Online Operations in Oracle 12.2

New Features and Enhancements

Christian Gohmann

BASEL ▪ BERN ▪ BRUGG ▪ DÜSSELDORF ▪ FRANKFURT A.M. ▪ FREIBURG I.BR. ▪ GENEVA
HAMBURG ▪ COPENHAGEN ▪ LAUSANNE ▪ MUNICH ▪ STUTTGART ▪ VIENNA ▪ ZURICH

trivadis
makes IT easier.

About me…

Christian Gohmann

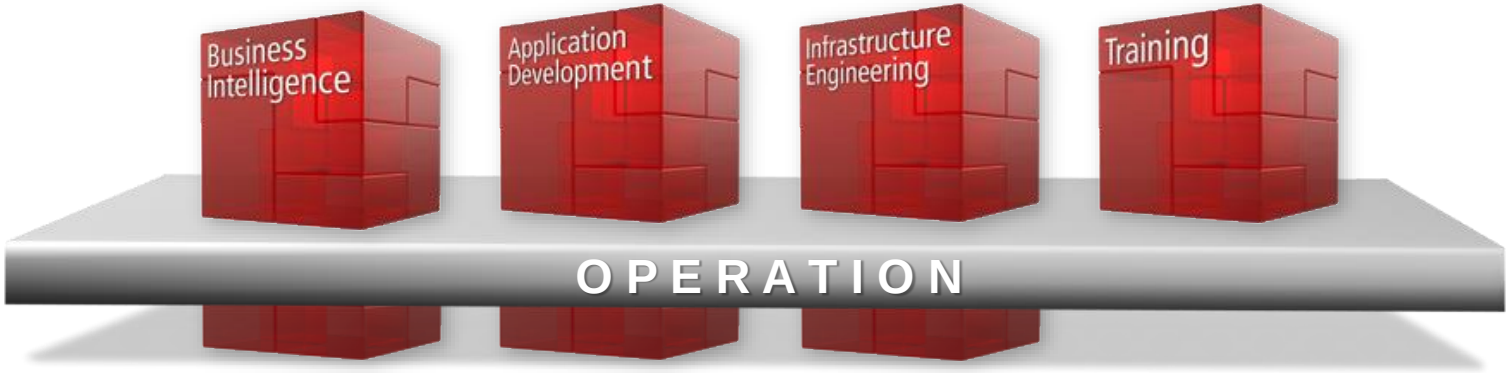
- **Senior Consultant** at **Trivadis GmbH**, Düsseldorf
- **Instructor** since 2014
  - O-AI-DBA (Oracle Architecture and Internals for DBAs)
- **Tool Owner** of TVD-Backup

- **Experiences with Oracle** since 2006 (8i – 12c)
  - Architecture, Installation & Configuration
  - High Availability Solutions (RAC, Data Guard)
  - Performance Analysis
  - Backup & Recovery (RMAN, Data Pump)
- **Blog:** http://www.christian-gohmann.de

trivadis
makes IT easier.

Our company.

Trivadis is a market leader in IT consulting, system integration, solution engineering and the provision of IT services focusing on ORACLE and Microsoft technologies
in Switzerland, Germany, Austria and Denmark. We offer our services in the following strategic business fields:



Trivadis Services takes over the interacting operation of your IT systems.

trivadis makes IT easier.

With over 600 specialists and IT experts in your region.

- 14 Trivadis branches and more than 600 employees
- 200 Service Level Agreements
- Over 4,000 training participants
- Research and development budget: CHF 5.0 million
- Financially self-supporting and sustainably profitable
- Experience from more than 1,900 projects per year at over 800 customers

trivadis makes IT easier.

Agenda

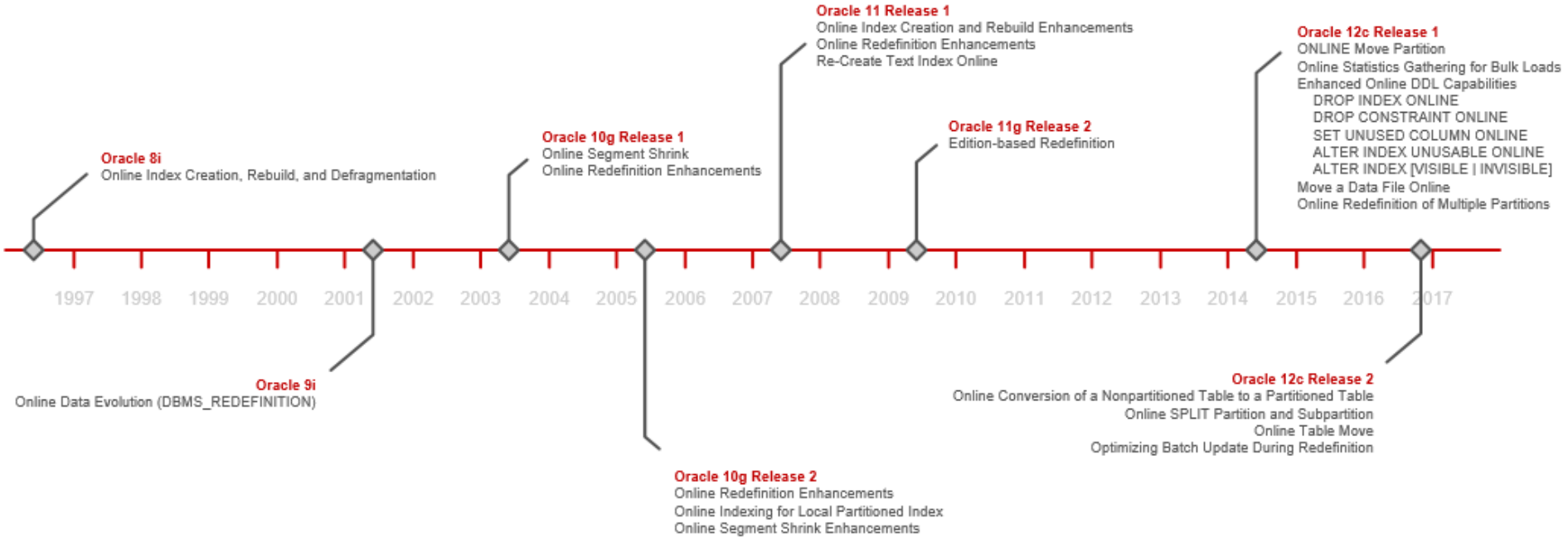
1. **Introduction**
   - History of Online Operations
   - How does it works?
2. ALTER TABLE Enhancements
   - Online Conversion of a non-partitioned Table to a partitioned Table
   - Online SPLIT Partition / Subpartition
   - Online Table Move
3. Online Redefinition Enhancements
   - Optimized Batch Update
   - Refresh Dependent Materialized Views
   - Rolling Back Online Redefinition
   - Monitoring Progress
   - Miscellaneous

trivadis makes IT easier.

Introduction

trivadis
makes IT easier.

History of Online Operations



trivadis
makes IT easier.

How does it works?

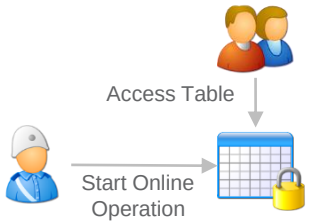
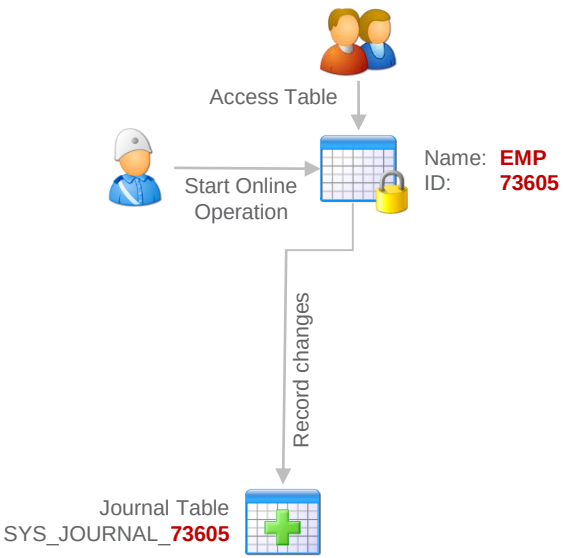


Table or Partition is **locked in ROW EXCLUSIVE mode**

```
SQL> LOCK TABLE "BOOKS" IN ROW EXCLUSIVE MODE NOWAIT;
```


trivadis makes IT easier.

How does it works?



Journal Table SYS_JOURNAL_nnnnn is created to capture every DML against the source table

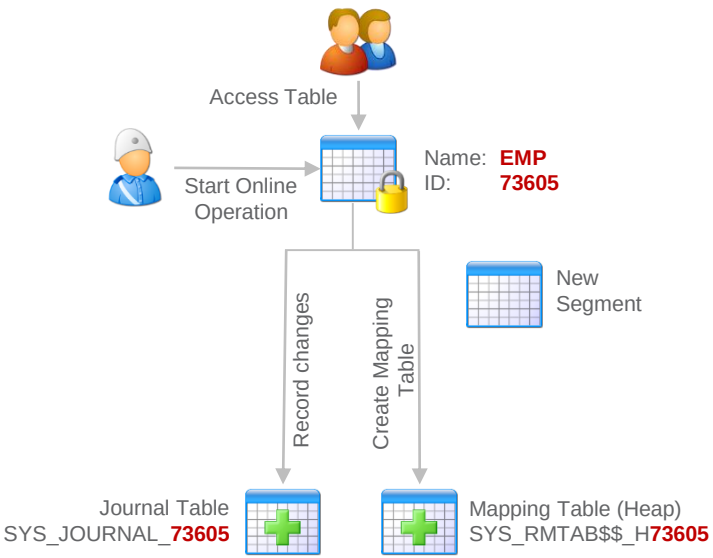
- Created as Index Organized Table
- Uses the Object ID of the source Table / Partition

```
SQL> create table "DEMO_OSP"."SYS_JOURNAL_73605" (
        rid rowid,
        opcode char(1),
        primary key(rid)
     ) organization index tablespace "DEMO_OSP_DATA";

-- Example Data
RID                 O
------------------ -
AAASDBAAFAAAACQABo D  -- Delete
AAASDBAAFAAAACTAAx U  -- Update
AAASDBAAFAAANIsAAA I  -- Insert
```


trivadis makes IT easier.

How does it works?



Create Mapping Table, which **maps the ROWID** of the **Source** Table / Partition with the **newly created** Table / Partition

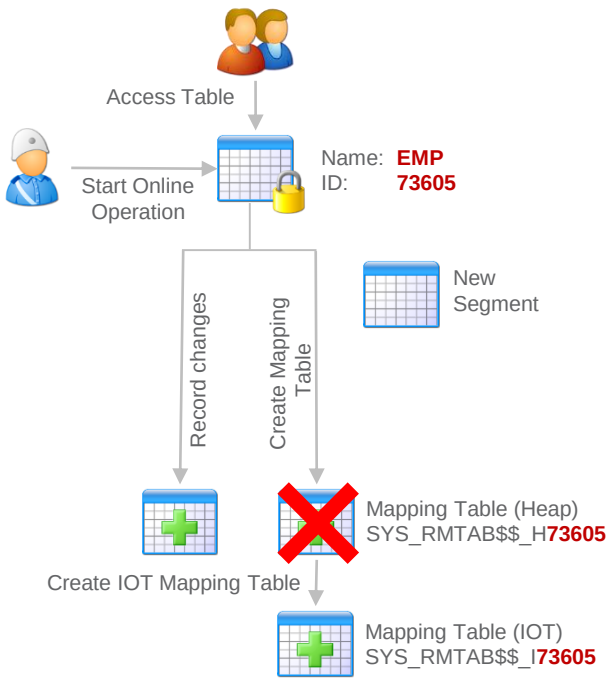
- Created as Heap Organized Table
- Undocumented ROWID_MAPPING_TABLE clause is used (not usable directly in a session)

```
SQL> create table "DEMO_OSP"."SYS_RMTAB$$_H73605" (
        src_rowid rowid not null
        tgt_rowid rowid not null)
     segment creation immediate nologging
     tablespace "DEMO_OSP_DATA" rowid_mapping_table;
```

- One row for each row of the Source Table
  - Requires 20 bytes + row overhead + Table overhead

trivadis makes IT easier.

How does it works?



An Index Organized Table version of the Mapping Table is created

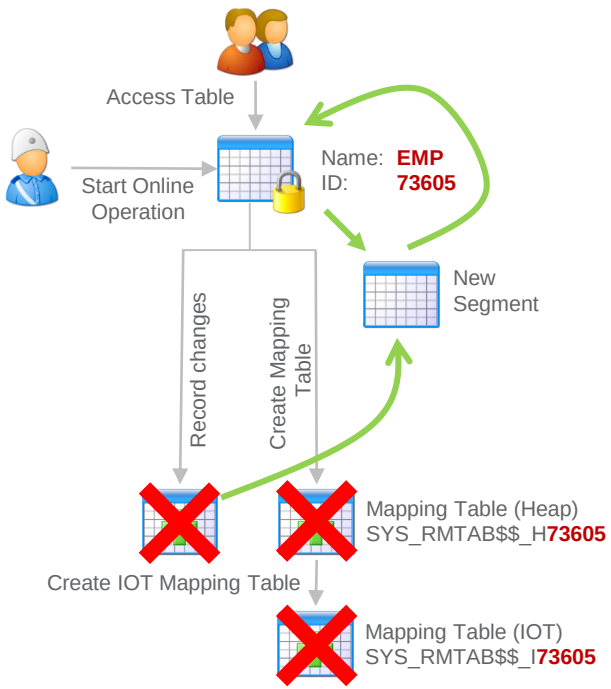
- Uses as source the Heap Organized Mapping Table
- ROWID_MAPPING_TABLE clause is used again

```
SQL> create table  "DEMO_OSP"."SYS_RMTAB$$_I73605" (
        src_rowid, tgt_rowid,
        constraint pk_SYS_RMTAB$$_I73605
        primary key (src_rowid))
        segment creation immediate organization index
        tablespace  "DEMO_OSP_DATA"  noparallel
        nologging rowid_mapping_table as (
           select src_rowid, tgt_rowid
             from "DEMO_OSP"."SYS_RMTAB$$_H73605"));
```

- Heap Organized Mapping Table is dropped after the creation

trivadis makes IT easier.

How does it works?



Newly created Table/Partition is filled with the required data

- Captured changes from the Journal Table are merged into the newly created Table / Partition
- Existing Object is updated / extended with new segment(s) after the merge (in the Data Dictionary)

Cleanup all temporarly created objects

```
SQL> drop table "DEMO_OSP"."SYS_JOURNAL_73605" purge
SQL> drop table  "DEMO_OSP"."SYS_RMTAB$$_I73605" purge
```


trivadis makes IT easier.

ALTER TABLE Enhancements

trivadis makes IT easier.

Online Conversion of a Table to a partitioned Table 1/4

- Enhancement of the ALTER TABLE command to allow **conversion of a non-partitioned tables to a partitioned table** online
  - Concurrent **DML operations are permitted**
  - Trying to run the ALTER TABLE statement against a partitioned table terminates with *ORA-14427: table does not support modification to a partitioned state DDL*
- Tables with **Domain Indexes cannot be online converted** (ORA-14427)
- Totally **two times the space** of the non-partitioned table is additionally required

trivadis makes IT easier.

Online Conversion of a Table to a partitioned Table 2/4

- **Indexes are maintained** during conversion
- Use optional **UPDATE INDEXES clause to change partition state of indexes** as well
  - Indexed columns or uniqueness of an index cannot be changed
  - TABLESPACE clause can be added to set the target location for the indexes
- Default behaviour for indexes not mentioned in UPDATE INDEXES clause

Index Type	Touched?	New Index Type	Rebuild?
Global Partitioned Index	✗		✗
Non-Prefixed Index	✓	Global Non-partitioned Index	✗
Prefixed Index	✓	Local Partitioned Index	✗
Bitmap Index	✓	Local Partitioned Index	✗

trivadis makes IT easier.

Online Conversion of a Table to a partitioned Table 3/4

- Example

```
SQL> ALTER TABLE BOOKS MODIFY
  PARTITION BY RANGE (release_date)
  ( PARTITION BOOKS_PRE_2015
       VALUES LESS THAN (TO_DATE('01.01.2015 00:00:00', 'dd.mm.yyyy HH24:MI:SS')),
    PARTITION BOOKS_2015
       VALUES LESS THAN (TO_DATE('01.01.2016 00:00:00', 'dd.mm.yyyy HH24:MI:SS')),
    PARTITION BOOKS_2016
       VALUES LESS THAN (TO_DATE('01.01.2017 00:00:00', 'dd.mm.yyyy HH24:MI:SS')),
    PARTITION BOOKS_2017
       VALUES LESS THAN (TO_DATE('01.01.2018 00:00:00', 'dd.mm.yyyy HH24:MI:SS'))
   ) UPDATE INDEXES ONLINE;
```


trivadis makes IT easier.

Online Conversion of a Table to a partitioned Table 4/4

DEMO

trivadis
makes IT easier.

Online Split Partition / Subpartition

- Partitions and Subpartitions can be split online while DML operations are ongoing
- Indexes are maintained automatically
  - No need to explicit set UPDATE INDEXES clause
- Syntax

```
SQL> ALTER TABLE BOOKS SPLIT PARTITION BOOKS_ACTION_THRILLER
   VALUES ('Action')
   INTO ( PARTITION BOOKS_ACTION,
          PARTITION BOOKS_THRILLER
) ONLINE;
```

- Totally **two times the space** of Table (Sub)partition is **additionally required**

trivadis makes IT easier.

Online Table Move

- Moving an table to a new segment can be done online while DML is still permitted
  - Until Oracle 12c Release 1 only Index-organized tables could be moved online
  - Rowids will be changed during the move
- Indexes are maintained during move when UPDATE INDEXES clause is used
  - Otherwise they will be UNSABLE after the move
- Any statistics become invalid and have to gathered again
- Syntax:

```
SQL> ALTER TABLE sh.orders MOVE ONLINE [UPDATE INDEXES];

-- Move table to a new tablespace and activate basic compression
SQL> ALTER TABLE sh.orders MOVE ONLINE COMPRESS UPDATE INDEXES TABLESPACE ts_new;
```


trivadis makes IT easier.

Online Redefinition Enhancements

trivadis makes IT easier.

Optimized Batch Updates 1/3

- Optimized performance of **bulk updates using direct mode**
  - Update is not logged in the redo logs
  - Recovery of the database would lead to logical block corruption
- Requires to run EXECUTE_UPDATE procedure of DBMS_REDEFINITION package
  - Internally well known components of Online Redefinition are used
  - Removes also fragmentation of the affected rows
  - Ensures that the update is atomic

```
SQL> EXEC DBMS_REDEFINITION.EXECUTE_UPDATE (update_stmt => 'UPDATE ... SET ...');
```

- In the case of an error, use ABORT_UPDATE to undo changes

```
SQL> EXEC DBMS_REDEFINITION.ABORT_UPDATE;
```


trivadis makes IT easier.

Optimized Batch Updates 2/3

- Restrictions
  - Only one execution of EXECUTE_UPDATE at the same time
  - DML changes of other sessions are gone after EXECUTE_UPDATE finishes
  - UPDATE Trigger are not permitted
  - Partition-extended names are not permitted
  - User-defined Types and some Oracle-supplied types are not permitted
  - No support for hidden, virtual, unused, identity and pseudo columns
  - The table cannot be an object table
  - Virtual Private Database (VPD) policies are not permitted
  - Check constraints cannot exists
  - No support of row archival feature
- **Enabling Force Logging has no effect on EXECUTE_UPDATE**

trivadis
makes IT easier.

Optimized Batch Updates 3/3

- Examples
  - #1: Table Size: 166 MB, 4 Million Rows, Update amount: 100% (One Column)
  - #2: Table Size: 416 MB, 10 Million Rows, Update amount: 100 % (One Column)
  - #3: Table Size: 416 MB, 10 Million Rows, Update amount: 25% (One Column)

#	Test	Time (s)	Divergence	Redo Size (MB)	Divergence
1	Conventional Update	36		1265	
	Optimized Batch Update	48	+ 33 %	539	- 58 %
2	Conventional Update	541		3238	
	Optimized Batch Update	700	+ 29 %	1000	- 70 %
3	Conventional Update	137		805	
	Optimized Batch Update	567	+ 313 %	997	+ 23 %

- If the table uses NOLOGGING, the generated amount of Redo is further reduced

trivadis makes IT easier.

Refresh Dependent Materialized Views 1/2

- During Online Redefintion, fast refreshable MViews can be refreshed automatically
  - Avoiding a time consuming complete refresh after the Online Redefinition finished
  - ROWID and Join MViews are not supported
- Set REFRESH_DEP_MVIEWS Parameter for procedures REDEF_TABLE or START_REDEF_TABLE to Y
  - Can be changed later using DBMS_REDEFINITION.SET_PARAM procedure

```
SQL> BEGIN
   DBMS_REDEFINITION.SET_PARAM (
      redefinition_id => 1,
      param_name => 'refresh_dep_mviews',
      param_value => 'Y'
);
```

  - Check column REFRESH_DEP_MVIEWS in DBA_REDEFINITION_STATUS

trivadis makes IT easier.

Refresh Dependent Materialized Views 2/2

- Refresh occurs during each execution of **SYNC_INTERIM_TABLE** and **FINISH_REDEF_TABLE**
  - Documentation states that refresh occurs while executing START_REDEF_TABLE too - but this is not working
  - MView is in the state NEEDS_COMPILE after FINISH_REDEF_TABLE (Bug?)
- Example

```
SQL> BEGIN
   DBMS_REDEFINITION.START_REDEF_TABLE (
      uname => USER,
      orig_table => 'DEMO_ORG_TABLE',
      int_table => 'DEMO_INT_TABLE',
      refresh_dep_mviews => 'Y'
   );
END;
/
```


trivadis makes IT easier.

Rolling Back Online Redefinition 1/3

- Roll back the changes done by the Online Redefinition
  - Every DML change will be preserved
  - Materialized View and Materialized View Log are used internally (not visible in Data Dictionary Views)
- To enable rollback functionality, **START_REDEF_TABLE** has to be started with parameter **ENABLE_ROLLBACK** set to TRUE

```
SQL> BEGIN
   DBMS_REDEFINITION.START_REDEF_TABLE (
      uname => USER, orig_table => 'DEMO_ORG_TABLE',
      int_table => 'DEMO_INT_TABLE', enable_rollback => TRUE
   );
END;
```

- Former original table will be maintained after the Online Redefinition finished
  - **SYNC_INTERIM_TABLE** can be used to force a synchronisation of the former original table

trivadis makes IT easier.

Rolling Back Online Redefinition 2/3

- Use **ROLLBACK** procedure to roll back to the original table definition

```
SQL> BEGIN
   DBMS_REDEFINITION.ROLLBACK (
      uname => USER,
      orig_table => 'DEMO_ORG_TABLE',
      int_table => 'DEMO_INT_TABLE'
   );
END;
/
```

- If the capability to roll back the changes is not required anymore, run **ABORT_ROLLBACK** to remove the internally used database objects
  - Until the objects for the rollback exists, no other Online Redefinition can be started for this table

trivadis makes IT easier.

Rolling Back Online Redefinition 3/3

DEMO

trivadis
makes IT easier.

Monitoring Progress 1/2

- New View **V$ONLINE_REDEF** introduced with 12.2.0.1
- **One status row** for each running Online Redefinition
- **Updated during execution of each step** within the redefinition process chain
  - Provides the SQL statements and its SQL ID for executed refresh statements
  - Can be joined with V$SQL and V$SQL_MONITOR to check the progress of the statement
- After the Online Redefinition is completed, the progress is not visible anymore

trivadis makes IT easier.

Monitoring Progress 2/2

- Process states over the lifetime of a online redefinition

```
SQL> SELECT original_table_name AS "ORG_TABLE",
            interim_table_name AS "INTERIM_TABLE", operation, suboperation,
            detailed_message, progress, refresh_statement_sql_id AS "SQL_ID
       FROM V$ONLINE_REDEF;

ORG_TABLE  INTERIM_TABLE  OPERATION            SUBOPERATION                  DETAILED_MESSAGE               PROGRESS          SQL_ID
---------- -------------- -------------------- ----------------------------- ------------------------------ ----------------- --------------
DEMO_OR_LR DEMO_OR_LR_NEW START_REDEF_TABLE     create/verify materialized... create/verify MV logs          step 2 out of 7
DEMO_OR_LR DEMO_OR_LR_NEW COPY_TABLE_DEPENDENTS copy the indexes                                             step 3 out of 7
DEMO_OR_LR DEMO_OR_LR_NEW COPY_TABLE_DEPENDENTS copy the constraints                                         step 4 out of 7
DEMO_OR_LR DEMO_OR_LR_NEW SYNC_REDEF_TABLE      refresh the materialized view                                step 2 out of 3   3c4n9a8hx9mvm
DEMO_OR_LR DEMO_OR_LR_NEW FINISH_REDEF_TABLE    validate register dependent   validate ... dependent indexes step 2 out of 10
DEMO_OR_LR DEMO_OR_LR_NEW FINISH_REDEF_TABLE    validate register dependent   validate ... constraints       step 2 out of 10
```


trivadis makes IT easier.

Miscellaneous

Support for Tables with BFILE columns

- No ORA-12088 error is thrown during *_REDEF_TABLE procedures

Locking Enhancements

- **Exclusive Lock** during execution of FINISH_REDEF_TABLE is **not required** anymore

Restart Online Redefinition after a failure

- Check column **RESTARTABLE** in **DBA_REDEFINITION_STATUS**, if the operation can be restarted or not
- E.g. when the tablespace runs out of space

trivadis makes IT easier.

Oracle 12c Release 2 Database New Features Guide
http://docs.oracle.com/database/122/NEWFT/toc.htm

trivadis
makes IT easier.

Christian Gohmann
Senior Consultant

Tel. +49-211-58 6664 702
christian.gohmann@trivadis.com

trivadis
makes IT easier.