Performing Reorganizations with Oracle Redefinition Package
Overview

- The Need for Reorganizations
- The Redefinition Package
  - Redefinition Commands
  - Redefinition Process and Steps
  - Limitations
  - Deficiencies
  - Error Handling
- OEM and Automation of the Redefinition Package
- Comparison – Online Redefinition vs. Live Reorganization
- Summary and Conclusions
The Need for Reorganizations

- **Improve Performance**
  - Packing the rows in contiguous space
  - Eliminating Chained Rows

- **Protect Availability**
  - Prevent out-of-space error conditions

- **Recover Wasted Disk Space**
  - Inside the block and below the High Water Mark

- **Optimize Database Layout**
  - Relocating to different tablespace to relieve I/O contention
The Redefinition Package

- Introduced in Oracle version 9.0
  - Intended to change table attributes online
  - Redefinition copies the table, so reorganization is by-product

- DBMS_REDEFINITION Commands
  - CAN_REDEF_TABLE
  - START_REDEF_TABLE
  - SYNC_INTERIM_TABLE
  - REGISTER_DEPENDENT_OBJECT (10G only)
  - FINISH_REDEFTABLE
A Manual Redefinition Script

/* Create interim table */
create table mytable_123456
(n number,
m number,
v varchar2(1000));

/* Start redefinition */
begin
  dbms_redefinition.start_redef_table('QUEST', 'MYTABLE', 'MYTABLE_123456', null, dbms_redefinition.cons_use_pk);
end;
/

NOTE: START_REDEF_TABLE turns interim table into a materialized view

/* Create the primary key index on the interim table */
create unique index mytable_pk_123456 on mytable_123456 (n);

/* Sync tables */
begin
  dbms_redefinition.sync_interim_table('QUEST', 'MYTABLE', 'MYTABLE_123456');
end;
Redefinition Script (Cont.)

/* Create the second index */
create index mytable_i_123456 on mytable_123456 (v);

/* Sync tables */
begin
   dbms_redefinition.sync_interim_table('QUEST', 'MYTABLE', 'MYTABLE_123456');
end;
/

/* Add primary key to interim table */
alter table mytable_123456 add constraint mytable_pk_123456 primary key(n);

/* Add foreign key to interim table */
alter table mytable_123456 add constraint mytable_fk_123456 foreign key (m) referencing parent_mytable disable;

/* Add foreign key referencing the interim table */
alter table child_mytable add constraint child_mytable_fk_123456 foreign key(j) referencing mytable_123456 disable;
Redefinition Script (Cont.)

/* sync tables */
begin
   dbms_redefinition.sync_interim_table('QUEST', 'MYTABLE', 'MYTABLE_123456');
end;
/

/* Finish redefinition */
begin
   dbms_redefinition.finish_redef_table('QUEST', 'MYTABLE', 'MYTABLE_123456');
end;
/

/* Drop old table (having interim name) and rename objects to their final names */
drop table mytable_123456 cascade constraints;
alter index mytable_pk_123456 rename to mytable_pk;
alter table mytable rename constraint mytable_pk_123456 to mytable_pk;
alter index mytable_i_123456 rename to mytable_i;
alter table mytable rename constraint mytable_fk_123456 to mytable_fk;
alter table child_mytable rename constraint child_mytable_fk_123456 to child_mytable_fk;
Limitations of Redefinition

<table>
<thead>
<tr>
<th>Feature not Supported</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables with LONG or LONG RAWs</td>
<td>Used widely by ERP applications</td>
</tr>
<tr>
<td>Tables with materialized views</td>
<td></td>
</tr>
<tr>
<td>Tables with materialized view logs</td>
<td></td>
</tr>
<tr>
<td>Materialized views</td>
<td></td>
</tr>
<tr>
<td>Individual partitions</td>
<td>Must reorganize entire table</td>
</tr>
<tr>
<td>User-defined data types</td>
<td></td>
</tr>
<tr>
<td>Tables with BFILES</td>
<td></td>
</tr>
<tr>
<td>Clustered tables</td>
<td></td>
</tr>
<tr>
<td>Tables in the SYS or SYSTEM schema</td>
<td></td>
</tr>
<tr>
<td>Temporary tables</td>
<td></td>
</tr>
<tr>
<td>Advanced Queuing Tables</td>
<td></td>
</tr>
<tr>
<td>Reorg in NOLOGGING mode</td>
<td>Additional load on Db during reorganization</td>
</tr>
<tr>
<td>Preserve constraint names</td>
<td>Only 9.2 and above</td>
</tr>
<tr>
<td>Rename indexes, triggers and constraints online</td>
<td>Only on 10G</td>
</tr>
<tr>
<td>Tables without primary keys</td>
<td>Only 9.2 and above</td>
</tr>
<tr>
<td>Available in Oracle versions prior to 9.0</td>
<td>7.3.4 through 8.1.7 offline only</td>
</tr>
</tbody>
</table>
Deficiencies of Redefinition

- Not Fully Online
  - Renames happen after Redefinition is complete
  - Old table is dropped before Renames
- Enables disabled referencing constraints
  - Does not preserve original state of constraints
- Does not handle invalid triggers in 9i
- Does not support LONG/LONG RAWs
- Does not support individual partitions/subpartitions
- Does not support “NoLogging” mode for interim table
- Cannot handle referential constraints to different schema
Deficiencies of Redefinition

- No Restart Capabilities
  - ABORT_REDEF_TABLE does not always clean up all objects
  - User must drop interim table and indexes
- Different Constraint Names (9i)
  - New constraints end up with "OEM$" in the name
- Prior to 9.2, tables must have primary keys
- On 9.2 and later, tables without primary keys end up with a hidden column and its index
  - Hidden index can cause redefinition to fail when space is tight
- Online redefinition/reorg not available prior to 9i
Redefinition Error Handling

- **Recoverable Errors**
  - Fix the problem
  - Edit the script to restart from point of failure

- **Unrecoverable Errors**
  - Run ABORT_REDEF_TABLE
    - Turns materialized view back into interim table
  - Manually drop interim objects
  - Restart Redefinition from the beginning
OEM-Automating Redefinition

- Different OEM in 9i and 10G
  - Some 9i problems fixed in 10G
  - Very nice, Web-based GUI in 10G
  - New 10G command REGISTER_DEPENDENT_OBJECT fixes the renaming of dependent objects
  - Issue with enabling disabled constraints still exists in 10G
  - Hidden column and its index still left on table in 10G
  - Still no support in 10G for LONGs, individual partitions, NoLogging, and referential constraints to different schema

- OEM 10G – large footprint on DB server
  - About 500MB disk space, over 100MB memory required
Live Reorganizations

- Log-based Architecture
  - Reads Oracle redo logs vs. using triggers
  - Update info stored in memory vs. in Oracle
  - Standard SQL used to post transactions to copy table

- Highly Scalable
  - Production live reorganizations of tables up to 1.2TB with millions of transactions successfully posted
How Does Live Reorg Work?

1. Reorg Data
2. Post Changes
3. Replace Original Table with Reorganized Table

The process begins with the original table and involves

- Reorg Begins
- Reorg Ends
- All Changes Have Been Posted to the Reorganized Table

Involves:
- Original Table
- Reorganized Table
- Redo Log

Changes over Time
## Features and Limitations

<table>
<thead>
<tr>
<th>Feature</th>
<th>Standard Reorg</th>
<th>LiveReorg</th>
<th>Redefinition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables with LOB’s IN ROW</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Tables with LOB’s with IN_ROW DISABLED</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>9i data types (timestamp, anydata, XML, etc.)</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Tables with LONG or LONG RAWs</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Tables with materialized views</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Materialized views</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Individual partitions</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>User-defined data types</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tables with BFILES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clustered tables</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advanced Queuing Tables</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Standard Reorg</td>
<td>LiveReorg</td>
<td>Redefinition</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>----------------</td>
<td>-----------</td>
<td>--------------</td>
</tr>
<tr>
<td>Tables in the SYS or SYSTEM schema</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Temporary tables</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reorg in NOLOGGING mode</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Preserve constraint names</td>
<td>X</td>
<td>X</td>
<td>9.2 only</td>
</tr>
<tr>
<td>Rename indexes, triggers and constraints online</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Tables without primary keys</td>
<td>X</td>
<td>X</td>
<td>9.2 only</td>
</tr>
<tr>
<td>Available in Oracle versions prior to 9.0</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Automated Restart Capability</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Interactive Job Monitor</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Create table and indexes using parallel processes</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create multiple indexes at the same time</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Overcoming Redefinition Deficiencies

- Handles all dependencies during “live” phase
  - All dependent objects retain original names and **STATE**
- Collects Analysis statistics on copy prior to switch
- Supports LONG/RAWs, including > 32K
- Ensures hidden index is accounted for during space audits
- “Live” reorgs of individual partitions/sub-partitions
- Reorganizations in “NoLogging” mode
- Handles foreign keys to different schema
- Automated restart/clean-up
- Tables without PK and no structural changes
- Supports 7.3.4 through 9.2
Redefinition Automation in 5.5

- Use Redefinition without need for installing and configuring OEM
- Safely select Redefinition candidates
- Will handle Redefinition deficiencies
- Automated restart for Redefinition
More Than Reorganizations

- Problem detection
- Object sizing
- Reorganization
- Restructuring
- Capacity Planning
- Fault prediction
- Oracle versions 7.3.4 through 9.2
- Support for 10G in Q4, 2004
Summary and Conclusions

- Redefinition is not designed for live reorganizations
- Intended for use when DB activity is light
- Several serious deficiencies still exist
- Not ready for “prime time” yet
- Investigate thoroughly before using in production environments, especially large, busy ones
Q & A
Thank you.

Additional Resources:

More information about Quest Central:
http://www.quest.com/quest_central

Download a trial of Quest Central® for Databases:
http://www.quest.com/quest_central/download.asp

Get a copy of this presentation:
http://www.quest.com/presentations

Questions?  info@quest.com