Configuring Change Delivery

by Ahmed Baraka

Introduction to Oracle Data Guard
In this lecture, we are going to talk about the basic concepts of Oracle Data Guard

Objectives

By the end of this lecture, you should be able to:

- Perform the procedure to create a classic Replicat process
- Describe the self-describing trail files
- Start and stop a Replicat process group
- Delete a Replicat process group

Configuring Oracle GoldenGate Roadmap

1. Prepare the environment
2. Configure Initial Load
3. Configure the Change Synchronization:
   a) Configure the Extract (Change Capture)
   b) Configure the Replicate (Delivery)

About the Replicate Process

- Reads data out of the GoldenGate trails and applies it on the target database
- Data filtering by table, row, or operation is supported
- Data transformation is supported
- Can run in nonintegrated (classic), integrated, or coordinated modes

About the `CheckpointTable`

- Optional in the target database but recommended
- It is advisable to create it in GoldenGate schema
- Can be specified in the GLOBALS file and/or parameter file
- Contains the following checkpoints:
  - Startup checkpoint
  - Current checkpoint
- Checkpoints are written:
  - When a transaction is completed by the Replicat
  - At regular Intervals

About Self-Describing Trail Files

- Introduced in Oracle GoldenGate 12.2
- Metadata of the source data structure encapsulated in each of the trail files (source definitions file not needed)
- Metadata typs:
  - Database Definition Record (DDR)
  - Table Definition Record (TDR)
- User `USE_TRAILDEFS` or `NO_USE_TRAILDEFS` in the `GLOBALS` parameter file

Setting Up Change Delivery (Replicat)

1. Create a checkpoint table
2. Edit Replicat parameters
3. Add the Replicat process
4. Create Data-definitions File (if needed)
5. Start the Replicat process

Create Replicate Process Example

1. Create a checkpoint table

```
GGSCI> DBLogin UserID <myuser>, Password <mypasswd>
GGSCI> Add CheckpointTable rcheckpt
```

Create Replicate Process Example (cont..)

2. Edit Replicat parameters

```
GGSCI> Edit Params rhr2
```

Create Replicate Process Example (cont..)

- Replicat parameter file sample (Identical structure, older than 12.2 trail files):

```
Replicat rhr2
UserID ogguser@mysid, Password mypasswd
AssumeTargetDefs
DiscardFile ./dirrpt/rhr2.dsc, Append, MEGABYTES 80
Map HR.EMPLOYEES, Target HR.UKEMPLOYEES;
Map HR.DEPARTMENTS, Target HR.UKDEPARTMENTS;
Map HR.JOBS, Target HR.UKJOBS;
```

Create Replicate Process Example (cont..)

- Replicat parameter file sample (Dissimilar structure, older than 12.2 trail files):

```
Replicat rhr2
UserID ogguser@mysid, Password mypasswd
SOURCEDEFS /app/ggs/dirdef/rhr2.def
DiscardFile ./dirrpt/rhr2.dsc, Append, MEGABYTES 80
Map HR.EMPLOYEES, Target HR.UKEMPLOYEES;
Map HR.DEPARTMENTS, Target HR.UKDEPARTMENTS;
Map HR.JOBS, Target HR.UKJOBS;
```

Create Replicate Process Example (cont..)

- Replicat parameter file sample (self-describing trail files):

```
Replicat rhr2
UserID ogguser@mysid, Password mypasswd
DDL include all
DiscardFile ./dirrpt/rhr2.dsc, Append, MEGABYTES 80
Map HR.EMPLOYEES, Target HR.UKEMPLOYEES;
Map HR.DEPARTMENTS, Target HR.UKDEPARTMENTS;
Map HR.JOBS, Target HR.UKJOBS;
```

Create Replicate Process Example (cont..)

3. Add the Replicat process

```
Add Replicat rhr2, ExtTrail /ggs/dirdat/d
ADD Replicat rhr2, Exttrail ./dirdat/d,
CHECKPOINTTABLE GGS.GG_CHKPT
```

Create Replicate Process Example (cont..)

4. Create Data-definitions file (if needed):
   a) Create a temporary parameter file in the source database:

```
DEFSFILE /app/ggs/dirdef/rhr2.def
USERID hr, PASSWORD hrpassword
TABLE HR.EMP_LEAVES;
```

   b) Generate the data-definitions file:

```
defgen paramfile ./dirprm/temp.prm
```

   c) Copy the data-definitions file to the target database

Start the Replicate Process Example

5. Start the Replicat processes

```
GGSCI> Start Replicat rhr2 AfterCSN <csn>
GGSCI> Start Replicat rhr2
GGSCI> Info Replicate rhr2
```

Stopping and Deleting the Replicate

- To stop a Replicat process:

```
GGSCI> STOP REPLICAT <group_name>
```

- To delete a Replicat process:

```
# stop the Replicate

# Login to the database:
DBLOGIN {USERID user, PASSWORD password}

# Delete the process group:
DELETE REPLICAT <group_name>
```

Handling Collisions

- If collisions cannot be avoided, you must handle them
- You can used the `HandleCollisions` parameter:
  - overwrite existing records
  - Ignore missing records
- After all changes have been applied, turn it off

Summary

In this lecture, you should have learnt how to:

- Perform the procedure to create a classic Replicat process
- Describe the self-describing trail files
- Start and stop a Replicat process group
- Delete a Replicat process group