re-create the password file and add entries for these users on the target platform.

The CONVERT DATABASE command lists all the directory objects and objects that use BFILE data types or external tables in the source database. You may need to update these objects with new directory and file names. If BFILEs are used in the database, you have to transport the BFILEs.

The generated pfile and transport script use Oracle Managed Files (OMF) for database files. If you do not want to use OMF, you must modify the pfile and transport script.

The transported database has the same DBID as the source database. You can use the DBNEWID utility to change the DBID. In the transport script as well as the output of the CONVERT DATABASE command, you are prompted to use the DBNEWID utility to change the database ID.



Summary

- In this lesson, you should have learned how to:
  - Use the Oracle database to automatically manage space
  - Proactively monitor and manage tablespace space usage
  - Use the Segment Advisor
  - Reclaim wasted space from tables and indexes by using the segment shrink functionality
  - Manage resumable space allocation
  - Describe the concepts of transportable tablespaces and databases