DZ_SDOTXT

- Build ID: 6
- Change Set: c6d7d954c3137fd1a076a3a3b49548f9d3bc1128

Utilities for the conversion and inspection of Oracle Spatial objects as text.

Generally there are few reasons for you to want to manifest Oracle Spatial objects as SQL text. So you should only be using this code if you need to generate an example for an OTN posting or Oracle SR, or if you are exchanging a very modest amount of data with a colleague who has limited access to Oracle. Overwhelmingly the proper way to exchange Oracle data is via datapump.

See the DZ_TESTDATA project as an example of what this module can do.

Summary

DZ_SDOTXT

Functions

dz_sdotxt_main.sdo2sql	Utility to convert an Oracle Spatial object into text format.
dz_sdotxt_main.sdo2sql_nvl	Utility to convert an Oracle Spatial object into text format with an NVL option to return a given output when input is NULL.
dz_sdotxt_main.blob2sql	Utility to convert a blob into textual hex usable as a right side assignment in SQL or PLSQL.
dz_sdotxt_main.blob2plsql	Utility to convert a blob into textual hex usable as a series of DBMS_LOB statements which can be then be combined and used as a bind variable in PLSQL dynamic SQL statememts.
dz_sdotxt_main. sdo2geomblob	Utility to convert a geometry into the secret bar-delimited CLOB format which is then compressed with UTL_COMPRESS into a BLOB.
dz_sdotxt_main. geomblob2sdo	Utility to convert a blob of compressed bar-delimited geometry back into MDSYS.SDO_GEOMETRY.
dz_sdotxt_main. dump_string_endpoints	Utility to convert to text the ordinates of the endpoints of a given linestring.
dz_sdotxt_main. dump_string_endpoints	Utility to convert to text the ordinates of the endpoints of two linestrings.
dz_sdotxt_main. dump_sdo_subelements	Utility to convert to text the component objects of a geometry collection.
dz_sdotxt_main. dump_single_point_ordinate	Utility to extract a given ordinate from within a MDSYS.SDO_GEOMETRY.
dz_sdotxt_main.dump_mbr	Utility to convert to text the MBR surrounding a given geometry object.
dz_sdotxt_main. label_ordinates	Utility to converts all vertices in a geometry into a pipelined flow of points labelled by vertice number.
dz_sdotxt_main. label_measures	Utility to converts all vertices in a geometry into a pipelined flow of points labelled by LRS measure value.
dz_sdotxt_main.break_2499	Utility to take a very long sdo text representation and add linefeeds to meet the sqlplus 2499 character length restrictions.

Functions

dz_sdotxt_main.sdo2sql

Utility to convert an Oracle Spatial object into text format.

Parameters

p_input	object to convert to text format
p_2d_flag	optional TRUE/FALSE flag to remove any third or fourth dimensions on geometries before text conversion.

p_output_srid	optional srid to transform geometries to before text conversion.
p_pretty_print	optional pretty print indent value

Returns

CLOB text value representing an Oracle Spatial object.

Notes

- Input objects may include MDSYS.SDO_GEOMETRY, MDSYS.SDO_GEOMETRY_ARRAY, MDSYS.SDO_POINT_TYPE, MDSYS.SDO_ELEM_INFO_ARRAY, MDSYS.SDO_ORDINATE_ARRAY, MDSYS.SDO_GEORASTER, MDSYS.SDO_RASTER and MDSYS.SDO_DIM_ARRAY.
- Note that Oracle is not PostgreSQL or other database systems where large objects can be comfortably dumped to text. Attempting to dump a 300,000 vertice geometry is going to fail, attempting to dump a 70 gig raster rdt table is going to fail, attempting to feed more than a few meg of generated text data back through sqlplus is also going to fail. These utilities are provided for very modest purposes primarily to inspect the details of small example spatial objects or package up one or two smaller sized objects for transport via text to a collaborator. In all situations the use of Oracle datapump to import and export spatial data is the way to go.

dz_sdotxt_main.sdo2sql_nvl

Utility to convert an Oracle Spatial object into text format with an NVL option to return a given output when input is NULL.

Parameters

p_input	MDSYS.SDO_GEOMETRY to convert to text format.
p_is_null	value to return if input object is NULL.
p_2d_flag	optional TRUE/FALSE flag to remove any third or fourth dimensions on geometry before text conversion.
p_output_srid	optional srid to transform geometry to before text conversion.
p_pretty_print	optional pretty print indent value

Returns

CLOB text value representing an Oracle Spatial object.

dz_sdotxt_main.blob2sql

Utility to convert a blob into textual hex usable as a right side assignment in SQL or PLSQL. In order to use the results in SQL this is greatly limited to 4000 characters and in PLSQL to 32676 characters which when dumping a BLOB is pretty limiting. Use blob2plsql for a more scaleable work-around.

Parameters

p_input	BLOB to convert to sql text.

Returns

CLOB result.

dz_sdotxt_main.blob2plsql

Utility to convert a blob into textual hex usable as a series of DBMS_LOB statements which can be then be combined and used as a bind variable in PLSQL dynamic SQL statememts.

Parameters

p_input	BLOB to convert to sql text.
p_lob_name	the lob variable name in the statements, default is dz_lob. Use different names if you are dumping multiple blobs.
p_delim_value	the delimiter to place at the end of each statement, the default is a line feed to make sqlplus happy. Set to NULL if you want no linefeeds.

Returns

CLOB result.

dz_sdotxt_main.sdo2geomblob

Utility to convert a geometry into the secret bar-delimited CLOB format which is then compressed with UTL_COMPRESS into a BLOB.

This utility has a very specific use case of squeezing down a larger geometry into a blob which can be expressed as sql text using blob2plsql and thus shared in a OTN forum or other situation whereby a collaborator may have limited access to Oracle datapump. I can think of no scenarios where this would be appropriate for production or other true ETL tasks. The proper way to share Oracle data is via datapump.

The easiest way to convert the blob created by this procedure back into MDSYS.SDO_GEOMETRY is via geomblob2sdo. Neither of these functions are that overly complex. Note that in many cases it may be easier to just convert your geometry to a WKB BLOB via MDSYS.SDO_UTIL.TO_WKBGEOMETRY and then dump to text via blob2plsql. To rebuild that blob into a geometry just push the blob into the MDSYS.SDO_GEOMETRY constructor. However, the Java-based WKB handling in Oracle Spatial is very old and only supports the most basic geometry types corresponding to the OGC Simple Features version 1.0. Its not going to work for LRS, 3D, or compound geometries. This utility uses the secret SDO_UTIL.TO_CLOB function to generate a bar-delimited version of the geometry object which while larger in size, should support all forms of Oracle Spatial geometries.

Parameters

p_input	MDSYS.SDO_GEOMETRY
p_comp_qual	UTL_COMPRESS.LZ_COMPRESS compression quality, the default of nine is the highest compression as why would you be doing this if you were not trying to pack things down as much as possible. Change if you like.

Returns

BLOB result.

dz_sdotxt_main.geomblob2sdo

Utility to convert a blob of compressed bar-delimited geometry back into MDSYS.SDO_GEOMETRY. The main purpose of this function is to unpack geometries converted to blobs with sdo2geomblob.

Parameters

p_input	BLOB of compressed bar-delimited geometry.

Returns

MDSYS.SDO_GEOMETRY result.

dz_sdotxt_main.dump_string_endpoints

Utility to convert to text the ordinates of the endpoints of a given linestring.

Parameters

p_input	MDSYS.SDO_GEOMETRY to convert endpoints to ordinates as text.

Returns

VARCHAR2 text value showing the ordinates of the endpoints of a linestring.

dz_sdotxt_main.dump_string_endpoints

Utility to convert to text the ordinates of the endpoints of two linestrings.

Parameters

p_input_1	MDSYS.SDO_GEOMETRY convert endpoints to ordinates as text.
p_input_2	MDSYS.SDO_GEOMETRY convert endpoints to ordinates as text.

Returns

VARCHAR2 text value showing the ordinates of the endpoints of two linestrings.

dz_sdotxt_main.dump_sdo_subelements

Utility to convert to text the component objects of a geometry collection.

Parameters

p_input	MDSYS.SDO_GEOMETRY collection to convert to text as individual objects.

Returns

CLOB text value of the individual components.

dz_sdotxt_main.dump_single_point_ordinate

Utility to extract a given ordinate from within a MDSYS.SDO_GEOMETRY.

Parameters

p_input	MDSYS.SDO_GEOMETRY from which to extract a given ordinate.
p_vertice_type	Either X, Y, Z or M.
p_vertice_position	vertice position in the geometry, default is 1.

Returns

NUMBER value of the ordinate.

dz_sdotxt_main.dump_mbr

Utility to convert to text the MBR surrounding a given geometry object.

Parameters

p_input	MDSYS.SDO_GEOMETRY to which derive the MBR to dump to text.

Returns

VARCHAR2 text value of the converted MBR.

dz_sdotxt_main.label_ordinates

Utility to converts all vertices in a geometry into a pipelined flow of points labelled by vertice number.

Parameters

p_input	MDSYS.SDO_GEOMETRY to convert to labeled points.

Returns

PIPELINED table of dz_sdotxt_labeled objects.

dz_sdotxt_main.label_measures

Utility to converts all vertices in a geometry into a pipelined flow of points labelled by LRS measure value.

Parameters

p_input	MDSYS.SDO_GEOMETRY to convert to labeled points.

Returns

PIPELINED table of dz_sdotxt_labeled objects.

dz_sdotxt_main.break_2499

Utility to take a very long sdo text representation and add linefeeds to meet the sqlplus 2499 character length restrictions.

Parameters

p_input	CLOB of sdo object representation.
p_delim_character	delimiter to add to input, default is linefeed chr(10).
p_break_character	character upon which to add a delimiter, default is comma.
p_break_point	character count near which to add delimiters, default is sqlplus 2499 limit.

Returns

CLOB with breaking characters added after commas.