

# Oracle Database 10g Release 2 Online Data Reorganization & Redefinition

*An Oracle White Paper*

*May, 2005*

# Oracle Database 10g Online Data Reorganization & Redefinition

Executive Overview .....	3
Introduction .....	3
Online Data Redefinition .....	4
Steps for Online Redefinition of Tables from SQL*PLUS.....	6
Intermediate Synchronization .....	7
Manually Defining Triggers, Constraints, Indexes & Grants.....	7
Abort and Cleanup After Errors .....	8
Example of an Online Table Redefinition .....	8
Example of Reorganizing an Individual Partition.....	9
Additional Online Data Reorganization Features .....	10
Online Table & Index Operations .....	10
Reclaiming Unused Space.....	11
Other Database Reorganization Features .....	11
Conclusion.....	12

# Oracle Database 10g Release 2 Online Data Reorganization & Redefinition

## EXECUTIVE OVERVIEW

The days when a company could take its system offline for any kind of maintenance are rapidly disappearing. As businesses become global and move toward e-commerce, systems now have to be highly available because the cost of outage for corporations involved in e-commerce can easily reach millions of dollars per hour. Today, it is unlikely that the customer will come back if your systems are unavailable, they will simply give that business to your competitor whose systems are online.

Online data reorganization, or the ability to allow users full access to the database during a data reorganization, improves the overall database availability and reduces planned downtime. Oracle9i introduced the online data redefinition capability that allows database administrators to perform not only data reorganization online but also data redefinition online.

Each release of the Oracle database has introduced enhanced online data reorganization features such as creating indexes online, rebuilding indexes online, coalescing indexes online, moving index-organized tables (IOTs) online.

The online data redefinition feature offers administrators unprecedented flexibility to modify table physical attributes and transform both data and table structure while allowing users full access to the database. This feature can also make the application upgrade process easier, safer and faster.

## INTRODUCTION

Oracle Database 10g Release 2 online data reorganization and redefinition features are designed to enhance database availability, manageability, and performance. This paper provides an overview of online data redefinition and some examples on how to use the feature.

Oracle Database 10g added the following online capabilities:

- Copy dependant objects i.e. triggers, constraints, indexes
- Convert LONG and LONG RAW columns to a LOB
- Use a unique key as an alternative to a primary key or rowid

- Specify columns to order data by
- Change a table without recompiling stored procedures
- Online segment shrink

Oracle Database 10g Release 2 added the following online capabilities:

- Reorganize a Single Partition
- Reorganize Advanced Queue and Clustered Tables
- Reorganize a table containing an ADT
- Improved Copy Dependent Objects Capabilities
  - Retain and Clone Statistics
  - Check and Not Null Constraints
  - Dependent objects for nested tables

Administrators can continue to use existing features like partition and parallel operations to complement Oracle Database 10g online features. For example, partitioning can be used to support large historical tables and their indexes by dividing historical data into time-related partitions that can be managed independently and added or dropped conveniently.

Use of the parallel option during data reorganization also improves performance by performing operations concurrently. This is especially useful when multiple partitions are involved, for example, rebuilding indexes in multiple partitions.

The Oracle Database 10g online capabilities improve data availability, database performance, response time, and disk space utilization, all of which are important in today's mission-critical environment.

## **ONLINE DATA REDEFINITION**

For large, active databases, it is sometime necessary to redefine large “hot” tables to improve the performance of queries or data manipulation language (DML) operations performed against these tables. Additionally business applications may require underlying database structure to be changed or transformed periodically. Oracle Online Redefinition mechanism provides a significant increase in availability compared to traditional methods of redefining tables that require tables to be taken offline.

When a table is redefined online, it is accessible by all read and write operations during the redefinition process. Administrators then have control over when to switch from the original to the newly redefined table. The switch process is very brief and is independent of the size of the table or the complexity of the redefinition. The redefinition process effectively creates a new table and improves its data block layout efficiency.

The Oracle Database 10g release 2 online data redefinition feature improves data availability, database performance, response time, and disk space utilization.

Online table redefinition allows administrators to:

- Modify the physical attributes or storage parameters of a table, advanced queue or clustered table
- Move a heap table or IOT to a different tablespace
- Add support for parallel queries
- Add or drop partitioning support
- Reorganize a Single Partition
- Recreate a heap table or IOT to reduce fragmentation
- Change a heap table to IOT and vice versa
- Add, drop, or rename columns in a table
- Convert a LONG or LONG RAW column to a LOB
- Transform data in a table
- Reorganize a table containing an Abstract Datatype

Furthermore, during an application upgrade, administrators can start the redefinition process, and then periodically synchronize the interim table so that the interim table includes the latest changes to the original table. This reduces the amount of time to complete the final redefinition step. Administrators can also validate and use the data in the interim table before completing the redefinition process. This is a very useful feature for administrators, who must ensure that the upgrade process goes smoothly.

Online Data Reorganization can be accomplished using either Oracle Enterprise Manager (OEM) or SQL\*Plus command line interface. Figure 1 below shows a sample OEM Reorganize Object Wizard view.



Figure 1 Sample OEM Reorganize Object Wizard View

## Steps for Online Redefinition of Tables from SQL\*PLUS

The Oracle Database 10g DBMS\_REDEFINITION PL/SQL package includes the following procedures:

- DBMS\_REDEFINITION.ABORT\_REDEF\_TABLE()
- DBMS\_REDEFINITION.COPY\_TABLE\_DEPENDENTS()
- DBMS\_REDEFINITION.FINISH\_REDEF\_TABLE()
- DBMS\_REDEFINITION.REGISTER\_DEPENDENT\_OBJECT()
- DBMS\_REDEFINITION.START\_REDEF\_TABLE()
- DBMS\_REDEFINITION.SYNC\_INTERIM\_TABLE()
- DBMS\_REDEFINITION.UNREGISTER\_DEPENDENT\_OBJECT()

There are also two tables that may be useful to query during the redefinition process:

- DBA\_REDEFINITION\_OBJECTS, objects used during redefinition
- DBA\_REDEFINITION\_ERRORS

To perform an online redefinition of a table, administrators need to take the following steps.

1. Create an empty interim table with all of the desired attributes.
2. Start the redefinition process by calling DBMS\_REDEFINITION.START\_REDEF\_TABLE(), providing the following:
  - The schema name
  - The table to be redefined
  - The interim table name
  - The column mapping. If the column mapping information is not supplied, then it is assumed that all the columns (with their names unchanged) are to be included in the interim table. If the column mapping is supplied, then only those columns specified explicitly in the column mapping are considered
  - Redefinition method to use; primary key, a unique index or rowed
  - Optionally specify columns to order data by
  - Optionally specify the partition to reorganize
3. Optionally, manually redefine triggers, constraints, indexes and grants on the interim table and then register them using using DBMS\_REDEFINITION.REGISTER\_DEPENDENT\_OBJECT().

This step is only required if the definition of the trigger, constraint, index or grant will be different from the one already defined on the table.

4. If triggers, constraints, indexes and grants are defined on the table. These can be cloned on the interim table by executing `DBMS_REDEFINITION.COPY_TABLE_DEPENDENTS()`. Once redefinition is complete, the referential constraints and triggers involving the interim table (created disabled) are enabled, all statistics will be copied, and all cloned objects will be renamed to their pre-definition name.
5. Optionally synchronize and validate data in the interim table periodically by executing `DBMS_REDEFINITION.SYNC_INTERIM_TABLE`
6. Execute the `DBMS_REDEFINITION.FINISH_REDEF_TABLE()` procedure to complete the redefinition of the table.
  - The original table is redefined such that it has all the attributes, indexes, constraints, grants and triggers of the interim table
  - All referential constraints and triggers are enabled

Any PL/SQL procedures and cursors defined on the original table are invalidated. They are automatically re-validated whenever they are used next.

### **Intermediate Synchronization**

After the redefinition process has been started by calling `START_REDEF_TABLE()` and before `FINISH_REDEF_TABLE()` has been called, it is possible that a large number of DML operations have occurred on the original table. If that's the case, it is recommended that the administrator periodically synchronize the interim table with the original table. This is done by calling the `DBMS_REDEFINITION.SYNC_INTERIM_TABLE()` procedure. Calling this procedure reduces the time taken by `FINISH_REDEF_TABLE()` to complete the redefinition process.

The `DBMS_REDEFINITION.SYNC_INTERIM_TABLE()` procedure can be called multiple times.

### **Manually Defining Triggers, Constraints, Indexes & Grants**

There may be times when you want to override the actions of `DBMS_REDEFINITION.COPY_TABLE_DEPENDENTS()`. For example, an index must be moved to a different tablespace. This can be achieved by making the relevant changes on the interim table, and then using `DBMS_REDEFINITION.REGISTER_DEPENDENT_OBJECT()` to register the new version of the object. Then when `DBMS_REDEFINITION.COPY_TABLE_DEPENDENTS()` is executed, it will ignore that object, in this case the index, which has already been defined.

In the unlikely event that a previously registered object is no longer required, it can be unregistered using the procedure `DBMS_REDEFINITION.UNREGISTER_DEPENDENT_OBJECT()`.

### **Abort and Cleanup After Errors**

In the event that an error occurred during the redefinition process, or if the administrator chooses to abort the redefinition process, the `DBMS_REDEFINITION.ABORT_REDEF_TABLE()` procedure can be used. This procedure drops temporary logs and tables associated with the redefinition process. After this procedure is called, the administrator can drop the interim table and its associated objects.

### **Example of an Online Table Redefinition**

This example illustrates online redefinition of a non-partitioned table `emp`, with columns: `empno`, `name`, `salary`, `phone`. The schema name is `u1` and the table is redefined as follows:

- The column `salary` is multiplied by a factor of 1.10 and renamed as `sal`
- The column `phone` is dropped
- A new column `deptno` is added, and a user-defined function `u1.emp_dept` is used to determine the value of `deptno`
- The redefined table is partitioned by range on `empno`
- Place the index `emp_idx2` in tablespace `tbs_2`

#### **Step 1. Create an interim table `int_emp`**

```
CREATE TABLE int_emp
(empno NUMBER ,   name VARCHAR2(100),
 sal NUMBER,      deptno NUMBER DEFAULT 10)
PARTITION BY RANGE(empno)
(PARTITION emp1000 VALUES LESS THAN (1000) TABLESPACE tbs_1,
PARTITION emp2000 VALUES LESS THAN (2000) TABLESPACE tbs_2);
```

#### **Step 2. Create the index on the interim table in different tablespace**

```
CREATE INDEX int_emp_idx2 ON int_emp (deptno) TABLESPACE tbs_2;
```

#### **Step 3. Start the redefinition process**

```
DBMS_REDEFINITION.START_REDEF_TABLE('u1', 'emp', 'int_emp',
'empno empno, name name, salary*1.10 sal, emp_dept (empno) deptno ');
```



**Step 4. Manually Register Changed Index**

```
DBMS_REDEFINITION.REGISTER_DEPENDENT_OBJECT('u1', 'emp',  
'int_emp', dbms_redefinition.cons_index, 'u1', 'emp_idx2', 'int_emp_idx2');
```

**Step5. Clone dependent objects**

```
DBMS_REDEFINITION.COPY_TABLE_DEPENDENTS('u1', 'emp',  
'int_emp', 1, TRUE, TRUE, TRUE, FALSE, no_errors);
```

**Step 6. Optionally, synchronize the interim table int\_emp**

```
DBMS_REDEFINITION.SYNC_INTERIM_TABLE('u1', 'emp', 'int_emp');
```

**Step 7. Complete the redefinition**

```
DBMS_REDEFINITION.FINISH_REDEF_TABLE('u1', 'emp', 'int_emp');
```

After this call the table emp is redefined such that it has all the attributes of the int\_emp table. The table emp becomes the newly redefined table.

**Step 8. Drop the interim table**

```
DROP TABLE int_emp;
```

**Example of Reorganizing an Individual Partition**

This example illustrates reorganizing a single partition, emp2000, in the partitioned emp table, where it is moved to a new tablespace called tbs\_3.

**Step 1. Create anon-partitioned interim table int\_emp**

```
CREATE TABLE int_emp  
(empno NUMBER, name VARCHAR2(100),  
sal NUMBER, deptno NUMBER DEFAULT 10)  
TABLESPACE tbs_3);
```

**Step 2. Start the redefinition process**

```
DBMS_REDEFINITION.START_REDEF_TABLE('u1', 'emp', 'int_emp',  
NULL, dbms_redefinition.cons_use_rowid, 'emp2000');
```

**Step 3. Optionally, synchronize the interim table int\_emp**

```
DBMS_REDEFINITION.SYNC_INTERIM_TABLE('u1', 'emp', 'int_emp',  
'emp2000');
```

**Step 4. Complete the redefinition**

```
DBMS_REDEFINITION.FINISH_REDEF_TABLE('u1', 'emp', 'int_emp',  
'emp2000');
```

**Step 5. Drop the interim table**

DROP TABLE int\_emp;

**ADDITIONAL ONLINE DATA REORGANIZATION FEATURES**

In addition to the DBMS\_REDEFINITION package, some Online Reorganization tasks can also be performed using SQL statements such as CREATE/ALTER INDEX.

**Online Table & Index Operations**

The table below describes the reorganization operations that are possible using the ONLINE clause in the SQL CREATE/ALTER INDEX and TABLE statements.

<b>Table Move Online</b>	<b>Index Create Online</b>	<b>Index Rebuild Online</b>	<b>Index Coalesce Online</b>
ALTER TABLE emp MOVE ONLINE; <ul style="list-style-type: none"><li>Parallel operations not supported</li><li>compressed</li></ul>	CREATE INDEX emp.ename.idx ON emp(ename) ONLINE; <ul style="list-style-type: none"><li>Parallel operations supported</li><li>Partitions supported</li><li>All index types except cluster</li><li>Supported index types: IOT, secondary, reverse key, functional, bitmap, key compressed</li></ul>	ALTER INDEX emp.ename_idx REBUILD ONLINE; <ul style="list-style-type: none"><li>Parallel operations supported</li><li>Partitions supported</li><li>All index types except cluster</li><li>Supported index types: IOT, secondary, reverse key, functional, bitmap, key compressed</li></ul>	ALTER INDEX emp.ename_idx COALESCE; <ul style="list-style-type: none"><li>Parallel operations supported</li><li>Partitions supported</li><li>All index types</li></ul>

## Reclaiming Unused Space

During the life time of the database, many updates and deletes to the data can occur, and this may result in space inside the database not being used efficiently. In Oracle Database 10g, this space can be reclaimed for tables, index organized tables, indexes, partitions and materialized views, provided the objects are stored in tablespaces using automatic segment space management.

The space is reclaimed online, and in-place, thus eliminating expensive database downtime and additional storage requirements. The space is retrieved by adding the clauses `SHRINK SPACE` to the `ALTER TABLE`, `ALTER INDEX`, `ALTER MATERIALIZED VIEW` and `ALTER MATERIALIZED VIEW LOG` commands.

Whilst the space is being reclaimed, indexes are maintained and are completely usable once the process has finished. An optional `CASCADE` clause provides the ability to reclaim space for all dependent objects. Therefore reclaiming space on the table, would also reclaim space on all the indexes on that table.

Further information on other options and some of the restrictions on reclaiming space can be found in Chapter 13 in the Reclaiming Unused Space section, in the Oracle Database 10g Administrator's Guide.

Datafiles can also be dropped whilst the database is open, see Chapter 9 in the Managing Datafiles and Tempfiles in the Oracle Database 10g Administrator's Guide.

Oracle Database 10g also includes the Segment Advisor, which by default runs automatically at regular intervals, looking for segments that are suitable for shrinking. When candidate segments are identified, it sends an alert and generates a recommendation as to whether online reorganization or segment shrink should be used.

## Other Database Reorganization Features

In addition there are also the following features which the DBA may find useful while performing an online reorganization:

- an analyze validate structure online feature, that allows administrators to validate the structure of an object while allowing DML changes
- a database quiesce capability, which allows the administrator to perform maintenance operations without requiring a database to be shut down. e.g., administrators can put the database into a "quiesced state," then change the schema of a database table and update a PL/SQL procedure to a new version that uses this new schema of the database table. This feature requires that Oracle's Database Resource Manager be enabled.

## **CONCLUSION**

Oracle already supports many online features such as adding constraints, triggers, and partitions online; creating, rebuilding, coalescing indexes online; moving index-organized tables online. With the online data redefinition and reorganization features, Oracle Database 10g Release 2, significantly improves overall database availability and manageability. The online redefinition feature also makes the application upgrade process easier, safer and faster. Other benefits from these online features include improved database performance, response time, and disk space utilization.

Oracle Database 10g Release 2 now offers the most complete online data organization and redefinition capabilities in the industry. These online features allow database users and administrators full access to the database during data reorganization and redefinition - an extremely important consideration in today's mission critical computing environment.



Oracle Database 10g Release 2 Online Data Reorganization and Redefinition  
May, 2005

Author: Lilian Hobbs

Contributing Authors: Paul Tsien

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](http://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 © 2005 Oracle Corporation  
All rights reserved.