Handling Errors

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:

- Set rules to handle errors caught by Oracle GoldenGate processes
- Use Exceptions table to save the caught exceptions
- Set rules to handle TCP/IP Errors

Configuring Oracle GoldenGate CDR

- Make the required column values available to Extract
- Configure the parameter files for conflict resolution
- Configure the parameter files for error handling

Configure the Parameter Files for Error Handling

- Use the `REPERROR` parameter to control how Replicat responds to errors
- Assign rules for handling errors that cannot be resolved by CDR, or for errors that you do not want to handle through CDR
- You can create an exceptions table
- Map the exceptions data to the exceptions table
- **Note**: DDL Errors are handled by DDLERROR

About `REPERROR` Parameter

- Can be used globally and/or on specific table (`MAP`)
- Can handle transaction-level or record-level errors

```
REPERROR ( <error to handle>, <response> )
```

```
REPERROR {
( {DEFAULT | DEFAULT2 | SQL_error | user_defined_error},
{ABEND | DISCARD | EXCEPTION | IGNORE | RETRYOP [MAXRETRIES n] |
TRANSABORT [, MAXRETRIES] [, DELAYSECS n | DELAYCSECS n] |
TRANSDISCARD | TRANSEXCEPTION
} ) | RESET }
```

REPERROR Error Specification Options

Error Specification Option	Description
DEFAULT	Sets the default response of the **REPERROR**
DEFAULT2	Sets the response when the **DEFAULT** option is set to **EXCEPTION**
SQL_error	Specific SQL Error number
user_defined_error	A user-defined error specified with the **RAISEERROR** option of a **FILTER** clause within a **MAP** statement.

REPERROR Error Response Options

Error Response	Description
Abend	Roll back the transaction and terminates processing abnormally
Discard	Write the operation to the discard file and continue processing
Exception	Work with MAPEXCEPTION option of MAP
Ignore	Ignore the error
RetryOP	Retry the offending operation
TransAbort	Abort the transaction and reposition to the beginning of it
TransDiscard	Abort the transaction and write its records in the DISCARD file
TransException	Work with MAPEXCEPTION or EXCEPTIONSONLY in MAP stmnt
Reset	Remove error-handling rules specified in previous REPERROR

Using REPERROR Globally Examples

```
REPERROR (DEFAULT, ABEND)
REPERROR (-1, IGNORE)
```

```
REPERROR (DEFAULT, EXCEPTION)
REPERROR (DEFAULT2, ABEND)
MAP hr.dept, TARGET hr.dept;
MAP hr.emp, TARGET hr.emp;
INSERTALLRECORDS
MAP hr.emp, TARGET hr.emp_exception, EXCEPTIONSONLY, COLMAP ...
```



```
REPERROR (-1, IGNORE)
MAP hr.dept, TARGET hr.dept;
REPERROR RESET
MAP hr.emp, TARGET hr.emp;
```

Using `REPERROR` Globally Examples (cont)

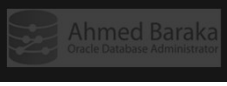
```
REPERROR (DEFAULT, ABEND)
REPERROR (1409, TransDiscard)
```

```
REPERROR (DEFAULT, ABEND)
REPERROR (1409, TransException)
MAP src, TARGET tgt, MapException (...);
```

Using `REPERROR` Globally and in MAP Statement Examples

```
REPERROR (DEFAULT, Abend)
REPERROR (409, TransDiscard)
MAP src1, TARGET tgt1, REPERROR (409, Ignore);
MAP src2, TARGET tgt2, REPERROR (403, Ignore);
```

```
REPERROR DEFAULT ABEND
REPERROR 1401 TRANSDISCARD.
MAP src, TARGET tgt, REPERROR(400 TRANSDISCARD);
```



Using Exceptions Table

- Is used with an exceptions `MAP` statement
- Conflicting operations are saved in it
- At minimum, an exceptions table should contain the same columns as the target table
- Additional information could be included
- An exceptions `MAP` statement contains: `INSERTALLRECORDS`, either `EXCEPTIONSONLY` or `MAPEXCEPTION`, and `COLMAP` (optional)

Using `EXCEPTIONSONLY` or `MAPEXCEPTION`

- **`EXCEPTIONSONLY`**
  - Two MAP statements are used
  - Valid for one pair of explicitly named and mapped source and target tables
- **`MAPEXCEPTION`**
  - Within the source table MAP statement itself
  - Accepts the names of the source and target tables in the MAP statement to be wildcarded
  - Optionally use the `COLMAP` clause with the `USEDEFAULTS`

Using Exceptions Table Example 1

- The source and target exceptions columns are identical
- No other columns in the exceptions table

```
REPERROR (DEFAULT, DISCARD)
MAP src, TARGET tgt_exception, EXCEPTIONSONLY, INSERTALLRECORDS;
```

Using Exceptions Table Example 2

- Additional columns in the exceptions table

```
REPERROR (DEFAULT, DISCARD)
MAP src, TARGET tgt_exception, EXCEPTIONSONLY, INSERTALLRECORDS
SQLEXEC (id qry, query 'select .. from tgt where rowid = :p1',
PARAMS(p1 = rowid )),
COLMAP ( USEDEFAULTS, res_date = @DATENOW (),
         optype = @GETENV ('LASTERR', 'OPTYPE'),
         ..
         name_before = @BEFORE (name),
         phone_before = @BEFORE (phone),
         ..
         name_current = qry.name,
         phone_current = qry.phone, ..)
```

Using Exceptions Table Example 3

- Using the `MAPEXCEPTION` clause:

```
MAP src.trx*, TARGET tgt.*,
MAPEXCEPTION (TARGET fin.trxexceptions,
 INSERTALLRECORDS,
 COLMAP (USEDEFAULTS,
         ...
         ACCT_NO = ACCT_NO,
         OPTYPE = @GETENV ('LASTERR', 'OPTYPE'),
         DBERR = @GETENV ('LASTERR', 'DBERRNUM'),
         DBERRMSG = @GETENV ('LASTERR', 'DBERRMSG')
         )
);
```

Handling TCP/IP Errors

- Reponses to TCP/IP errors are configured in `<OGG_HOME>\TCPERRS`

Column	Description
Error	The TCP/IP error to handle
Response	Response to the error: `RETRY` or `ABEND`
Delay	Period of time to wait before reconnect
Max Retries	Maximum number of reconnect attempts before aborting

Summary

In this lecture, you should have learnt how to do the following:

- Set rules to handle errors caught by Oracle GoldenGate processes
- Use Exceptions table to save the caught exceptions
- Set rules to handle TCP/IP Errors