DZ_TOPO

- Build ID: 5
- TFS Change Set: 8290

Utilities and helper objects for the manipulation and maintenance of Oracle Spatial topologies.

In order to avoid passing about all the gritty details of a topology workflow, the dz_topology object encapsulates the detail of the topology and topo map session for easy reuse by topology operators.

Example

```
obj_topo := dz_topology(
   p_topology_name    => 'CATROUGH_NP21'
  ,p_active_table     => 'CATROUGH_NP21_TOPO'
  ,p_active_column    => 'TOPO_GEOM'
);

obj_topo.set_topo_map_mgr(
  dz_topo_map_mgr(
     p_window_sdo       => ary_shape
    ,p_window_padding  => 0.000001
    ,p_topology_name    => 'CATROUGH_NP21'
  )
);

obj_topo.java_memory(p_gigs => 13);

FOR i IN 1 .. ary_featureids.COUNT
LOOP
  topo_shape := obj_topo.create_feature(
    ary_shape(i)
  );

  INSERT INTO catrough_np21_topo(
     featureid
    ,topo_geom
  ) VALUES (
     ary_featureids(i)
    ,topo_shape
  );

END LOOP;

obj_topo.commit_topo_map();
COMMIT;
```

Summary

DZ_TOPO

Functions

dz_topo_main.aggregate_topology	Function which takes an arbitrary number of topo geometries and returns an aggregated polygon representing the external boundary of all faces comprising the topo geometries.
dz_topo_main.face2sdo	Utility function which provides the sdo geometry polygon representing a single face of a topology.
dz_topo_main.face_at_point_sdo	Utility function which provides the sdo geometry polygon representing a single face of a topology located using an input point geometry.
dz_topo_main.nuke_topology	Fairly dangerous procedure to remove all traces of given topology including all topo tables registered to the topology.

dz_topo_main.unravel_face	Utility procedure intended to gather information for the possible manual removal of a topology face by identifying and first removing it's surrounding referenced components.
dz_topo_validate.check_all	Utility function to validate components of a given topology.
dz_topo_validate.check_basic	Utility function to validate a subset of a given topology.
dz_topo_validate.check_universal_face	Utility function to inspect and report on topology elements which interact with the universal face.
dz_topo_reports.edge_report	Reporting function to dump pertinent information about a given edge and how the edge relates to all its neighbors including notes about any corruption found amongst the edge neighbors.
dz_topo_reports.face_report	Reporting function to dump pertinent information about a given face and how the face relates to all its neighbors.

FUNCTIONS

dz_topo_main.aggregate_topology

Function which takes an arbitrary number of topo geometries and returns an aggregated polygon representing the external boundary of all faces comprising the topo geometries.

Parameters

p_topology_owner	optional owner of the topology
p_topology_name	optional name of the topology
p_topology_obj	optional object dz_topology
p_input_topo	object dz_topo_vry, a varray of topo geometries
p_remove_holes	optional TRUE/FALSE flag to remove resulting interior rings

Returns

MDSYS.SDO_GEOMETRY spatial type

Notes

- Dz_topo functions and procedures either utilize an existing dz_topology object or require the name and owner of the topology to internally initialize one. Thus you either must provide name and owner or an initialized dz_topology object to the function. The latter format is most useful when calling several topology procedures in sequence.
- By design this function does not invoke the .get_geometry() subroutines of Oracle Spatial topologies or utilize any vector spatial aggregation techniques. Rather the resulting polygon is built directly from chained together topology edges. The value of this approach is that it is not necessary to precompute relationships among existing topo geometries to obtain aggregated results. For example with a topology of watershed polygons, provide a drainage area based upon initial location and arbitrary upstream distance. Its unwieldy to impossible to prebuild topology layers that could represent every possibility of polygon combinations. This function provides the ability to aggregate together an arbitrary input of topo geometries.

dz_topo_main.face2sdo

Utility function which provides the sdo geometry polygon representing a single face of a topology.

Parameters

p_topology_owner	optional owner of the topology
p_topology_name	optional name of the topology
p_topology_obj	optional object dz_topology
p_face_id	single numeric face id

p_remove_holes	optional TRUE/FALSE flag to remove resulting interior rings

Returns

MDSYS.SDO_GEOMETRY spatial type

Notes

- Dz_topo functions and procedures either utilize an existing dz_topology object or require the name and owner of the topology to internally initialize one. Thus you either must provide name and owner or an initialized dz_topology object to the function. The latter format is most useful when calling several topology procedures in sequence.

dz_topo_main.face_at_point_sdo

Utility function which provides the sdo geometry polygon representing a single face of a topology located using an input point geometry.

Parameters

p_topology_owner	optional owner of the topology
p_topology_name	optional name of the topology
p_topology_obj	optional object dz_topology
p_input	sdo geometry input point

Returns

MDSYS.SDO_GEOMETRY spatial type

Notes

- Dz_topo functions and procedures either utilize an existing dz_topology object or require the name and owner of the topology to internally initialize one. Thus you either must provide name and owner or an initialized dz_topology object to the function. The latter format is most useful when calling several topology procedures in sequence.

dz_topo_main.nuke_topology

Fairly dangerous procedure to remove all traces of given topology including all topo tables registered to the topology. Do not use unless you are sure you do not need any of the data participating in your topology.

Parameters

p_topology_owner	Optional owner of the topology to remove.
p_topology_name	Name of the topology to utterly remove.

Notes

- Please be careful with this. I am not responsible for you destroying your topology.

dz_topo_main.unravel_face

Utility procedure intended to gather information for the possible manual removal of a topology face by identifying and first removing it's surrounding referenced components. This is usually done in cases of corruption in the topology. So upon providing a tg layer and face id, the procedure returns the list of tg ids, faces, edges and nodes that must be removed in order to delete the input face. A sdo window is also provided for use in constraining a topo map to the work area in question.

Parameters

p_topology_owner	Optional owner of the topology
p_topology_name	Optional name of the topology
p_topology_obj	Optional object dz_topology
p_tg_layer_id	the tg_layer to execute the unravel against
p_face_id	the face to execute the unravel against
p_tg_ids	output tg ids
p_face_ids	output face ids
p_edge_ids	output edge ids
p_node_ids	output node ids
p_window_sdo	output geometry window surrounding the action

Notes

- Note this procedure only returns information about a topology and does not make any changes to the input topology.
- Taking the results of this procedures and manually deleting resources in your topology is highly dangerous and should not be done unless all other options have been explored first.
- On the other hand this procedure could also function as the core of a nice little topology viewer showing you how your face in question ties together with other components.

dz_topo_validate.check_all

Utility function to validate components of a given topology.

Parameters

p_topology_owner	owner of the topology
p_topology_name	name of the topology

Returns

Pipelined table of VARCHAR2(4000) informational messages

Notes

- Note for a large topology this function may require a long time to complete.

dz_topo_validate.check_basic

Utility function to validate a subset of a given topology. At this time the function is the same as check_all.

Parameters

p_topology_owner	owner of the topology
p_topology_name	name of the topology

Returns

Pipelined table of VARCHAR2(4000) informational messages

Notes

- Note for a large topology this function may require a long time to complete.

dz_topo_validate.check_universal_face

Utility function to inspect and report on topology elements which interact with the universal face.

Parameters

p_topology_owner	owner of the topology
p_topology_name	name of the topology

Returns

Pipelined table of VARCHAR2(4000) informational messages

Notes

- This utility is primarily concerned with reporting on island edges which are not contained within the universal face.

dz_topo_reports.edge_report

Reporting function to dump pertinent information about a given edge and how the edge relates to all its neighbors including notes about any corruption found amongst the edge neighbors.

Parameters

p_topology_owner	optional owner of the topology
p_topology_name	optional name of the topology
p_topology_obj	optional object dz_topology
p_edge_id	the id of the edge to inspect

Returns

Pipelined table of VARCHAR2(4000 Char)

Notes

- Dz_topo functions and procedures either utilize an existing dz_topology object or require the name and owner of the topology to internally initialize one. Thus you either must provide name and owner or an initialized dz_topology object to the function. The latter format is most useful when calling several topology procedures in sequence.
- The edge report was created for debugging purposes.

dz_topo_reports.face_report

Reporting function to dump pertinent information about a given face and how the face relates to all its neighbors.

Parameters

p_topology_owner	optional owner of the topology
p_topology_name	optional name of the topology
p_topology_obj	optional object dz_topology
p_face_id	the id of the face to inspect

Returns

Pipelined table of VARCHAR2(4000 Char)

Notes

- Dz_topo functions and procedures either utilize an existing dz_topology object or require the name and owner of the topology to internally initialize one. Thus you either must provide name and owner or an initialized dz_topology object to the function. The latter format is most useful when calling several topology procedures in sequence.

- The face report was created for debugging purposes.