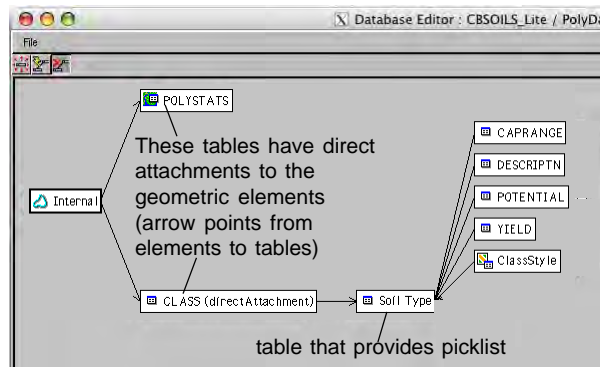


Database

Picklist Relational Structure

Assigned attributes provide the *Information* about geometric features in Geographic Information Systems. The spatial data you work with for GIS purposes may run the gamut from one extreme (geometric elements organized into files/objects that are georeferenced and are linked to a complete relational database) to the other extreme (geometric elements that do not exist until after you create them using photointerpretive techniques, real-time GPS input, or by some other means). You might also have a template object that contains the complete, reusable database structure along with georeference but no elements (see the Technical Guide entitled *Spatial Editor: Creating and Using Geodata Templates* and *Database: Choose Attributes Using a Picklist*). When working with newly created elements in new files/objects, you may have an established relational database framework to associate with these geometric elements (either as part of a template or from some other data source) or you may need to create all the tables and relationships yourself. You may also have files/objects with existing geometric elements and a fully structured relational database that need to be updated by adding new elements or adjusting element database attachments. Regardless of where your project falls on the spectrum, picklists are designed to make assigning attributes to geometric elements easier.

To understand how picklists work, you need to understand the use of relational databases in geospatial applications. Relational databases contain multiple tables that may not be directly attached to individual geometric elements and yet the appropriate record can be identified by selecting an element. For example, if a record directly attached to an element you select has soil type as the primary key, records in all other open tables that reference this primary key and have the same soil type are also selected. You can view these relationships between elements and tables in a TNT relational database by choosing Edit Relations from the right mouse button menu for the database in the Display process or using Tools/Database/Edit. This choice opens a tree view of the structure of your relational database as illustrated at the right. Tables that are related but have no direct attachments to elements are referred to as related-only tables. For more information on the tree view features, see the *Managing Relational Databases* tutorial.



All tables except the polystats table have a foreign key relationship to the Soil Type table illustrated, which means that the Soil Type table contains a primary key pointed to by a field in each of these other tables (i.e., arrows from these tables point to Soil Type table). All of the records in all of the related-only tables will be preserved for use in new objects if the geometric object is saved as a template.

Primary versus Foreign Key Fields

Relationships between tables, and thus between attributes and geometric elements, are established using key fields. A primary key is an important field in a table that is referred to by other tables. It is the means by which other tables are related to geometric elements. Soil type is an example of a primary key field that may be referenced by other tables that present additional information based on soil type, such as yield and wildlife habitat suitability. Many database systems require that primary key fields cannot be blank in any record and that the values be unique in every record. (The TNT products do not enforce uniqueness to allow for compound primary keys where the value is composed of multiple fields, such as first, middle, and last name fields in a single individual's record. Each one of these fields may have values duplicated in other records.) A foreign key field points to a primary key in another table and uses the same values presented for the primary key. Primary key/foreign key relationships can establish an association between a selected element with a single directly attached record and records in any number of other relational tables that lack direct attachments but share the same primary and foreign key values.

Use a One-to-One Attachment

Picklists are available when the primary key field resides in a related-only table. A table that makes use of the picklist should have an implied one-to-one attachment type for greatest ease in assigning attributes for a new vector or shape object. An implied one-to-one attachment type means that there is a record for every element, every element has a record attached (which means the number of elements equals the number of records), and the record number corre-

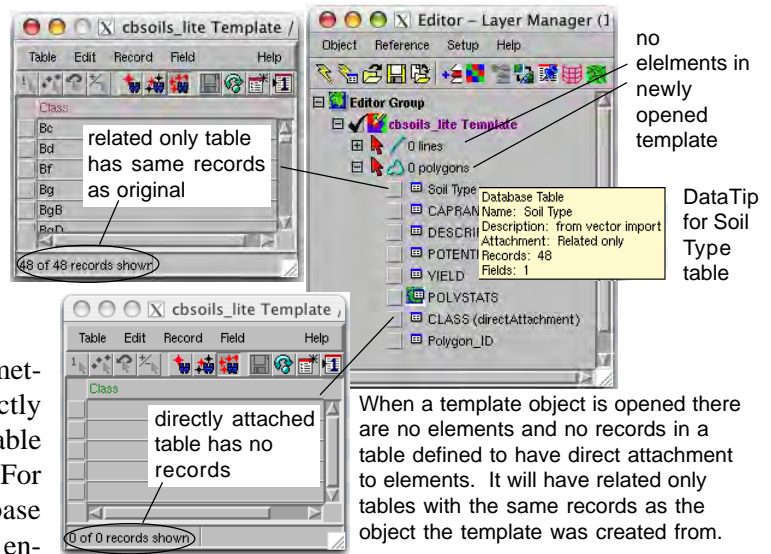
sponds to the element number. A corollary effect of these special conditions is that these tables with implied one-to-one attachment automatically have a record associated with every element. CAD objects do not support implied one-to-one tables; choose one record per element, one element per record, or one to one for this object type. Other related-only tables may also make use of picklists to populate fields, but a table that is directly attached to the elements and also is related by reference to some primary key in common is necessary for such a related-only table to achieve association with elements.

Three Common Situations

Although any combination of attributes and geometric elements is possible, three basic situations you may encounter when viewing, editing, or creating objects with attributes you want to assign from picklists are described below.

Using picklists with correctly structured databases or templates. When geometric data is acquired by import, the object will have existing elements and may have a relational database structure depending on the source of the data. You may need to add elements or reclassify existing elements in geometric objects that already have relational databases. TNT object templates have the same relational database structure as the objects they were created from but have no elements (see the TechGuide entitled *Spatial Editor: Creating and Using Geodata Templates*). If the relational database has a foreign key field in the table attached to the elements (i.e., from an existing object) or that will be attached to elements (i.e., via a template), you will automatically have a picklist to provide values for that field. You can proceed as described in the TechGuide entitled *Database: Choose Attributes Using a Picklist*. You may also have a complete relational database without any elements if you created a template from a similar object. When an object is saved as a template, the complete database structure is maintained, but tables directly attached to elements **will not have any records**. Records in related only tables **are copied** to the template object. Thus, when adding elements to a template object, you will automatically have the same picklists and picklist values as the original object to use to select attributes. A number of external file types, such as shapefiles, DGN, DXF, and TAB files, do not support relational structures. For imported files that do not support relational databases or do not have one set up, you will need to follow the procedures outlined below for adjusting a database to use picklists.

The database structure of this template object matches that of the treview for the object the template was created from illustrated on the front.



Creating a database to use picklists. In a new geometric object you want to create the table that will be directly attached to the elements first. Then create the related table that contains the picklist before you add any elements. For more information on creating new objects with a database structure that will provide picklists, see the TechGuides entitled *Database: Wizard to Create a New Picklist* and *Database: Creating Picklists Manually for New Objects*.

Adjusting a database to use picklists. The information used to populate the values in a picklist may exist in a text file or external database table. It may also be included as one of the fields in an existing table that is simply not correctly related to supply a picklist (e.g., it is directly attached to the elements or is not referenced by a foreign key). Proceed as above if you lack the necessary tables. If the picklist field is not a primary key, you may only need to adjust the table properties to establish the primary/foreign key relationship to get the picklist desired. You can set these relationships manually or use the picklist wizard to create a new table and establish the relationships for you. Native shapefiles and some other formats can have only one database table and every element must have a record attached. A means of creating and using picklists within the TNT products for such files is described in the TechGuide entitled *Database: Creating Picklists for Use with Shapefiles*.

