

## Oracle GoldenGate - Mappings

This page discusses the mapping of schema, table and column names in Oracle GoldenGate.

The page is based on Oracle GoldenGate version 11.2.1.0.1. For a basic configuration I used two Linux VMs (OEL5U6) running single instance Oracle 11.2.0.3 databases. I created both databases using DBCA.

This configuration uses following hosts and databases.

	Source	Target
Hostname	vm4	vm5
Database Name	NORTH	SOUTH

The configuration includes the following on both nodes:

- Creation of a GoldenGate schema owner called GG01.
- Specification of GGSHEMA as GG01 in GoldenGate parameters
- Creation of GOLDENGATE tablespace which is default tablespace for GG01

The GoldenGate process names are:

	Source	Target
Extract	ex1	-
Data Pump	dp1	-
Replicat	-	rep1

### Mapping Schema Names

GoldenGate can be configured to map all tables from a schema in the source database to a different schema in the target database. The mapping is performed by the replicat process on the target database; the trails are unaffected.

In the following example we will map all tables in the US03 schema in the source database to the US01 schema in the target database.

Shut down the EX1 extract process on the source server

Shut down the DP1 extract process on the source server

Shut down the REP1 replicat process on the target server

In the US03 schema in the source database create table T51 as follows:

```
CREATE TABLE t51
(
  c1 NUMBER PRIMARY KEY,
  c2 VARCHAR2(30),
  c3 NUMBER,
  c4 NUMBER
);
```

In the US01 schema in the target database create the same table:

```
CREATE TABLE t51
(
  c1 NUMBER PRIMARY KEY,
  c2 VARCHAR2(30),
  c3 NUMBER,
```

```
c4 NUMBER
);
```

In GGSCI on the source server add supplemental logging for the T51 table in the US03 schema:

```
[oracle@vm4]$ cd /home/oracle/goldengate
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> DBLOGIN USERID us03 PASSWORD us03
GGSCI (vm4) 2> ADD TRANDATA t51
Logging of supplemental redo data enabled for table US03.T51.
```

Modify the parameter files for each process as follows:

The extract process has the following parameters:

```
EXTRACT ex1
USERID gg01, PASSWORD gg01
EXTTRAIL /home/oracle/goldengate/dirdat/ex
TABLE US03.*;
```

The data pump process has the following parameters:

```
EXTRACT dp1
USERID gg01, PASSWORD gg01
RMTHOST vm5, MGRPORT 7809
RMTTRAIL /home/oracle/goldengate/dirdat/rt
TABLE US03.*;
```

The replicat process has the following parameters:

```
REPLICAT rep1
USERID gg01, PASSWORD gg01
ASSUMETARGETDEFS
DISCARDFILE /home/oracle/goldengate/discards, PURGE
MAP US03.* TARGET US01.*;
```

Start the EX1 extract process on the source server

Start the DP1 data pump process on the source server

Start the REP1 replicat process on the target server

In the US03 schema in the source database add some rows to the T51 table.

```
INSERT INTO t51 VALUES (1,'Alpha',10,100);
INSERT INTO t51 VALUES (2,'Beta',20,200);
INSERT INTO t51 VALUES (3,'Gamma',30,300);

COMMIT;
```

In the US01 schema in the target database check that the new rows have been replicated:

```
[oracle@vm5]$ sqlplus us01/us01
SQL> SELECT * FROM t51;

      C1 C2                                C3          C4
-----

```

1	Alpha	10	100
2	Beta	20	200
3	Gamma	30	300

On the target server, the report file for the replicat process includes details of the mapping. For example:

```
Wildcard MAP resolved (entry US03.*):
  MAP "US03"."T51" TARGET US02."T51";
Using following columns in default map by name:
  C1, C2, C3, C4
Using the following key columns for target table US02.T51: C1.
```

### Mapping Table Names

GoldenGate can be configured to map all table names between the source database and the target database. The table names can be in the same or in different schemas. The mapping is performed by the replicat process on the target database; the trails are unaffected.

In the following example we will map table US03.T52 in the source database to table US01.T53 in the target database. The tables have identical columns.

Shut down the EX1 extract process on the source server

Shut down the DP1 extract process on the source server

Shut down the REP1 replicat process on the target server

In the US03 schema in the source database create table T52 as follows:

```
CREATE TABLE t52
(
  c1 NUMBER PRIMARY KEY,
  c2 VARCHAR2(30),
  c3 NUMBER,
  c4 NUMBER
);
```

In the US01 schema in the target database create table T53 as follows:

```
CREATE TABLE t53
(
  c1 NUMBER PRIMARY KEY,
  c2 VARCHAR2(30),
  c3 NUMBER,
  c4 NUMBER
);
```

In GGSCI on the source server add supplemental logging for the table US03.T52:

```
[oracle@vm4]$ cd /home/oracle/goldengate
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> DBLOGIN USERID us03 PASSWORD us03
GGSCI (vm4) 2> ADD TRANDATA t52
Logging of supplemental redo data enabled for table US03.T52.
```

Modify the parameter files for each process as follows:

The extract process has the following parameters:

```
EXTRACT ex1
USERID gg01, PASSWORD gg01
EXTTRAIL /home/oracle/goldengate/dirdat/ex
TABLE US03.*;
```

The data pump process has the following parameters:

```
EXTRACT dp1
USERID gg01, PASSWORD gg01
RMTHOST vm5, MGRPORT 7809
RMTTRAIL /home/oracle/goldengate/dirdat/rt
TABLE US03.*;
```

The replicat process has the following parameters:

```
REPLICAT repl
USERID gg01, PASSWORD gg01
ASSUMETARGETDEFS
DISCARDFILE /home/oracle/goldengate/discards, PURGE
MAP US03.T52 TARGET US01.T53;
```

Start the EX1 extract process on the source server

Start the DP1 data pump process on the source server

Start the REP1 replicat process on the target server

In the US03 schema in the source database add some rows to the T52 table.

```
INSERT INTO t52 VALUES (1,'Alpha',10,100);
INSERT INTO t52 VALUES (2,'Beta',20,200);
INSERT INTO t52 VALUES (3,'Gamma',30,300);

COMMIT;
```

In the US01 schema in the target database check that the new rows have been replicated to table T53:

```
[oracle@vm5]$ sqlplus us01/us01

SQL> SELECT * FROM t53;

   C1 C2                C3      C4
-----
    1 Alpha             10      100
    2 Beta              20      200
    3 Gamma            30      300
```

On the target server, the report file for the replicat process includes details of the mapping. For example:

```
MAP resolved (entry US03.T52):
  MAP "US03"."T52" TARGET US01.T53;
Using following columns in default map by name:
  C1, C2, C3, C4
Using the following key columns for target table US01.T53: C1.
```

## Mapping Column Names

GoldenGate can be configured to map column names between tables in the source database and the target database. The tables can be in the same or in different schemas. The mapping is performed by the replicat process on the target database; the trails are unaffected.

Column mapping is more complicated than schema name or table name mapping. It requires a definition file to be generated for the source table using the *defgen* utility. This definition file must be copied to the target server for use as a reference by the replicat process.

In the following example we will map table US03.T54 in the source database to table US01.T55 in the target database. The column mappings are as follows:

T54	T55
c1	col1
c2	col3
c3	col4
c4	col2

Shut down the EX1 extract process on the source server

Shut down the DP1 extract process on the source server

Shut down the REP1 replicat process on the target server

In the US03 schema in the source database create table T54 as follows:

```
CREATE TABLE t54
(
  c1 NUMBER PRIMARY KEY,
  c2 VARCHAR2(30),
  c3 NUMBER,
  c4 NUMBER
);
```

In the US01 schema in the target database create table T55 as follows:

```
CREATE TABLE t55
(
  col1 NUMBER PRIMARY KEY,
  col2 NUMBER,
  col3 VARCHAR2(30),
  col4 NUMBER
);
```

Note that column names and order differ between tables T54 and T55.

In GGSCI on the source server add supplemental logging for the table US03.T54:

```
[oracle@vm4]$ cd /home/oracle/goldengate
[oracle@vm4]$ ggsci

GGSCI (vm4) 1> DBLOGIN USERID us03 PASSWORD us03

GGSCI (vm4) 2> ADD TRANDATA t54

Logging of supplemental redo data enabled for table US03.T54.
```

On the source database create a parameter file for the *defgen* utility. This can be any text file. Alternatively it can be created using ggsci so that it is stored with the other GoldenGate parameter files:

```
[oracle@vm4]$ cd /home/oracle/goldengate
[oracle@vm4]$ ggsci

GGSCI (vm4) 1> EDIT PARAMS defgen1
```

Add the following entries to the parameter file:

```
DEFSFILE ./dirdef/defgen1.def
USERID us03 PASSWORD us03
TABLE us03.t54;
```

The DEFSFILE parameter specifies the location of the definition file that will be generated by *defgen*. In this case we are only generating a definition for table US03.T54. However, the definition file can contain definitions for multiple files.

Generate the definition file using the *defgen* utility:

```
[oracle@vm4]$ cd /home/oracle/goldengate
[oracle@vm4]$ ./defgen paramfile /home/oracle/goldengate/dirprm/defgen1.prm
```

The above command generated the following output:

```
*****
      Oracle GoldenGate Table Definition Generator for Oracle
      Version 11.2.1.0.1 OGGCORE_11.2.1.0.1_PLATFORMS_120423.0230
      Linux, x64, 64bit (optimized), Oracle 11g on Apr 23 2012 05:08:19

Copyright (C) 1995, 2012, Oracle and/or its affiliates. All rights reserved.

      Starting at 2013-04-15 06:06:57
*****

Operating System Version:
Linux
Version #1 SMP Tue Jan 18 20:11:49 EST 2011, Release 2.6.32-100.26.2.el5
Node: vm4.juliandyke.com
Machine: x86_64

      soft limit      hard limit
Address Space Size  :      unlimited      unlimited
Heap Size          :      unlimited      unlimited
File Size          :      unlimited      unlimited
CPU Time           :      unlimited      unlimited

Process id: 15048

*****
**          Running with the following parameters          **
*****
DEFSFILE ./dirdef/defgen1.def
USERID us03 PASSWORD ****
TABLE us03.t54;
Retrieving definition for US03.T54

Definitions generated for 1 table in ./dirdef/defgen1.def
```

In this case the *defgen1.def* file contains the following definitions:

```
*+- Defgen version 2.0, Encoding UTF-8
*
* Definitions created/modified 2013-04-15 06:06
*
* Field descriptions for each column entry:
*
* 1 Name
* 2 Data Type
* 3 External Length
* 4 Fetch Offset
* 5 Scale
* 6 Level
```

```

*      7      Null
*      8      Bump if Odd
*      9      Internal Length
*     10      Binary Length
*     11      Table Length
*     12      Most Significant DT
*     13      Least Significant DT
*     14      High Precision
*     15      Low Precision
*     16      Elementary Item
*     17      Occurs
*     18      Key Column
*     19      Sub Data Type
*
Database type: ORACLE
Character set ID: windows-1252
National character set ID: UTF-16
Locale: neutral
Case sensitivity: 14 14 14 14 14 14 14 14 14 14 14 14 14 11 14 14 14
*
Definition for table US03.T54
Record length: 204
Syskey: 0
Columns: 4
C1   64   50       0  0  0  1  0       50   50   50  0  0  0  0  1   0  1  2
C2   64   30     56  0  0  1  0       30   30   0  0  0  0  0  1   0  0  0
C3   64   50     92  0  0  1  0       50   50  50  0  0  0  0  1   0  0  2
C4   64   50    148  0  0  1  0       50   50  50  0  0  0  0  1   0  0  2
End of definition

```

The definitions appear to be a little strange for an Oracle database. However, remember that GoldenGate supports replication between databases from different vendors.

Copy the definition file to the target server. For example:

```

[oracle@vm4]$ cd /home/oracle/goldengate/dirdef
[oracle@vm4]$ scp defgen1.def vm5:`pwd`

```

Modify the parameter files for each process as follows:

The extract process has the following parameters:

```

EXTRACT ex1
USERID gg01, PASSWORD gg01
EXTTRAIL /home/oracle/goldengate/dirdat/ex
TABLE US03.*;

```

The data pump process has the following parameters:

```

EXTRACT dp1
USERID gg01, PASSWORD gg01
RMTHOST vm5, MGRPORT 7809
RMTTRAIL /home/oracle/goldengate/dirdat/rt
TABLE US03.*;

```

The replicat process has the following parameters:

```

REPLICAT repl
USERID gg01, PASSWORD gg01
SOURCEDEFS ./dirdef/defgen1.def
DISCARDFILE /home/oracle/goldengate/discards, PURGE
MAP US03.T54 TARGET US01.T55, &
COLMAP (col1=c1, col3=c2, col4=c3, col2=c4);

```

In the replicat parameter file the SOURCEDEFS parameter is used to specify the location of the definition file. The COLMAP clause of the MAP parameter is used to specify non-default mappings between columns. Note that it is possible for the target table to have fewer columns than the source table.

The syntax for each mapping in the COLMAP clause is:

The replicat process will abend if this clause is incorrect.

Start the EX1 extract process on the source server

Start the DP1 data pump process on the source server

Start the REP1 replicat process on the target server

In the US03 schema in the source database add some rows to the T54 table.

```
INSERT INTO t54 VALUES (1, 'Alpha', 10, 100);
INSERT INTO t54 VALUES (2, 'Beta', 20, 200);
INSERT INTO t54 VALUES (3, 'Gamma', 30, 300);
COMMIT;
```

In the US01 schema in the target database check that the new rows have been replicated to table T55:

```
[oracle@vm5]$ sqlplus us01/us01
SQL> SELECT * FROM t55;
```

COL1	COL2	COL3	COL4
1	100	Alpha	10
2	200	Beta	20
3	300	Gamma	30

On the target server, the report file for the replicat process includes details of the mapping. For example:

```
MAP resolved (entry US03.T54):
  MAP "US03"."T54" TARGET US01.T55, COLMAP (col1=c1, col3=c2, col4=c3, col2=c4);
Using the following key columns for target table US01.T55: COL1.
```

Let us consider another example. The source table is unchanged. However, we now want to replicate this table to US01.T56 in the target database. Table US01.T56 has the following definition:

```
CREATE TABLE t56
(
  c1 NUMBER PRIMARY KEY,
  c2 VARCHAR2(30),
  c3 NUMBER
);
```

The column mappings are as follows:

T54	T56
c1	c1
c2	c2
c4	c3



Columns C1 and C2 in the source table, map to columns with the same names in the target table. Column C4 in the source table maps to column C3 in the target table. Column C3 is not mapped

As the source table has not been altered, the definition file on the target server is unchanged

The replicat parameter file contains the following:

```
REPLICAT repl
USERID gg01, PASSWORD gg01
SOURCEDEFS ./dirdef/defgen1.def
DISCARDFILE /home/oracle/goldengate/discards, PURGE
MAP US03.T54 TARGET US01.T56, &
COLMAP (USEDEFAULTS, c3=c4);
```

The USEDEFAULTS keyword specifies that column names are identical between the two tables except where a column mapping has been explicitly defined. In this case, therefore we only need to specify the mapping between column C4 in the source table and column C3 in the target table. In the US03 schema in the source database add some rows to the T54 table.

```
INSERT INTO t54 VALUES (1, 'Alpha', 10, 100);
INSERT INTO t54 VALUES (2, 'Beta', 20, 200);
INSERT INTO t54 VALUES (3, 'Gamma', 30, 300);
COMMIT;
```

In the US01 schema in the target database check that the new rows have been replicated to table T56:

```
[oracle@vm5]$ sqlplus us01/us01
```

```
SQL> SELECT * FROM t56;
```

C1	C2	C3
1	Alpha	100
2	Beta	200
3	Gamma	300

On the target server, the report file for the replicat process includes details of the mapping:

```
MAP resolved (entry US03.T54):
  MAP "US03"."T54" TARGET US01.T56, COLMAP (USEDEFAULTS, c3=c4);
Using the following default columns with matching names:
  C1=C1, C2=C2, C3=C3
Using the following key columns for target table US01.T56: C1.
```