

Database

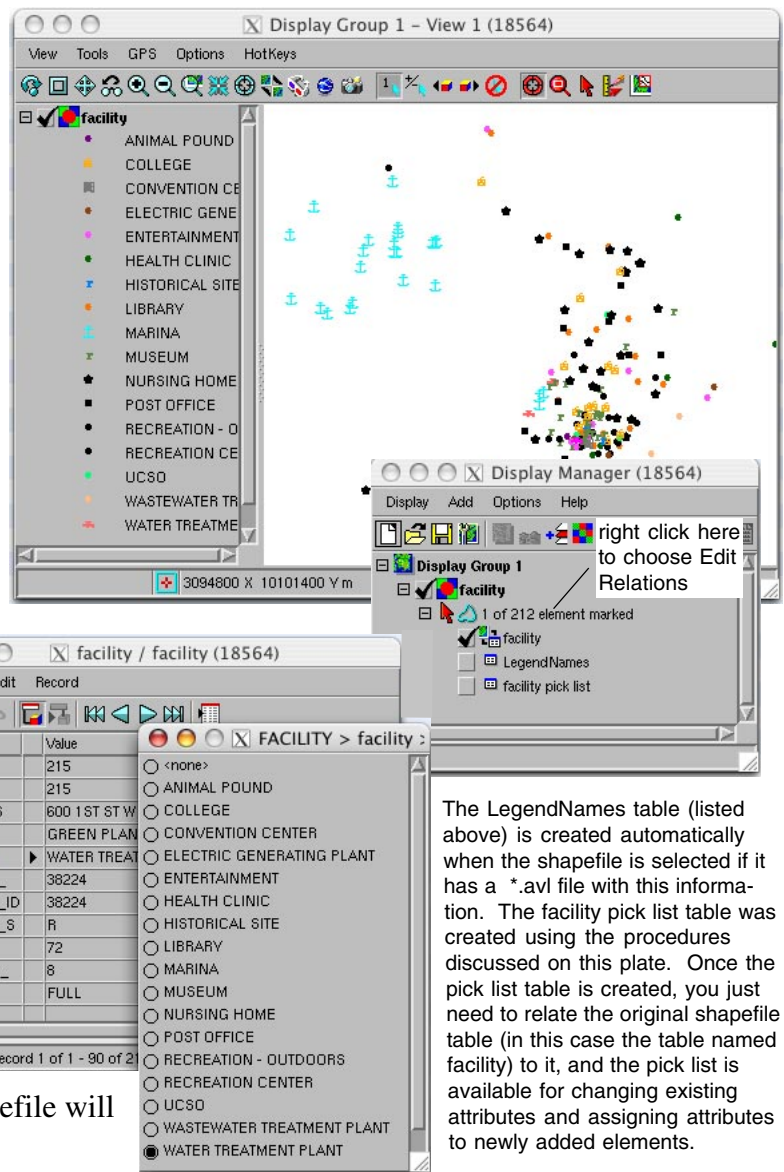
Create a Picklist for a Shapefile

The TNT products support direct use of shapefiles in any process that takes geometric object input (for example, Display, Edit, Merge to Vector). Shapefiles are files with an SHP extension that contain one type of geometric element (point, line, or polygon) associated with a single database table. However, shapefiles are incomplete without one or more same-named additional files that may accompany the shapefile with different extensions. Examples of these same-named files are *.dbf that provides a database table, *.prj to provide georeference, *.avl to provide styles, a variety of index files (*.shx, *.sbn, *.sbx), and so on. The TNT products make use of these auxiliary files when available.

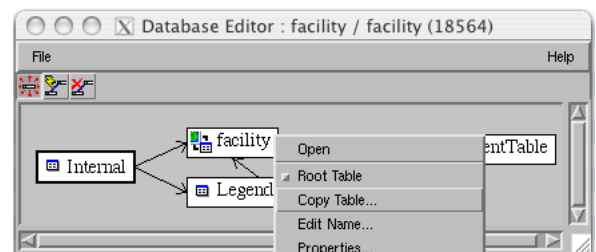
The accompanying DBF and SHX files are the only additional files required for use of a shapefile (any of the other same-named files are optional). The SHX file provides an index between the elements in the SHP file and the records in the DBF file. Each element in the shapefile must have an attached record in this database file. However, this type of database file allows for only one table, which means it does not support relational databases. You can, however, create a relational database for use with a shapefile in the TNT products. Additional tables and other TNT features, such as display parameters, are stored in a TNT same-named *.rlk file. Other products displaying the same shapefile will not see these additional tables.

All the information you need for your picklist table is likely already in the shapefile and just needs to be copied and have some other adjustments made. The easiest way to copy a table is to choose Edit Relations from the right mouse button menu for the elements, then right click on the desired table representation (in this case facility) in the Database Editor window that opens. Choose Copy Table from the menu that opens. This selection lets you name your copy and produces a new table with all the fields as the original and the same attachment type. If the original was attached to elements, the copy will be attached to the same elements. The copy, however, will not be linked to the *.dbf as the original is, which means adding more choices (new records) must occur in a TNT product.

The only necessary steps remaining are to change some of the properties of both the original and copied tables. You may choose additional steps to simplify the copied table, which will provide the same picklist whether or not it's



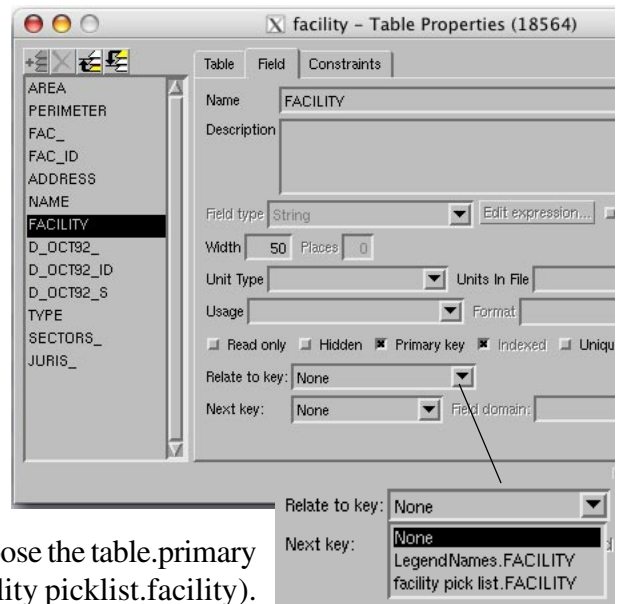
The LegendNames table (listed above) is created automatically when the shapefile is selected if it has a *.avl file with this information. The facility pick list table was created using the procedures discussed on this plate. Once the pick list table is created, you just need to relate the original shapefile table (in this case the table named facility) to it, and the pick list is available for changing existing attributes and assigning attributes to newly added elements.



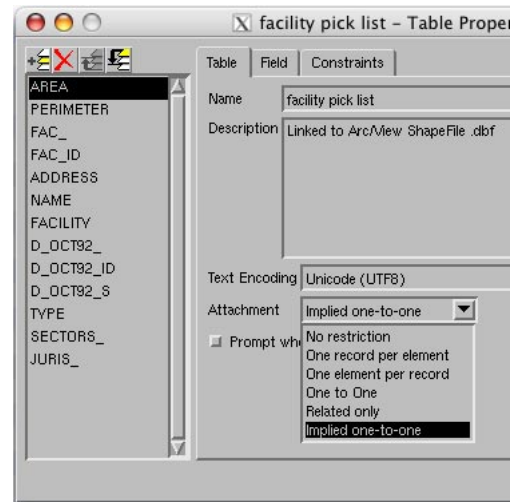
simplified. Both the necessary and optional procedures are described below.

It does not matter whether you make the changes to the original or copied table first. In this example the copied table is named *facility picklist* to make it clear what the table is intended for. In actual usage *facility type* would be a more likely name.

Original Table. With the Database Editor window that you used to copy the table still open, choose Properties from the right mouse button menu for the original table. Click on the Field tab in the Table Properties window that opens. Click on the primary key entry in the list of fields at the left of the window. In this example, it is Facility. Turn off the Primary Key check box. This step is actually not necessary, but will avoid some confusion. Next click on the downward arrow at the right of the Relate to key field and choose the table.primary key field you want to relate to in the copied table (in this case *facility picklist.facility*). You are done with the changes to the original table and just need to click the OK button.



Copied Table. Still in the Database Editor window, right click on the copied table and choose Properties from the menu. Click on the Table tab if it is not already active. Click on the downward arrow to the right of the Attach field and change Implied one-to-one to Related only. This step is the last one necessary to get your picklist. However, to reduce the size of your table you may want to take two additional steps. The first is to remove all of the fields from the table that are not necessary to provide the pick list. In this example, you would delete all fields except Facility using the Delete icon at the top of the list. At this point you are done in the Table Properties window and can click OK. The second step in reducing the size of the table is to remove duplicate records. Because you are using a shapefile, which requires a record in the original table for every element, you will have many duplicate records. **When presented as a pick list, this issue is automatically resolved for you—each attribute value is presented only once.**



However, you may be able to greatly reduce the size of the table by removing duplicate records. Consider the case where you may have hundreds of thousands of records but only 50 different attribute values. Removing duplicate records would reduce the number of records in the table from hundreds of thousands to 50. To remove duplicate records, right click on the copied table representation in the Database Editor window and choose Delete Duplicate Records. You can now choose File/Close in the Database Editor window and you are ready to use your picklist for changing existing attributes or assigning attributes to added elements.

You are not limited to attribute values in the initial shapefile table for your picklist. If you need to add more, simply add new records to the table that provides the picklist. If you modify existing attribute values in your picklist tables, be sure to update the fields in the original table that use these values. It will not matter for usage in TNT, but if you export to a spatial data format that requires all foreign key values to match a value for the corresponding primary key, such as Oracle Spatial, if you do not update you will get errors on export.

