|  |  |  |
| --- | --- | --- |
| **using DBMS\_METADATA To Get The DDL For Objects (Doc ID 188838.1)** | [IMG_256](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=337666588718551%26id=188838.1%26_adf.ctrl-state=15lbvnujsb_114%20/o%20To%20Bottom)  [To Bottom](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=337666588718551&id=188838.1&_adf.ctrl-state=15lbvnujsb_114 \\o To Bottom) | IMG_257 |

IMG_258

|  |
| --- |
| **APPLIES TO:**  PL/SQL - Version 9.0.1.0 and later  Oracle Database - Enterprise Edition - Version 9.0.1.0 to 11.1.0.7 [Release 9.0.1 to 11.1]  Information in this document applies to any platform.  \*\*\* Checked for relevance on 22-Mar-2016 \*\*\*  **GOAL**   The purpose of this document is to illustrate the usage of **dbms\_metadata** to generate the DDL for objects.  **SOLUTION**  **Oracle 9 and Oracle 9i:**  The DBMS\_METADATA package is a powerful tool for obtaining the complete definition of a schema object. It enables you to obtain all of the attributes  of an object in one pass. The object is described as DDL that can be used to (re)create it.  The GET\_DDL function is used to fetch the DDL for all tables in the current schema, filtering out nested tables and overflow segments.  The SET\_TRANSFORM\_PARAM (with the handle value equal to DBMS\_METADATA.SESSION\_TRANSFORM meaning "for the current session") is used to  specify that storage clauses are not to be returned in the SQL DDL.    Afterwards, the session-level transform parameters are reset to their defaults.  Once set, transform parameter values remain in effect until specifically reset  to their defaults.    Note: Please note that you would be required to run catmeta.sql for the creation of the views related to DBMS\_METADATA.  This Script is available under $ORACLE\_HOME/rdbms/admin directory.    For E.g if you have created a table :  create table idx3\_tab (     name  varchar2(30),     id    number,     addr  varchar2(100),     phone varchar2(30)) tablespace users;  And then wanted to generate the table creation script, run the following  query:  select dbms\_metadata.get\_ddl('TABLE','IDX3\_TAB') from dual;  The output would be:    CREATE TABLE "SCOTT"."IDX3\_TAB"     (    "NAME" VARCHAR2(30),          "ID" NUMBER,          "ADDR" VARCHAR2(100),          "PHONE" VARCHAR2(30)     ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  PCTINCREASE 0    FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT) TABLESPACE "USERS"  To get the create table definition without the storage clause you could do as follows:  EXECUTE DBMS\_METADATA.SET\_TRANSFORM\_PARAM(DBMS\_METADATA.SESSION\_TRANSFORM,'STORAGE',false);  The output should be PL/SQL procedure successfully completed.  And then if you run  set long 100000  select dbms\_metadata.get\_ddl('TABLE','IDX3\_TAB') from dual;  would return:  CREATE TABLE "SCOTT"."IDX3\_TAB"    (    "NAME" VARCHAR2(30),         "ID" NUMBER,         "ADDR" VARCHAR2(100),         "PHONE" VARCHAR2(30)    ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 LOGGING TABLESPACE "USERS"  Another example would be:  create type person as object (    name varchar2(20),    age number);  /  create type v0 as varray(5) of person;  /  create type n1 as table of v0;  /  create type n2 as object (n2\_c1 n1);  /  create table tab11 (    c1 n2)  nested table c1.n2\_c1 store as tab11\_c1\_n1 (    varray column\_value store as lob tab11\_c1\_v1)  RETURN AS LOCATOR;    set long 100000  select dbms\_metadata.get\_ddl('TABLE','TAB11') from dual;  would show an output like   CREATE TABLE "SCOTT"."TAB11"    (    "C1" "SCOTT"."N2"    ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 LOGGING TABLESPACE "USERS"  NESTED TABLE "C1"."N2\_C1" STORE AS "TAB11\_C1\_N1"  (PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 LOGGING  VARRAY "COLUMN\_VALUE" STORE AS LOB "TAB11\_C1\_V1"   (ENABLE STORAGE IN ROW CHUNK 4096 PCTVERSION 10   CACHE )) RETURN AS LOCATOR  This tool would avoid the work of writing a select statement which would combine data dictionary views to get the desired output.  If we want the definition of all the objects in the database we could get the definition from the export dump.    **Oracle 10.1.0.X:**  The information above is essentially the same on 10g, However, in 10g the  file to be run is catdp.sql as it is part of Data Pump.  If you run catmeta in 10g, you will get errors :   ERROR:      ORA-06502: PL/SQL: numeric or value error      ORA-31605: the following was returned from LpxXSLResetAllVars in routine      kuxslResetParams:      LPX-1: NULL pointer  **Oracle 10.2.0.X and in 11.1.0.X:**  The information above is essentially the same on 10.2 and 11g. However, in 10.2 and 11g the file to be run is catdph.sql as it is part of Data Pump.  **REFERENCES**  [NOTE:1016836.6](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=188838.1&id=1016836.6) - How to Capture Table Constraints onto a SQL Script |

set line 200

set pagesize 0

set long 99999

set feedback off

set echo off

select dbms\_metadata.get\_ddl('TABLESPACE','CMS1TEST') from dual;

select dbms\_metadata.get\_ddl('TABLESPACE','CMS') from dual;

select dbms\_metadata.get\_ddl(''TABLE','IDX3\_TAB','schema\_name') from dual;

DBMS\_METADATA.GET\_DDL (

object\_type     IN VARCHAR2,

name            IN VARCHAR2,

schema          IN VARCHAR2 DEFAULT NULL,

version         IN VARCHAR2 DEFAULT 'COMPATIBLE',

model           IN VARCHAR2 DEFAULT 'ORACLE',

transform       IN VARCHAR2 DEFAULT 'DDL')

RETURN CLOB;