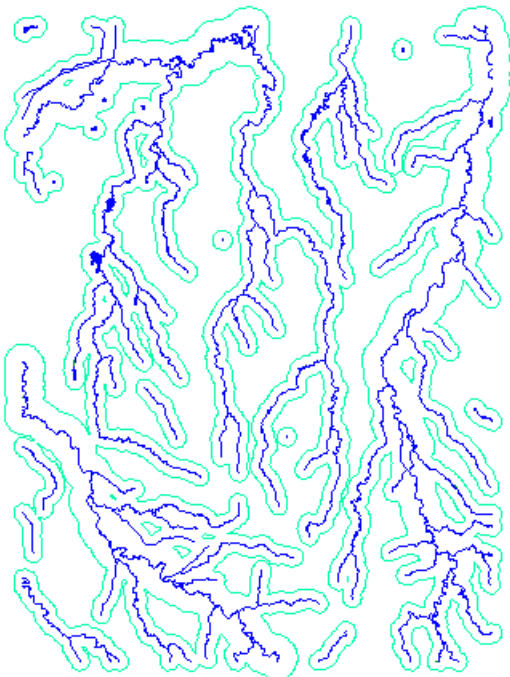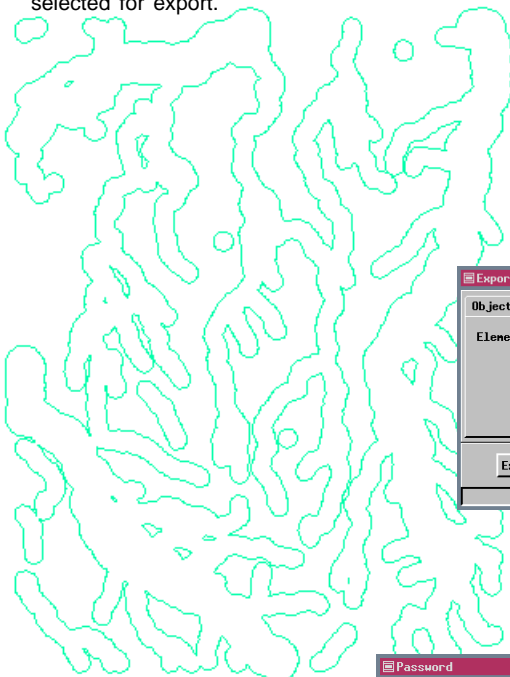# Exporting Vector Objects to Oracle® Spatial Layers

Buffer zones (green) were generated in TNTmips for the hydrology (blue) that was imported from Oracle Spatial. The buffer zones only were selected for export.

You can export any TNTmips® vector object to an Oracle Spatial layer. You may want to import and interactively edit a vector as the basis for some analysis, such as vector combinations, buffer zone generation, or polygon fitting, and then export the result. Any object you export will create a layer that at least temporarily has polygonal, planar, or network topology matching that of the object. Until that topology is modified by subsequent Oracle Spatial actions, it is available for analysis within Oracle. The analysis might be as simple as using the mutually exclusive areas of polygons exported from a vector object with polygonal topology.

Oracle Spatial enforces very strict database rules that are not required by the TNT products. The following are such Oracle Spatial requirements:
* The primary key field must be unique, which means you cannot have two records in the same table with the same primary key value.
* All records must have a value for the primary key field.
* All foreign key field entries must be present in the primary key field.
* Referenced fields must be the same type (for example, string).

See the back of this sheet for the messages you will receive and how to remedy these conditions.

If your table does not meet these criteria, you will get a message on export that names the fields that are unacceptable to Oracle Spatial and why. Using TNTmips, you will have to modify the database tables associated with the vector you want to export so that it conforms to these requirements before you can successfully export them. You generally get the spatial table and one or more other tables exported even if there is a condition present that keeps all tables from being exported.

If more than one record from a table is attached to an element, only one attachment will survive because of the way relationships are structured in the spatial table.
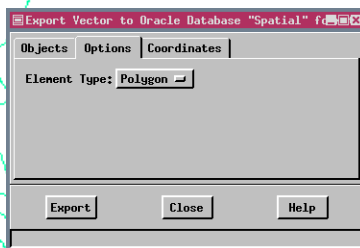
You need to specify the element type to export because only one element type at a time can be exported.
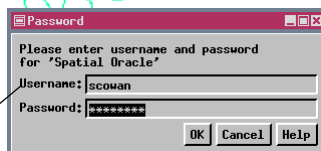
After you select the desired Data Source Name, such as Oracle Spatial, the Password prompt opens. Enter your Oracle user name and password, click OK, then click on the Export button.
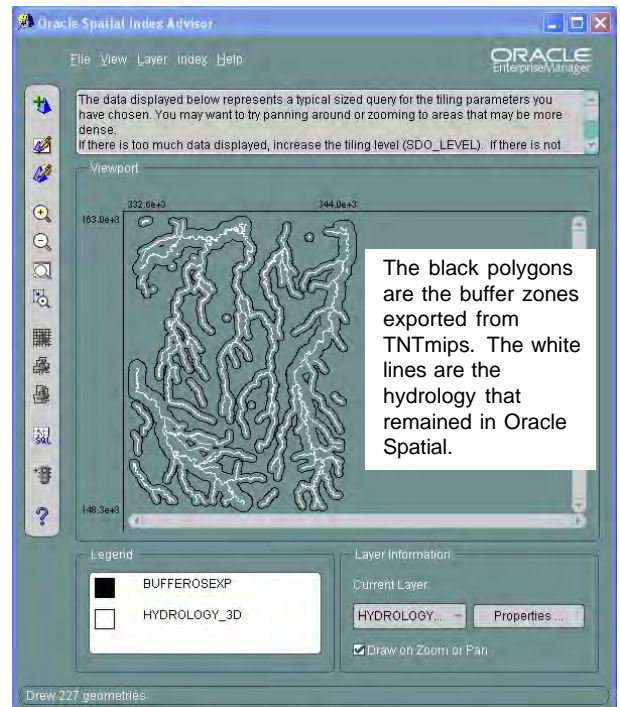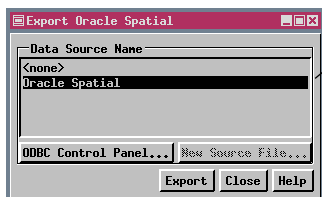

The black polygons are the buffer zones exported from TNTmips. The white lines are the hydrology that remained in Oracle Spatial.

Export Warnings and How To Fix Them

Oracle Spatial enforces very strict database rules that are not required by TNTmips. You can, of course, create tables that comply with these rules in TNTmips or modify existing tables to comply with these export rules. When you get warning messages on export, be aware that these are Oracle Spatial's limitations, not limitations in TNTmips. There is a common Message window for all such warnings; you need to click on the Details button to get information on the specific requirement that is not met. The following are the SQL execution errors that have been encountered on export of some standard data, what those errors mean, and how to go about fixing them.

- **unique constraint (schema.field_PK_table) violated**

  The identified field in the identified table is a primary key (PK) with the same value found in more than one record. To fix, select multiply-attached elements (use Tabular View in the View Selected Elements mode). Identify whether you have duplicate records (records that have the same values for all fields). If you do, choose Edit Relations, right click on the box that represents the named table, and choose Delete Duplicate Records. If you do not have duplicate records, you will need to decide which of the records is correct for the primary key in question and delete the record or alter the primary key field value. If no records were found when you selected multiply-attached elements, follow the steps for removing duplicate records except choose Delete Unattached Records when editing relations.
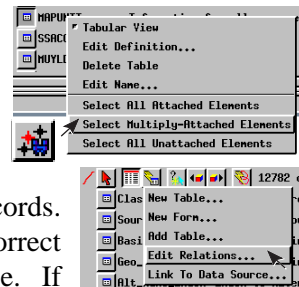
- **cannot validate (schema.field_FK_table) - parent keys not found**

  The identified field in the identified table is a foreign key (FK) that has values not found in the referenced primary key table. The last line of the message will identify the table and field referenced by this foreign key in the form table (field). Sort on the identified fields in both the foreign key and primary key tables (if not already sorted). When value(s) is found, check to see that the record is attached to elements (if not, delete the record), then take appropriate action. Appropriate action would be to fix a typographical error in either the foreign or primary key field or to add a corresponding record in the primary key table. It could also be that a value has not been entered for a foreign key field, in which case you must enter a value that is already in the primary key table or enter a new value and also add a record with that value to the primary key table.

- **column type incompatible with referenced column type**

  Field types for a primary/foreign key pair are not the same (for example, one is an integer and one is a string). The tables and fields involved are identified in the last two lines in the message window in the form field_fk_table for the foreign key and table (field) for the primary key. The solution is straightforward but involves several steps because you cannot change the field type after it is added to a table. Edit the definition of the table with the foreign key. Select the identified key field. Click on the Add Field icon. Set the field type to String Expression if you need to create a string field from a numeric field or Computed if you need to create a numeric field from a string field or convert between numeric types. Click on Edit Expression and enter the expression required for your circumstances. Click OK in the Query Editor and Table Definition windows. Open a tabular view of the foreign key table, right click on the field name and choose Make Field Permanent. Return to the table definition and delete the original foreign key field that was the wrong field type. Change the name of the new field to be the old field name. Set up the relationship to the primary key table.
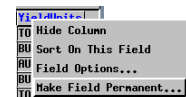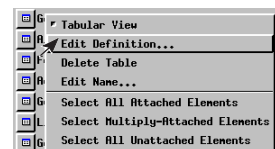
- **cannot insert Null into ("schema"."table"."field")**

  A field that has the "Cannot be left blank" constraint set in the TNT products does not have values in all records. A blank field can occur if you have entered values in tabular view and not entered values in all records for that field. Constraints are only enforced in Single Record View. Either enter values for the constrained field in all records or choose Edit Definition and remove the field constraints.