

Using Event Marker System

Practice Overview

In this practice you will implement an example of using Oracle GoldenGate Event Maker System. You will create an events table. When a specific record is inserted into the table, the Replicat process will stop. If you insert another specific record, the Extract process will end.



Ahmed Baraka
Oracle Database Administrator

Implementing Event Marker System

A. Configure GoldenGate for Event Marker System

In this section of the practice, you will take the steps to configure GoldenGate for Event Marker System.

1. Stop the Extract and the Replicat, if they are running.
2. On both, the source and the target databases, create a new event table.

```
sqlplus hr/oracle
CREATE TABLE HR.EVENTS ( EVENT VARCHAR2(30), EVENT_TIME DATE DEFAULT SYSDATE PRIMARY
KEY);

sqlplus hrtrg/oracle
CREATE TABLE HRTRG.EVENTS ( EVENT VARCHAR2(30), EVENT_TIME DATE DEFAULT SYSDATE
PRIMARY KEY);
```

3. In the source system, add the supplemental logging data of the table.

This step is actually optional in your case because the schema-level supplemental logging is enabled for the HR schema. This step is documented here just as a reminder that supplemental logging is required.

```
DBLogin UserID ogg@db1, Password oracle
ADD TRANDATA HR.EVENTS
```

4. Add the following TABLE statements to the Extract parameter file.

You need to have two TABLE parameters for the same source table. One parameter will take action when specific record is received. If the record in process does not apply the condition defined in the first parameter, the second parameter will take effect. Because the second parameter in this example does not have any condition, the record will just pass through as is to the Data Pump process.

Notice that the order of those TABLE parameters is important.

```
TABLE HR.EVENTS, FILTER (@STREQ (EVENT, 'STOP EXTRACT' )), EVENTACTIONS (IGNORE
TRANS,STOP);
TABLE HR.EVENTS;
```

5. Add the following parameter ALLOWDUPTARGETMAP to the Extract parameter file, anywhere after the USERID and before the TABLE parameters.

This parameter is needed when you use the TABLE parameter with the same source more than once in the parameter file. If this parameter is not there, the Extract will use only the first TABLE statement of that source.

```
ALLOWDUPTARGETMAP
```

6. Add the following MAP statements to the Replicat parameter file.

```
MAP HR.EVENTS, TARGET HRTRG.EVENTS, FILTER (@STREQ (EVENT, 'STOP REPLICAT' )), 
EVENTACTIONS (IGNORE TRANS,STOP);
```

7. Start the Extract and Replicat processes.

B. Test the Event Marker System configuration

8. Stop the Replicat by inserting the following record in the source table.

```
INSERT INTO HR.EVENTS VALUES ('STOP REPLICAT', SYSDATE);  
COMMIT;
```

9. Verify that the Replicat has been stopped.

10. Stop the Extract by inserting the following record in the source table.

```
INSERT INTO HR.EVENTS VALUES ('STOP EXTRACT', SYSDATE);  
COMMIT;
```

11. Verify that the Extract has been stopped.

Summary

Event Marker System allows you to send commands to Oracle GoldenGate processes from SQL. In this practice, you learnt how to send STOP commands to Replicat and Extract processes from the source database using SQL commands.



Ahmed Baraka
Oracle Database Administrator

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
ALLOWDUPTARGETMAP
ExtTrail ./dirdat/es
Table HR.JOB_HISTORY;
Table HR.EMPLOYEES,
TOKENS ( TK-OSUSER = @GETENV ('GGENVIRONMENT' , 'OSUSERNAME'),
          TK-HOST = @GETENV('GGENVIRONMENT' , 'HOSTNAME'));
Table HR.JOB;
Table HR.DEPARTMENTS;
Table HR.LOCATIONS;
Table HR.REGIONS;
Table HR.SAMPLE;
TABLE HR.EVENTS, FILTER (@STREQ (EVENT, 'STOP EXTRACT' )), EVENTACTIONS (IGNORE
TRANS,STOP);
TABLE HR.EVENTS;
```

```
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.JOB
      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 = @DATEDIFFD', 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.EVENTS, TARGET HRTRG.EVENTS, FILTER (@STREQ (EVENT, 'STOP REPLICAT' )),  
EVENTACTIONS (IGNORE TRANTOP);  
Map HR.* , Target HRTRG.*;
```



Ahmed Baraka
Oracle Database Administrator