Using Macros and User Tokens

Practice Overview

In this practice you will perform the following:

- Develop a macro library and use it in the process parameter file
- Use the user tokens in your Oracle GoldenGate configuration

Using Macros and User Tokens

A. Develop Macro library and use it

In this section of the practice, you will build a macro library and use it in the Extract and Replicat parameter files.

1. Stop the Extract and the Replicat, if they are running.

2. In ~~both~~ the source ~~and target~~ system~~s~~, create a macro library file and add the code in it as follows:

Macro libraries should have `mac` extension.

```
sh vi /u01/app/oracle/product/ogg/dirprm/header.mac
```

```
REPORTROLLOVER AT 03:00 ON sunday
DISCARDROLLOVER AT 03:00 ON sunday
```

3. Open the Extract parameter file and remove the settings of the two parameters. Include the macro library in the beginning of the parameter file (right after the Extract parameter).

```
~~REPORTROLLOVER AT 03:00 ON sunday~~
~~DISCARDROLLOVER AT 03:00 ON sunday~~
INCLUDE /u01/app/oracle/product/ogg/dirprm/header.mac
```

4. Perform the same previous steps on the Replicat parameter file.

5. Start the Extract and Replicat processes and verify that they are in RUNNING status.

6. Verify that the parameters in the macro library apply in the processes.

```
-- in ggsrv1:
send extract esrv1 getparaminfo

-- in ggsrv2:
send replicat rsrv2 getparaminfo
```

B. Use User Tokens

In this section of the practice, you will use the user tokens to gather information about the source system and send it through to the target system. The Replicat will store this information as part of populating the transaction history table.

7. Stop the Extract and the Replicat.

8. In db2, ~~alter~~ add two columns to the history table, as follows:~~.~~

```
ALTER TABLE HRTRG.EMP_HISTORY ADD ( OSUSERNAME VARCHAR2(~~30~~20), HOSTNAME
VARCHAR2(30));
```

9. Open the Extract parameter file. Modify the TABLE parameter file of the EMPLOYEES table as follows:~~.~~

```
Table HR.EMPLOYEES,~~ FETCHCOLS (DEPARTMENT_ID,SALARY),~~
 TOKENS ( TK-OSUSER = @GETENV ('GGENVIRONMENT' , 'OSUSERNAME'),
          TK-HOST = @GETENV('GGENVIRONMENT' , 'HOSTNAME'));
```

10. Open the Replicat parameter file. Modify the MAP parameter which maps the EMPLOYEES table to the EMP_HISTORY table as follows.

```
Map HR.EMPLOYEES, Target HRTRG.EMP_HISTORY, INSERTALLRECORDS,
 ColMap (USEDEFAULTS,
         OP_TYPE = @GetEnv('GGHEADER', 'OPTYPE'),
         TRAN_TIME = @GetEnv('GGHEADER', 'COMMITTIMESTAMP'),
         BEFORE_AFTER = @GETENV ('GGHEADER', 'BEFOREAFTERINDICATOR'),
         OSUSERNAME = @TOKEN('TK-OSUSER'),
         HOSTNAME = @TOKEN('TK-HOST')
);
~~Map HR.EMPLOYEES, Target HRTRG.EMP_HISTORY,INSERTALLRECORDS,~~
~~ ColMap (EMPLOYEE_ID=EMPLOYEE_ID,~~
~~         ASALARY=SALARY,~~
~~         ACOMMISSION_PCT = COMMISSION_PCT,~~
~~         BSALARY = @BEFORE(SALARY),~~
~~         BCOMMISSION_PCT = @BEFORE(COMMISSION_PCT),~~
~~         OP_TYPE = @GetEnv('GGHEADER', 'OPTYPE'),~~
~~         TRAN_TIME = @GetEnv('GGHEADER', 'COMMITTIMESTAMP'),~~
~~         OSUSERNAME = @TOKEN('TK-OSUSER'),~~
~~         HOSTNAME = @TOKEN('TK-HOST')~~
~~);~~
```

~~22.~~11. Start the Extract and the Replicat

~~23.~~12. In db1, update an employee record.

```
UPDATE HR.EMPLOYEES SET JOB_ID='IT_PROG' WHERE EMPLOYEE_ID=107;
COMMIT;
```

~~24.~~13. In db2, verify the history table has been updated.

```
SELECT SALARY, to_char(TRAN_TIME,'DD-MON-RR HH24:MI:SS') TRAN_TIME, OP_TYPE,
BEFORE_AFTER , OSUSERNAME, HOSTNAME
from HRTRG.EMP_HISTORY
WHERE EMPLOYEE_ID=107 ORDER BY TRAN_TIME DESC;~~SELECT ASALARY, BSALARY,
to_char(TRAN_TIME,'DD-MON-RR HH24:MI:SS') TRAN_TIME, OSUSERNAME, HOSTNAME from
HRTRG.EMP_HISTORY WHERE EMPLOYEE_ID=107 ORDER BY TRAN_TIME DESC;~~
```

C. Summary

M~~For data filtering and manipulation in Oracle GoldenGate, you can use a wide range of built-in functions to implement complex scenarios. In this practice, you gained experience on using the `IF` function, some date and string functions, and the `SQLEXEC` function.~~

- ~~Which function to use depends on your business needs. The technical details of using all the functions can be obtained from the documentation “Reference for Oracle GoldenGate for Windows and UNIX”.~~acro library is the proper way to share the same code in more than one parameter file.
- You can create user tokens in the Extract side so that you pass specific information from the source system to the target system. Replicat can read the user tokens and process them.

By the end of this practice, following are the code in each process parameter file:

```
Extract esrv1
INCLUDE /u01/app/oracle/product/ogg/dirprm/header.mac
USERID ogg, PASSWORD oracle
ExtTrail ./dirdat/es
Table HR.JOB_HISTORY;
Table HR.EMPLOYEES,
 TOKENS ( TK-OSUSER = @GETENV ('GGENVIRONMENT' , 'OSUSERNAME'),
 TK-HOST = @GETENV('GGENVIRONMENT' , 'HOSTNAME'));
Table HR.JOBS;
Table HR.DEPARTMENTS;
Table HR.LOCATIONS;
Table HR.REGIONS;
Table HR.SAMPLE;

Extract psrv1
RmtHost ggsrv2, MgrPort 7810
RmtTrail ./dirdat/rt
Passthru
Table HR.*;

Replicat rsrv2
INCLUDE /u01/app/oracle/product/ogg/dirprm/header.mac
DiscardFile ./dirrpt/rsrv2.dsc, Purge
USERID ogg, PASSWORD oracle
Map HR.EMPLOYEES, Target HRTRG.EMPLOYEES,
 SQLEXEC (id GET_TITLE, QUERY ' SELECT JOB_TITLE FROM HRTRG.JOBS
       WHERE JOB_ID = :V_JOB_ID ',
       PARAMS (V_JOB_ID = JOB_ID)),
 COLMAP (USEDEFAULTS, IT_JOB_FLAG = @IF (@STREQ (JOB_ID, 'IT_PROG'), 'Y', 'N'),
 WORKING_DAYS = @DATEDIFF ('DD', HIRE_DATE, @DATENOW ()),
 TITLE = @GETVAL(GET_TITLE.JOB_TITLE))
         );
Map HR.EMPLOYEES, Target HRTRG.EMP_HISTORY, INSERTALLRECORDS,
ColMap (USEDEFAULTS,
    OP_TYPE = @GetEnv('GGHEADER', 'OPTYPE'),
    TRAN_TIME = @GetEnv('GGHEADER', 'COMMITTIMESTAMP'),
    BEFORE_AFTER = @GETENV ('GGHEADER', 'BEFOREAFTERINDICATOR'),
    OSUSERNAME = @TOKEN('TK-OSUSER'),
    HOSTNAME = @TOKEN('TK-HOST')
);
Map HR.*, Target HRTRG.*;
```