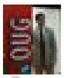
GET_DDL and SET_TRANSFORM_PARAM in DBMS_METADATA



Deiby Gomez / 11.25.2015 at 6:40am

In our daily tasks as a Database Administrator it's common to see requests to get the DDL sentence from any object. Everybody knows that there is the package DBMS_METADATA that can be used for this kind of requests. But most of the DBAs uses DBMS_METADATA.GET_DDL with the basic parameters like ("type of object", "object_name", "owner"), few DBAs knows that we can use a transformation for this function. There are many transformations that we can use, as per Oracle Documentation, The basic transformation are the following:

Object Type	Name	Meaning
All objects	**PRETTY**	If TRUE, format the output with indentation and line feeds. Defaults to TRUE.
	SQLTERMINATOR	If TRUE, append a SQL terminator (; or /) to each DDL statement. Defaults to FALSE.
TABLE	**SEGMENT_ATTRIBUTES**	If TRUE, emit segment attributes (physical attributes, storage attributes, tablespace, logging). Defaults to TRUE.
	STORAGE	If TRUE, emit storage clause. (Ignored if SEGMENT_ATTRIBUTES is FALSE.) Defaults to TRUE.
	TABLESPACE	If TRUE, emit tablespace. (Ignored if SEGMENT_ATTRIBUTES is FALSE.) Defaults to TRUE.
TABLE	**CONSTRAINTS**	If TRUE, emit all non-referential table constraints. Defaults to TRUE.
	REF_CONSTRAINTS	If TRUE, emit all referential constraints (foreign key and scoped refs). Defaults to TRUE.
	CONSTRAINTS_AS_ALTER	If TRUE, emit table constraints as separate ALTER TABLE (and, if necessary, CREATE INDEX) statements. If FALSE, specify table constraints as part of the CREATE TABLE statement. Defaults to FALSE. Requires that CONSTRAINTS be TRUE.
	OID	If TRUE, emit the OID clause for object tables. Defaults to FALSE.
	SIZE_BYTE_KEYWORD	If TRUE, emit the BYTE keyword as part of the size specification of CHAR and VARCHAR2 columns that use byte semantics. If FALSE, omit the keyword. Defaults toFALSE.
INDEX	**SEGMENT_ATTRIBUTES**	If TRUE, emit segment attributes (physical attributes, storage attributes, tablespace, logging). Defaults to TRUE.
	STORAGE	If TRUE, emit storage clause. (Ignored if SEGMENT_ATTRIBUTES is FALSE.) Defaults to TRUE.
	TABLESPACE	If TRUE, emit tablespace. (Ignored if SEGMENT_ATTRIBUTES is FALSE.) Defaults to TRUE.
TYPE	**SPECIFICATION**	If TRUE, emit the type specification. Defaults to TRUE.
	BODY	If TRUE, emit the type body. Defaults to TRUE.
PACKAGE	**SPECIFICATION**	If TRUE, emit the package specification. Defaults to TRUE.

		to TRUE.
	BODY	If TRUE, emit the package body. Defaults to TRUE.
VIEW	**FORCE**	If TRUE, use the FORCE keyword in the CREATE VIEW statement. Defaults to TRUE.
All objects	**DEFAULT**	Calling SET_TRANSFORM_PARAM with this parameter set to TRUE has the effect of resetting all parameters for the transform to their default values. Setting this FALSE has no effect. There is no default.
	INHERIT	If TRUE, inherits session-level parameters. Defaults to FALSE. If an application calls ADD_TRANSFORM to add the DDL transform, then by default the only transform parameters that apply are those explicitly set for that transform handle. This has no effect if the transform handle is the session transform handle.

As you can see there are some interesting transformations, for example "pretty", or the one that add a ";" at the end of the sentences. What can you tell met about "CONSTRAINTS_AS_ALTER" which present you the constraints as "alter table" sentences?, very good, right?.
Well it is not the same when you read the concept than seeing examples. I will show you some of those transformations for TABLES objects.

Firstable, the following sentences will be called as "The original" one, after every transformation I will revert it back to the original state. This means that every transformation presented here are performed on the original version of the DDL.

Original Version:

```
SQL> select DBMS_METADATA.GET_DDL('TABLE','TABLE1','DGOMEZ') ddl from dual;

DDL
--------------------------------------------------------------------------------------------
--
CREATE TABLE "DGOMEZ"."TABLE1"
( "ID" NUMBER,
"VALUE1" VARCHAR2(20),
"VALUE2" VARCHAR2(20),
CONSTRAINT "PK_CONS" PRIMARY KEY ("ID")
USING INDEX (CREATE INDEX "DGOMEZ"."INDEX1" ON "DGOMEZ"."TABLE1" ("ID")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
TABLESPACE "TABLESPACE1" ) ENABLE,
CONSTRAINT "FK_CONS" FOREIGN KEY ("VALUE1")
REFERENCES "DGOMEZ"."TABLE2" ("VALUE1") ENABLE
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "TABLESPACE1"
```

Pretty:

```
SQL> execute
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'PRETTY',false);

PL/SQL procedure successfully completed.

SQL> select DBMS_METADATA.GET_DDL('TABLE','TABLE1','DGOMEZ') ddl from dual;


DDL
--------------------------------------------------------------------------------------------
--
CREATE TABLE "DGOMEZ"."TABLE1" ("ID" NUMBER, "VALUE1" VARCHAR2(20), "VALUE2" VARCHAR2(20),
CONSTRAINT "PK_CONS" PRIMARY KEY ("ID") USING INDEX (CREATE INDEX "DGOMEZ"."INDEX1" ON
"DGOMEZ"."TABLE1" ("ID") PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS TABLESPACE
"TABLESPACE1" ) ENABLE, CONSTRAINT "FK_CONS" FOREIGN KEY ("VALUE1") REFERENCES
"DGOMEZ"."TABLE2"
("VALUE1") ENABLE) SEGMENT CREATION DEFERRED PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING TABLESPACE "TABLESPACE1"
```

SQLTERMINATOR

```
SQL> execute
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'SQLTERMINATOR',true);

PL/SQL procedure successfully completed.

SQL> select DBMS_METADATA.GET_DDL('TABLE','TABLE1','DGOMEZ') ddl from dual;

DDL
-------------------------------------------------------------------------------------------
--
CREATE TABLE "DGOMEZ"."TABLE1"
( "ID" NUMBER,
"VALUE1" VARCHAR2(20),
"VALUE2" VARCHAR2(20),
CONSTRAINT "PK_CONS" PRIMARY KEY ("ID")
USING INDEX (CREATE INDEX "DGOMEZ"."INDEX1" ON "DGOMEZ"."TABLE1" ("ID")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
TABLESPACE "TABLESPACE1" ) ENABLE,
CONSTRAINT "FK_CONS" FOREIGN KEY ("VALUE1")
REFERENCES "DGOMEZ"."TABLE2" ("VALUE1") ENABLE
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "TABLESPACE1" ;
```

SEGMENT_ATTRIBUTES

```
SQL> execute
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'SEGMENT_ATTRIBUTES',false);

PL/SQL procedure successfully completed.

SQL> select DBMS_METADATA.GET_DDL('TABLE','TABLE1','DGOMEZ') ddl from dual;

CREATE TABLE "DGOMEZ"."TABLE1"
( "ID" NUMBER,
"VALUE1" VARCHAR2(20),
"VALUE2" VARCHAR2(20),
CONSTRAINT "PK_CONS" PRIMARY KEY ("ID")
USING INDEX (CREATE INDEX "DGOMEZ"."INDEX1" ON "DGOMEZ"."TABLE1" ("ID")
) ENABLE,
CONSTRAINT "FK_CONS" FOREIGN KEY ("VALUE1")
REFERENCES "DGOMEZ"."TABLE2" ("VALUE1") ENABLE
)
```

STORAGE

```
SQL> execute
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'STORAGE',false);
```

```
PL/SQL procedure successfully completed.

SQL> select DBMS_METADATA.GET_DDL('TABLE','TABLE1','DGOMEZ') ddl from dual;

DDL
--------------------------------------------------------------------------------------------
--
CREATE TABLE "DGOMEZ"."TABLE1"
( "ID" NUMBER,
"VALUE1" VARCHAR2(20),
"VALUE2" VARCHAR2(20),
CONSTRAINT "PK_CONS" PRIMARY KEY ("ID")
USING INDEX (CREATE INDEX "DGOMEZ"."INDEX1" ON "DGOMEZ"."TABLE1" ("ID")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
TABLESPACE "TABLESPACE1" ) ENABLE,
CONSTRAINT "FK_CONS" FOREIGN KEY ("VALUE1")
REFERENCES "DGOMEZ"."TABLE2" ("VALUE1") ENABLE
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "TABLESPACE1"
```

TABLESPACE

```
SQL> execute
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'TABLESPACE',false);

PL/SQL procedure successfully completed.

SQL> select DBMS_METADATA.GET_DDL('TABLE','TABLE1','DGOMEZ') ddl from dual;

DDL
--------------------------------------------------------------------------------------------
--
CREATE TABLE "DGOMEZ"."TABLE1"
( "ID" NUMBER,
"VALUE1" VARCHAR2(20),
"VALUE2" VARCHAR2(20),
CONSTRAINT "PK_CONS" PRIMARY KEY ("ID")
USING INDEX (CREATE INDEX "DGOMEZ"."INDEX1" ON "DGOMEZ"."TABLE1" ("ID")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS ) ENABLE,
CONSTRAINT "FK_CONS" FOREIGN KEY ("VALUE1")
REFERENCES "DGOMEZ"."TABLE2" ("VALUE1") ENABLE
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGIN
```

CONSTRAINTS

```
SQL> execute
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'CONSTRAINTS',false);

PL/SQL procedure successfully completed.

SQL> select DBMS_METADATA.GET_DDL('TABLE','TABLE1','DGOMEZ') ddl from dual;

DDL
--------------------------------------------------------------------------------------------
-
CREATE TABLE "DGOMEZ"."TABLE1"
( "ID" NUMBER,
"VALUE1" VARCHAR2(20),
"VALUE2" VARCHAR2(20),
CONSTRAINT "FK_CONS" FOREIGN KEY ("VALUE1")
REFERENCES "DGOMEZ"."TABLE2" ("VALUE1") ENABLE
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "TABLESPACE1"
```

REF_CONSTRAINTS

```
SQL> execute
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'REF_CONSTRAINTS',false);

PL/SQL procedure successfully completed.

SQL> select DBMS_METADATA.GET_DDL('TABLE','TABLE1','DGOMEZ') ddl from dual;

DDL
-------------------------------------------------------------------------------------------
-
CREATE TABLE "DGOMEZ"."TABLE1"
( "ID" NUMBER,
"VALUE1" VARCHAR2(20),
"VALUE2" VARCHAR2(20),
CONSTRAINT "PK_CONS" PRIMARY KEY ("ID")
USING INDEX (CREATE INDEX "DGOMEZ"."INDEX1" ON "DGOMEZ"."TABLE1" ("ID")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
TABLESPACE "TABLESPACE1" ) ENABLE
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "TABLESPACE1"
```

CONSTRAINTS_AS_ALTER

```
SQL> execute
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'CONSTRAINTS_AS_ALTER',tr
ue);

PL/SQL procedure successfully completed.

SQL> select DBMS_METADATA.GET_DDL('TABLE','TABLE1','DGOMEZ') ddl from dual;

DDL
-------------------------------------------------------------------------------------------
--
CREATE TABLE "DGOMEZ"."TABLE1"
( "ID" NUMBER,
"VALUE1" VARCHAR2(20),
"VALUE2" VARCHAR2(20)
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "TABLESPACE1"
ALTER TABLE "DGOMEZ"."TABLE1" ADD CONSTRAINT "PK_CONS" PRIMARY KEY ("ID")
USING INDEX (CREATE INDEX "DGOMEZ"."INDEX1" ON "DGOMEZ"."TABLE1" ("ID")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
TABLESPACE "TABLESPACE1" ) ENABLE
```

OID

```
SQL> execute DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'OID',true);

PL/SQL procedure successfully completed.

SQL> select DBMS_METADATA.GET_DDL('TABLE','TABLE1','DGOMEZ') ddl from dual;

DDL
-------------------------------------------------------------------------------------------
-
CREATE TABLE "DGOMEZ"."TABLE1"
( "ID" NUMBER,
"VALUE1" VARCHAR2(20),
"VALUE2" VARCHAR2(20)
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "TABLESPACE1"
ALTER TABLE "DGOMEZ"."TABLE1" ADD CONSTRAINT "PK_CONS" PRIMARY KEY ("ID")
USING INDEX (CREATE INDEX "DGOMEZ"."INDEX1" ON "DGOMEZ"."TABLE1" ("ID")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
TABLESPACE "TABLESPACE1" ) ENABLE
```

SIZE_BYTE_KEYWORD

```
SQL> execute
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'SIZE_BYTE_KEYWORD',true)
;

PL/SQL procedure successfully completed.

SQL> select DBMS_METADATA.GET_DDL('TABLE','TABLE1','DGOMEZ') ddl from dual;

DDL
-------------------------------------------------------------------------------------------
-
CREATE TABLE "DGOMEZ"."TABLE1"
( "ID" NUMBER,
"VALUE1" VARCHAR2(20 BYTE),
"VALUE2" VARCHAR2(20 BYTE)
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "TABLESPACE1"
ALTER TABLE "DGOMEZ"."TABLE1" ADD CONSTRAINT "PK_CONS" PRIMARY KEY ("ID")
USING INDEX (CREATE INDEX "DGOMEZ"."INDEX1" ON "DGOMEZ"."TABLE1" ("ID")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
TABLESPACE "TABLESPACE1" ) ENABLE
```