MANAGING DBS

Tutorial

Managing Relational Databases

with

TNTmips®

TNTedit™

TNTview®

Before Getting Started

TNTmips®, TNTedit™, and TNTview® provide great flexibility when working with database materials. Databases can be in the internal format, in a variety of external formats supported by ODBC (Open Database Connectivity) and OLE DB (Object Linking and Embedding, Database), or in a spatial database format, such as Oracle Spatial or ESRI Personal Geodatabase. You can visualize information directly from databases as pinmaps (provided coordinate locations are included with each record) or as shape objects or the information can be related to elements in any object type for viewing and use in analysis operations.

Prerequisite Skills This booklet assumes you have completed the exercises in the *Displaying Geospatial Data* and *TNT Product Concepts* tutorial booklets. Those exercises introduce essential skills and basic techniques that are not covered again here. Please consult these booklets for any review you need.

Sample Data The exercises presented in this booklet use sample data distributed with the TNT products. You can download the data from MicroImages' web site. The first exercises in this booklet use the CB_SOILS Project File in the DB_MANAG directory of DATA. The objects in the CONSTRNT, PROPINFO, UNTDSTAT, and USA_OLD Project Files, as well as a text file and format files in this directory, are also used. Make a read-write copy of the sample data on your hard drive so changes can be saved when you use these objects.

More Documentation This booklet is intended only as an introduction to the database functions in TNTmips, TNTedit, and TNTview. Consult the reference materials at microimages.com for more information.

TNTmips Pro, TNTmips Basic, and TNTmips Free TNTmips comes in three versions: TNTmips Pro, TNTmips Basic, and TNTmips Free. HyperIndex Linker is available only to customers using TNTmips Pro. If you did not purchase the professional version (which requires a software license key), you are using TNTmips Free or TNTmips Basic, which limit the size of your project materials. All exercises in this booklet can be completed in TNTmips Free using the sample geodata provided.

Merri P. Skrdla, Ph.D., 20 August 2009
© MicroImages, Inc. 2002–2009

It may be difficult to identify the important points in some illustrations without a color copy of this booklet. You can print or read this booklet in color from MicroImages' web site. The web site is also your source of the newest tutorial booklets on other topics. You can download an installation guide, sample data, and the latest version of TNTmips Free.

http://www.microimages.com

Welcome to Managing Databases

Attributes stored in databases can be visualized directly (pin mapping or shape objects) or in conjunction with elements in vector, CAD, TIN, shape or even raster objects. The use of attributes is one of the key differences that separates geographic information system (GIS), or geospatial analysis, software from simpler graphics packages.

There are three booklets in this tutorial series that address different aspects of database / attribute utilization and management in TNTmips: this booklet, *Pin Mapping* , and *Managing Geoattributes* . This booklet focuses on the databases themselves with sections on viewing and manipulating the structure of a database (what tables it contains and how they are related), adding tables and making forms, and introducing statistics and computed fields to augment the information already present in your database. How to use databases in other formats is also described. The *Pin Mapping* booklet describes how to visualize databases directly when each record contains location information. The *Managing Geoattributes* booklet describes how to get the database information associated with elements in other object types. Obtaining histograms and XY plots from field values and saving tables in CAD format for inclusion in hardcopy layouts are also described in the *Managing Geoattributes* booklet.

All of the procedures described in this booklet are available in TNTmips, TNTedit, and TNTview. The primary database management utility is available in TNTmips as both a stand alone process and from within the Display process. The stand alone process is absent in TNTview, but all the features are available in the main display process (choose Edit Relations from the menu associated with the Make Table / Form icon).

STEPS

- ☑ use your operating system to copy the needed .dbf and .txt files (see page 2) to your local drive
- ☑ launch TNT
- ☑ use Tools / Manage Project Files or your operating system to copy the RVC files in the DB_MANAG directory of DATA
- ☑ choose Main / Display

Page 4 defines some key database terminology. Pages 5–17 provide instruction for using the features of the Database Editor. Pages 18–28 lead you through a variety of techniques, such as adding tables, using computed fields, introducing statistics and constraints, and using picklists. Pages 29–34 describe database import, linking, OLE DB, and ODBC.

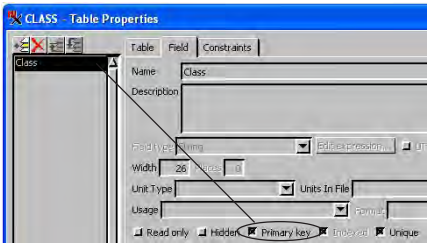
Database Terminology

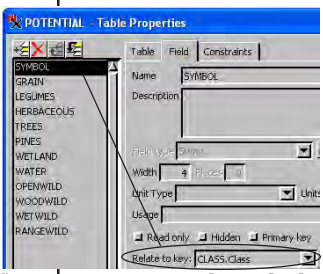
Vocabulary: a **primary key** field provides unique identification of related elements, such as the soil type of a polygon in a soil map. Once the records in a table containing a primary key field are attached to the elements in an object, other tables that contain this same information can be related to the same elements by referencing the primary key in the directly attached table. When the primary key field in one table contains the same information and is referenced by a field in another table, it is called a **foreign key**. The field that references this foreign key is a **foreign key field**.

In some cases in order to be unique, a primary key must be made from a number of field values, such as first, middle, and last name fields. The first of these fields is designated the primary key and the other fields are each referred to as a **next key** with the order established by the designation.

Database software often only allows one primary key in a table, but this is not strictly enforced in the TNT products since organization of materials into tables may make two primary keys a logical choice for reference by other tables (such as address and resident's name).

A database in TNTmips can be either a main level object used for pin mapping or a subobject of a vector, CAD, TIN, sketch, shape, or raster object. Databases that are subobjects contain attributes associated with elements (vector, CAD, TIN, sketch, shape) or cell values (raster). These attributes can be organized into one or more tables. Each table covers different aspects of the same common theme defined by the database.





In order to retrieve attribute information for selected elements, individual records within a database table must be attached or related to those elements. (Table attachment types and establishing attachments between records and elements are described in the *Managing Geoattributes* booklet.) A relational database lets you define a primary key in one table that is directly attached to individual elements and then relate all other tables with the same attribute to the same elements by establishing a foreign key relationship. Thus, even a database as complex as that associated with the Crow Butte soil map (23 soil-related attribute tables) requires minimal effort to establish a retrievable relationship between polygons and the records in each of the tables. In the Crow Butte soils polygon database, only two of the 23 tables are directly attached to the elements, and one of these (the polygon statistics table) had the attachments created automatically by the process that generated the table. The other 21 tables are indirectly related by foreign key fields.

View Existing Database Structure

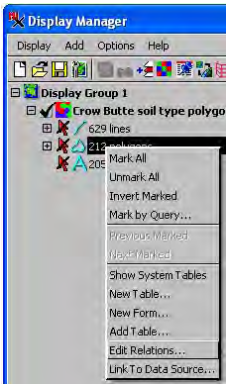
TNTmips provides the tools for you to manage a database as a whole rather than simply table by table. The Database Editor is similar to the Relationships tool in Microsoft Access. The Database Editor shows you the relationships between tables and lets you change these relationships as well as providing access to all of the other database functions.

To view and edit database structure in the Display process or TNTview, select the object with which the database is associated for display, expand the listing for the layer, and choose Edit Relations from the menu for the appropriate element type. You can use the Database Editor whether or not the element type with which the database is associated is active for selection. In TNTmips, you can also select Tools / Database / Edit to run the process. You need to then specify the element type if a vector object is chosen.

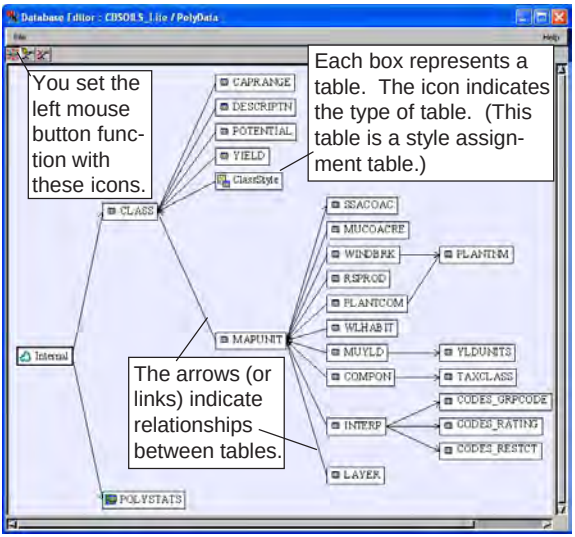
When you add a layer, you need to turn expand its entry if you want to list all database tables or view attributes in any but the first table, which opens automatically when you select an element if you have not expanded the layer's entry.

STEPS

- ☑ click on the New icon in the Display Manager, choose 2D Display, and select CBSOILS_LITE from the CB_SOILS Project File
- ☑ expand the layer
- ☑ right-click on the polygon element row, and select Edit Relations from the menu



- ☑ click on the Move Table icon if it is not currently the active icon on the Database Editor toolbar



Move Table

Relate

Unrelate

Keep this database open in the Database Editor through the exercise on page 9.

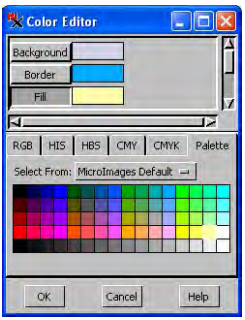
Colors and Text Style

STEPS

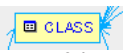
- ☑ right-click over the background of the Database Editor window and select Text Style from the menu

- ☑ click on [Font] to select a different font, choose one from the list, and click [OK] in the Scalable Font Selection window
- ☑ click on a different tile in the Foreground Color palette, then click [OK]
- ☑ right-click over the background of the Database Editor window and select Colors from the menu

- ☑ click on [Fill] and choose a new Fill color from the palette for the table rectangles then click on [Border] and select a contrasting color for the rectangle outlines and links



- ☑ click [OK] in this Color Editor window

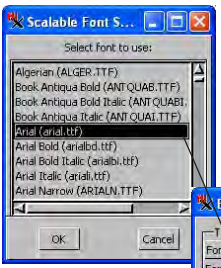


- ☑ redo any of the steps on this page you want to further adjust color and style

All the features available for styling text in TNTmips are available for table names in the Database Editor. The font initially selected provides good readability at relatively small size. The table rectangles are resized if necessary to accommodate changes in text width and height.

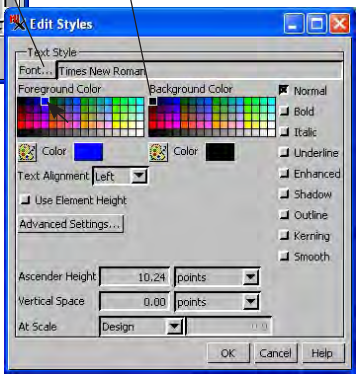
The text style settings chosen for the sample shown toward the bottom of this page are 8-point Arial in blue. A change in size and font may increase the table rectangle sizes so the database no longer fits entirely within the Database Editor window (shown on page 5). You can resize the window, scroll to view all the tables, or move the tables so they once again all fit within the window if you find a larger font is to your liking.

You can change the fill and outline colors for the table rectangles. The outline color is also the color in which the links are drawn, so red is not a recommended choice (selected links are highlighted in red). A "missing" color can also be set to distinguish linked tables that cannot be found from other tables in the database.



The font you select here is for the Database Editor interface only—it does not affect how the table itself is displayed.

This background color is used in addition to the foreground color with Enhanced and Shadow styles.



Mouse Functions in Move Table Mode

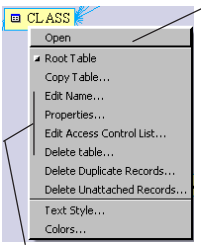
You can use the mouse three different ways to get at features in the Database Editor: click left, click right, and double-click (left). On a Macintosh with a single-button mouse, the right mouse button is simulated by holding the Command key (Open Apple) and clicking. The function of the left mouse button is determined by the icon button that is toggled on in the toolbar. The three available left mouse button functions are: move table, relate, and unrelate. The right mouse button opens a context sensitive menu.

Double-clicking on any table name in the Database Editor opens the table. You can select elements from a table opened in this manner if you opened the Database Editor from the Display process, as you did on page 5.

The right mouse button menus are available in all three left mouse button modes (Move Table, Relate, and Unrelate). You can make your selection from a right mouse button menu using either the right or the left mouse button.

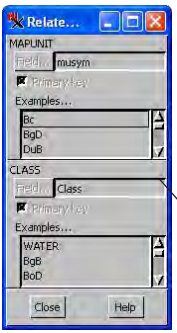
Moving tables with the left mouse button is, for the most part, a cosmetic adjustment for display purposes. It may also serve to make the endpoints of the links more apparent in a particularly complex database structure.

Table right-click menu



Opens table, just like double-clicking on box.

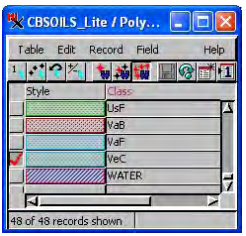
These four functions are also available when you right-click on a table in the Layout Controls or Group Controls window list.



Link right-click menu

STEPS

- ☑ with CBSOILS_LITE still displayed in a 2D group, double-click on the CLASS rectangle in the Database Editor window
- ☑ scroll to the bottom of the tabular view that opens and click in the Select Record box for the VeC class



- ☑ click on the Exclusive icon on the table's toolbar and view all the polygons with soil type VeC

- ☑ click and hold the left mouse button on the CLASS table box (Database Editor window), then drag the mouse to reposition the box
- ☑ click the right mouse button on the CLASS table box and note the menu selections
- ☑ right-click on the link between the CLASS and MAPUNIT tables and choose Show Link
- ☑ note information about the fields that relate the two tables is presented, then click on [Close] in the Relate window

Breaking Links

Vocabulary: The relationship between database records and elements in an object may be either **direct** or **indirect**. Direct relationships are made at the element level while indirect relationships are established as part of the table definition. Directly related records are referred to as **attached records**.

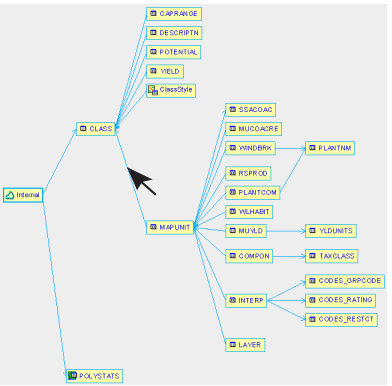
STEPS

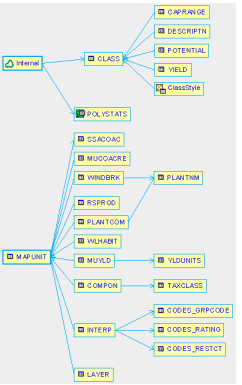
- ☑ click on the Unrelate icon in the toolbar
- ☑ click on the link between the CLASS table and the MAPUNIT table
- ☑ click on [Yes] in the Question window
- ☑ if you feel the display is too disorganized after breaking the link, right-click on the MAPUNIT box and turn on the Root Table toggle

Because all the tables in the CBSOILS_LITE polygon database are either directly attached or related through key fields, we will break one of the links and then reestablish it so you can see how relating tables in the Database Editor works. You could have tables in the database that weren't either attached to the elements or related to another table if you'd copied a table from another database (Tools / Database / Copy) or if you'd added a table in the display process without establishing a relationship at that time. (Adding tables in the display process is described in a later exercise.)

The table boxes will be rearranged when the link is broken. Note that the MAPUNIT table is still approximately in the center of the window horizontally. You can move the table box to the left of the window by turning on the Root Table toggle on the table's right mouse button menu (result shown below, right). Beside the simple left click with the Unrelate icon selected, you can also break a link by clicking the right mouse button on the link and choosing Break Link from the pop up menu in any of the editing modes.

Tables shown with links to the Internal table have records that are directly attached to the elements.





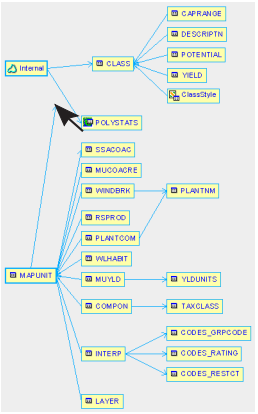
DO NOT break links that represent direct attachments to elements without first verifying that a key field relationship can be established. Links without key field relationships cannot be reestablished in the simple procedure described on the next page—you must select each element and attach the appropriate record to recover from such an error.

Relating Tables

There must be at least one table with direct attachments to the elements in an object for associated database information to be shown when elements are selected. The time required to associate database information with the corresponding elements is minimized when you can make use of indirect relationships to tables that have direct attachments.

You can relate any two tables, but for the relationship to be meaningful the two tables need to each have a field that contains the same kind of information expressed in the same way. For example, the CLASS table and MUCOACRE table both contain soil type information, but a meaningful relationship between the two can't be directly established because the soil types are appended to the county code in the latter table (BgB in one is 045BgB in the other). A sampling of field values is provided in the Relate window so you can tell if the two selected fields express the information similarly.

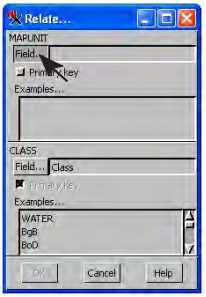
The primary key (if there is one) is provided as the best guess for the field you want to establish the relationship between tables. You just need to select the corresponding field in the table you are relating. You can of course, also change the field you are relating to if the designated primary key is not the one desired.



Click anywhere on the table you want to relate. As you drag the mouse a link appears.

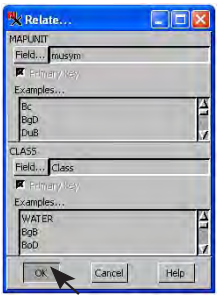
STEPS

- ☑ click on the Relate icon in the toolbar
- ☑ click and hold on the MAPUNIT table then drag the mouse until the cursor reaches the CLASS table
- ☑ release the mouse
- ☑ click on [Field] for the MAPUNIT table (upper) in the Relate window



- ☑ double click on musym in the Select Field window

- ☑ click on [OK]

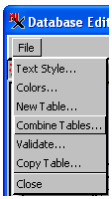


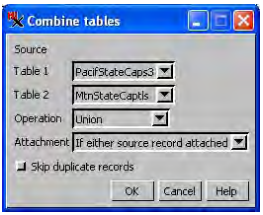
- ☑ turn off the Root Table designation for the MAPUNIT table (if you turned it on)

Combining Tables (Unions)

STEPS

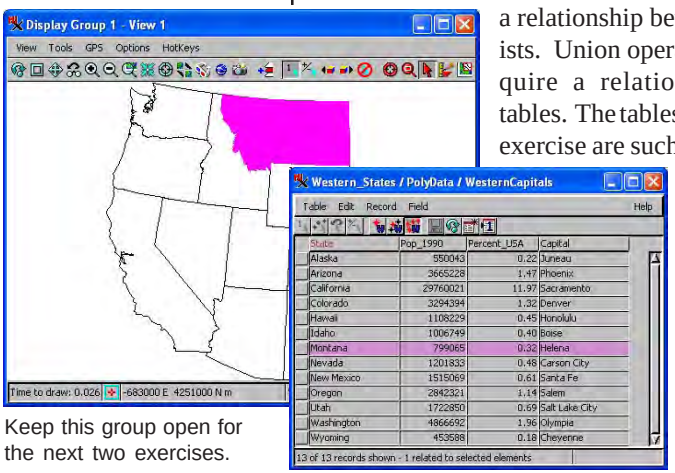
- ☑ open a new 2D Display, add the WESTERN_STATES object in the UNTDSTAT Project File, and open the Database Editor for the polygons (see page 5 if you don't remember how)
- ☑ choose File / Combine Tables in the Database Editor window
- ☑ click on [Table 1] and select PacifStateCaps
- ☑ click on [Table 2] and select MtnStateCaptls
- ☑ set the Operation to Union, Attachment to If either source record attached, and click [OK]
- ☑ name the output table WesternCapitals and enter a description if desired
- ☑ open each input table and the output table to compare contents





When tables are combined, records from the two selected tables are copied to a new table, and the original tables remain. Once you determine the new table contains the expected data with attachments, you can delete the input tables if desired.

There are six methods available for combining tables: two unions (Union, Outer Union) and four joins (see next page). Outer Union is the safer Union method because it retains all records and fields from the input tables (except duplicate records if you elect to skip them). The Union method assumes that the contents of the first field in each table are equivalent, as are the contents of the second field and so on, regardless of field names. This assumption is desirable when separate tables have equivalent data in fields with different names, such as State_Name and State. If the second field is Population in one table and Average Temperature in the other, the combined results will be incorrectly labeled (and quite misleading) for some of the records. Such data is appropriately handled by join operations, provided a relationship between tables exists. Union operations do not require a relationship between tables. The tables selected for this exercise are such that you get the same results using either union method because the records in the two tables have the same fields.



Keep this group open for the next two exercises.

Combining Tables (Joins)

Union operations are intended to add the records from one table to the records in another table. Join operations are intended to add fields from one table to existing, related* records in another table. The "outer" options take care of the less than ideal cases for both unions and joins.

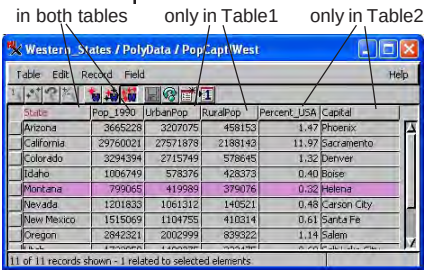
The tables selected for joining in this exercise illustrate some of the differences between a union and a join. One of the tables has records only for the western states while the other has records for all 50 states and the District of Columbia. The combined table has records for only the eleven western states. Each of the input tables has four fields, but the combined table has six fields because two of the fields are held in common (the state name and its 1990 population). The combined table has the four original fields of Table 1 plus the two unique fields from Table 2.

The Element Attachment options do not affect the information in the output table. They only affect how those records are related to the elements based on the original tables' attachments.

*Related in this case means sharing the same attribute value for a key field.

STEPS

- ☑ choose File / Combine Tables in the Database Editor window
- ☑ for Table 1 select US_POPULATION
- ☑ for Table 2 select WESTERNCAPITALS (table created in previous exercise)
- ☑ set the Operation to Natural Join, Attachment to If either source record attached, and click [OK]
- ☑ name the output table PopCaptlWest and enter a description if desired
- ☑ open each of the input tables and the output table to compare contents



combination method	number of records	number of fields	no values for some fields
Outer Union	62	6	yes
Union	62	4	no
Natural Join	11	6	no
Left Outer Join	49	6	yes
Right Outer Join	13	6	yes
Outer Join	51	6	yes

Combination Results with US_Population as Table 1 and WesternCapitals as Table 2. All key field attribute values for Table 1 are found in Table 2. Table 1 has 49 records; Table 2 has 13 records. Each input table has four fields, two of which are in common. This table is intended to test and extend your understanding of table combinations—*not all results are desirable*.

Natural join provides the most useful results for this situation, but other methods may be more appropriate with other data.

Note: The Skip Duplicate Records option doesn't eliminate duplicate records found in the original tables, it just doesn't copy records that would become duplicate records to the new table. This option applies only to unions since joining does not create duplicate records.

More About Joining Tables

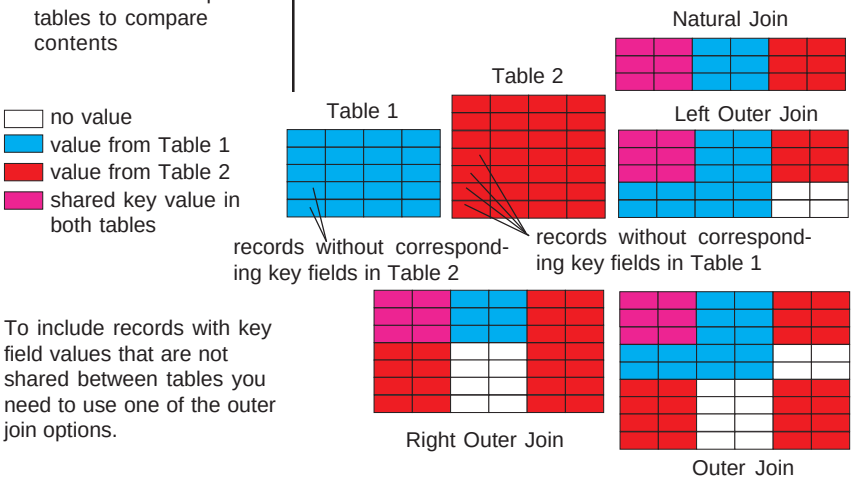
STEPS

- ☑ choose File / Combine Tables
- ☑ for Table 1 select US_POPULATION
- ☑ for Table 2 select WESTERNCAPITALS
- ☑ set the Operation to Left Outer Join, Attachment to *If both source records attached*, and click [OK]
- ☑ name the output table LeftOuterJoin
- ☑ repeat steps 1–3, set the Operation to Right Outer Join, Attachment to *If both source records attached*, and click [OK]
- ☑ name the output table RightOuterJoin
- ☑ repeat steps 1–3, set the Operation to Outer Join, Attachment to *If both source records attached*, and click [OK]
- ☑ name the output table OuterJoin
- ☑ open each of the input tables and the output tables to compare contents

The order in which the tables are selected for a natural join does not affect the number of records in the output tables. It does, however, affect the order in which the fields appear with all fields from Table 1 appearing before the unique fields in Table 2. The order in which the tables are selected does affect the number of records in the output table for the outer join operations. The number of records in a natural join is restricted to the number of records that have the same attribute value for the primary or foreign key field.

The left outer join option combines the unique fields in Table 2 with the records in Table 1. The right outer join option combines the unique fields in Table 1 with the records in Table 2. The outer join option uses all records from both tables. The field order is the same for all three outer join choices.

The number of records in the left outer join result is the same as the number of records in Table 1. The number of records in the right outer join result is the same as the number of records in Table 2. The number of records in the outer join result is the total number of records in Tables 1 and 2 minus the number of records in the natural join result.



To include records with key field values that are not shared between tables you need to use one of the outer join options.

Copying Tables from Other Databases

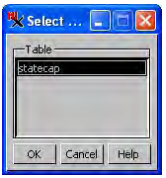
The Copy option on the File menu in the Database Editor lets you select a table from any database in RVC format including linked databases. Linked databases are databases that remain in their original format that also have an associated RVC file created by TNT's Import process. The link object contains information that allows TNT to read the database and also any additional objects created when the table is in use in the TNT products, such as style objects, style assignment tables, and layout information for the Database Editor. TNTmips links to many types of external databases including Access, MySQL, and ESRI Personal Geodatabase. Linking allows you to work with the database in the TNT products and in the software package that created the database.

In this exercise you select a table from a linked dbf file. The copy process creates a new table in RVC format that is part of the polygon database in this case. There is no longer a connection between the original dbf file and this table. You would need to establish a link in the Import process if you want the table to remain a linked table.

You could also use the procedure from the exercise on page nine to relate these two tables. This exercise gives you a different means of reaching the same end. The means of duplicating a table already in the database of interest is discussed in the next exercise. Choosing Copy from the File menu in the Database Editor is the equivalent of choosing Add Table from the right mouse button menu for the database in the Display Manager.

STEPS

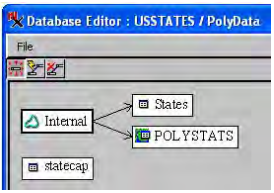
- ☑ create a new 2D group and select the USSTATES object from the UNTDSTAT Project file
- ☑ choose File / Copy
- ☑ select the table in the STATECAP database and click OK



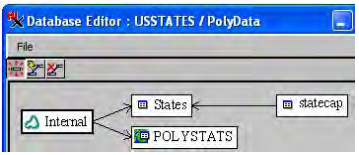
- ☑ right-click on the newly added table and choose Properties from the menu
- ☑ on the Table tab verify that the attachment type is set to Related Only
- ☑ on the Field tab with the State field highlighted, set Relate to Key to States.STATENAME

- ☑ click on OK

Keep this object and the Database Editor open for the next eight exercises.



after table is added

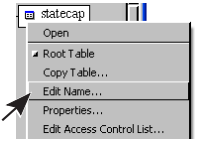


after relationship is established

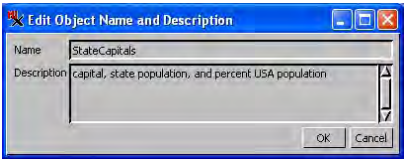
Renaming and Copying Tables

STEPS

☑ click the right mouse button on the box for the table just copied, select Edit Name and change the name to StateCapitals

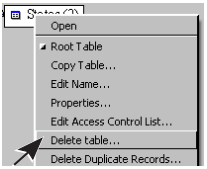


☑ type in *capital, state population, and percent USA population* in the Description field, then click [OK]



☑ click the right mouse button on the States box, choose Copy Table, accept the default name, and click [OK]; note the attachment of this table directly to elements as with the States table

☑ click the right mouse button on the copied table (States1), choose Delete Table, and click [Yes] in the Question window



You may want to change a table's name and / or add a description. The description is not evident in the Database Editor but is shown when tables are listed in the Display Manager. You can also edit a table's name and description by right-clicking on the entry for that table when tables are shown in the Group or Layout Controls window.

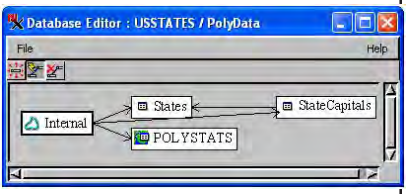
The Copy Table option on the right mouse button menu for a table in the Database Editor creates a duplicate of the table the option is selected from. The attachments in the copy are also as in the original. This utility is useful when you want to create another table that contains at least one of the same fields with the same field values as in the original. For example, you could create a second table that contained more information about each state without having to enter all the state names again. A similar feature is available in Tools / Database / Copy, which also lets you copy a table to a different database. However, direct attachments to elements are not maintained by this latter process as they are with the Copy Table function in the Database Editor.

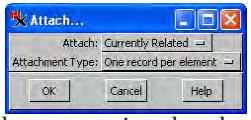
You could also use this function to make a table with many fields more readily readable in tabular view by creating two (or more) tables with fewer fields to eliminate horizontal scrolling. You would simply copy the table, then use the Edit Definition option to delete different fields from each table, being sure to maintain the primary key field in both cases.

The delete table function is available from the right mouse button menu over a table entry in the Display Manager and from the Table menu when viewing the table. The Delete choice is inactive for linked tables.

Changing Related Only to Directly Linked

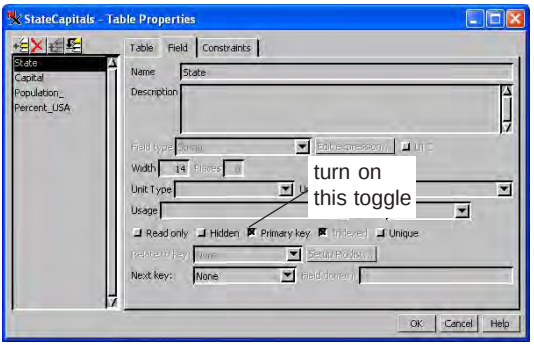
Database information often evolves over time with new tables being added and some information becoming obsolete. Vectors are often initially acquired with a single table containing only one or a few attributes. When new tables are added, they can be immediately related if they contain the same information found in the primary key field of the initial table. If you continue obtaining new information, it is likely at some point that all the information or all the information you care about in the initial table is duplicated in the more recently acquired table(s).





For this exercise we make the assumption that the State Code is not of interest. The original table (States) had a record for every polygon (370). The table you imported has only 50 capital/state records. For those states with multiple polygons (for example, Florida and Louisiana), the single record for that state is attached to all the appropriate polygons.

When the States table is deleted, the foreign key status of the State field in the StateCapitals table is removed. Make this field a primary key so that it can be used as a foreign key by subsequently added tables.



STEPS

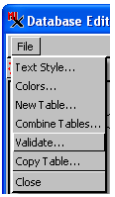
- ☑ continuing from the previous exercise, click on the Relate icon
- ☑ click in the Internal box and hold the mouse button while dragging to the StateCapitals box; when you reach the box, release the mouse button
- ☑ in the Attach window, set Attach to Currently Related and the Attachment Type to One Record Per Element, and click OK
- ☑ right click on the States box and choose Delete Table
- ☑ click Yes in the Question window
- ☑ right click on the StateCapitals table name in the Display Manager, and choose Mark All Unattached Elements
- ☑ note that the single unattached element is the District of Columbia, which does not have a record in the StateCapitals table
- ☑ right click on the table name again and choose Properties
- ☑ on the Field panel with the State field highlighted, click on the Primary Key toggle, then click OK

Database Validate and Attachment Types

STEPS

☑ continuing from the last exercise, choose File / Validate in the Database Editor window

Just as with any work in progress, a little clean up is needed from time to time. TNTmips provides a Vector Validate process that checks and corrects vector topology. There is also a means to validate databases. The Validate feature of the Database Editor makes sure that all relationships in the database point to tables and fields that actually exist, rebuilds all indexes, and cleans up attachments by eliminating any direct element-to-record attachments that are invalid (see the sidebar for an explanation of invalid attachments).

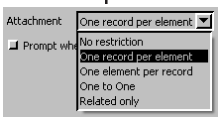


It may be helpful to run database validate whenever you change a table's attachment type, remove a table, or delete any records. Since we have changed the attachment type of a table and deleted a table, now is a good time to run validate.

Invalid attachments are:

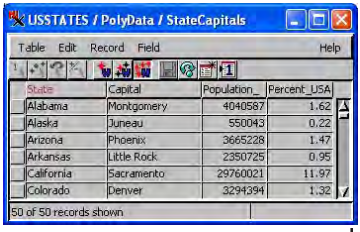
- attachments to implied one-to-one tables (a single attachment is valid, but because of the table's attachment type, there's no reason to store them)
- attachments to related only tables
- attachments to invalid or deleted records
- attachments to deleted tables
- redundant attachments

Often when a vector has been imported, the attachment type for tables in its databases is "No Restriction." This option may seem like the easiest choice since you do not need to decide what the relationship between the elements and database records will be. It does, however, have consequences if you edit, merge, or use vector combinations for example. Any operation in these processes that splits elements will lead to the proliferation of records. Other than unnecessarily increasing the size of the databases, this duplication of records creates a management problem. What if it turns out there is a typographical error or the information in a field needs to change for some other reason? Instead of changing it in one place as you would for a Related Only or One Record Per Element case, you need to find all occurrences of the entry and change them. For more about attachment types, see the *Managing Geoattributes* booklet.



Additional Database Editor Functions

There are a number of other database management features available from the File or right mouse button menus in the Database Editor. Some of these are also available in the table row of the Display Manager or on the Table menu (tabular and single record view), and some are available only in the Database Editor window or as part of some other process, such as Vector Extract, Vector Combine, or TNT's Editor.



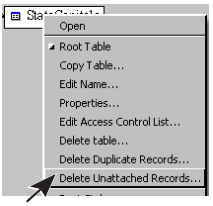
In addition to the Delete Unattached Records option, which is demonstrated in this exercise, you can choose to delete duplicate records. Duplicate records have identical attribute values. When you remove duplicate records, all elements previously attached to any one of the records are attached to the remaining record. If the attachment type for the table is One Element per Record, the Delete Duplicate Records option is unavailable because the table's attachment type prevents the elements attached to the duplicate records from being attached to the remaining record. You will need to change the attachment type before duplicate records can be deleted. You can also create new tables while using the Database Editor (File menu). This last feature is also available in the Display Manager.

Recall there are two ways to get to the Database Editor: Tools / Database / Edit and from the table row of the Display Manager in any process that displays geometric objects. The features are the same regardless of which way you open the Database Editor, except there is an Open Database choice on the File menu in the standalone version, which lets you work on a number of different databases one after another.

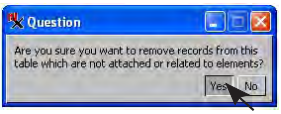
STEPS

- ☑ double-click on the StateCapitals box or right-click on the box and choose Open
- ☑ click on the View All Records icon in the table and note the Alaska record, which is not represented in the vector object (also Hawaii)

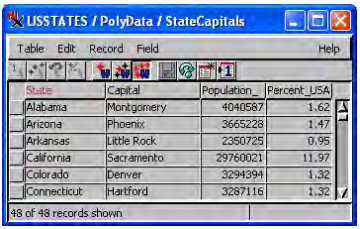
- ☑ right-click on the StateCapitals box in the Database Editor window, choose Delete Unattached Records



- ☑ click on [Yes] in the Question window



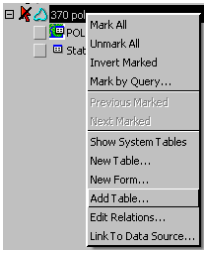
- ☑ note the absence of the record for Alaska (and Hawaii)



Adding Tables

STEPS

- ☑ with USSTATES from the UNTDSTAT Project File still open from the previous exercise, right-click on the polygon row in the Display Manager, and choose Add Table



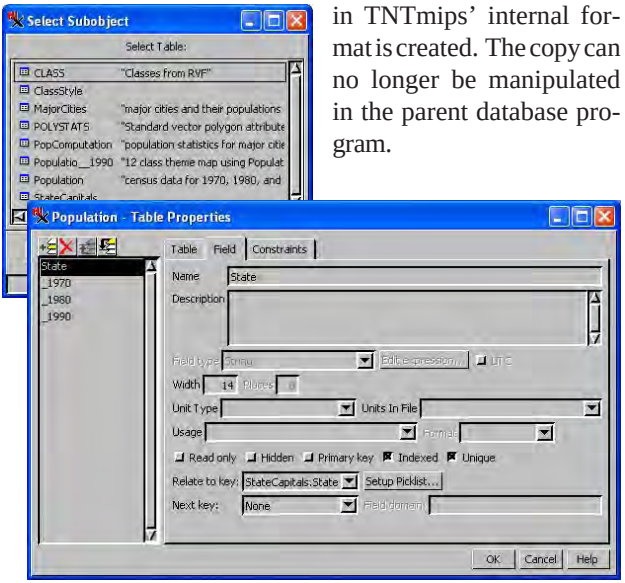
- ☑ select USA in the USA_OLD Project File, then select PolyData
- ☑ select Population in the Select Subobject window
- ☑ choose StateCapitals.State from the Relate to key option menu (Field panel)
- ☑ set the Attachment Type option to Related Only (Table panel), then click [OK]
- ☑ repeat steps 1–5 substituting Major Cities in step 3
- ☑ repeat steps 1–5 substituting PopComputation in step 3

Not all tables are brought forward from this old vector object because they may be incomplete or object specific.

There are a variety of reasons to copy a table from one database to another. In this example, an old vector object (USA state outlines created by raster-to-vector conversion of a low resolution, poorly georeferenced scanned map) has a number of associated tables with population and cultural information. A higher resolution outline map with good georeferencing has since been acquired, but the only originally associated polygon attributes are the state codes and names (standard attributes tables for the lines and polygons have also been calculated).

Adding tables is initiated from the right mouse button menu for the database in the Display Manager. Bringing in tables with computed fields (as we do in the last step of this exercise) is generally not recommended unless you are sure that all referenced tables and fields also exist in the recipient database. The order in which tables are added in this exercise ensures that the necessary tables and fields already are found in this database.

When you add a table from a linked database, a copy in TNTmips' internal format is created. The copy can no longer be manipulated in the parent database program.



Database Viewing

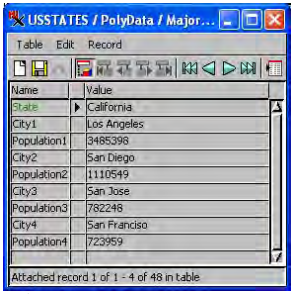
TNTmips provides two methods of database viewing: tabular and single record views. Tabular view presents all the records, or only records attached to selected elements, as rows in a table in which the columns are the fields. Single record view shows the fields for a single record. You can switch between view types from the Table menu in either view or from the icon on the tables' toolbar.

Single record view is designed for viewing all information about one element at a time and for data entry. Single record view works best for data entry because a new record is created automatically whenever you click on an element that lacks a related record. It also avoids the problem of accidentally entering information that actually belongs in the row above or below that is often encountered when entering information in a table.

Tabular view has a number of features not found in single record view because they are related to viewing multiple records. You have the ability to add statistics and substatistics rows to the table. You can assign separate colors to records and statistics rows and interleave two colors for easier record identification. The ability to save selected records or statistics records as a new table is available in tabular view. You can view all records in a tabular view or just those records associated with the active element or with all marked elements. You can also select elements by their attributes from a tabular view (as described on page 7).

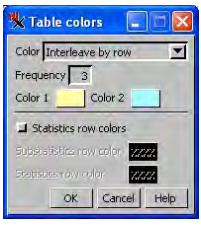
STEPS

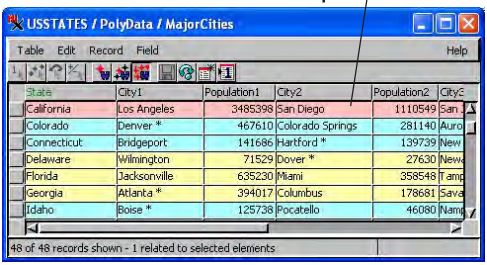
- ☑ click on the Select icon in the View window's toolbar
- ☑ open the MajorCities table and switch to single record view (icon or Table menu)
- ☑ click on California in the View window



- ☑ review the information presented, then click on the Switch to Tabular View icon
- ☑ click on the View All Records icon
- ☑ choose Colors from the Table menu
- ☑ set the Color option to Interleave By Row and the Frequency to 3, pick pale colors for Color 1 and 2, then click [OK]

This color indicates record is related to a marked element.



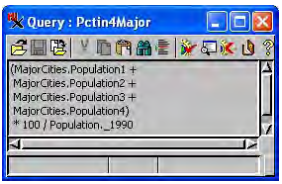


Note that you need to scroll horizontally to see all fields in this table.

Virtual Fields

STEPS

- ☑ open the PopComputation table
- ☑ choose Table / Properties, click on MajorCity1 in the field list, note that the Field Type is String Expression, and click on [Edit Expression]
- ☑ note that the expression simply pulls the City1 field information for the related polygon from the MajorCities table, then click [OK] in the Query window
- ☑ click on Pctin4Major in the field list, note that the Field Type is Computed, and click on [Edit Expression]
- ☑ note the arithmetic combination of fields from two different tables to derive the percentage of a state's population living in its 4 largest cities



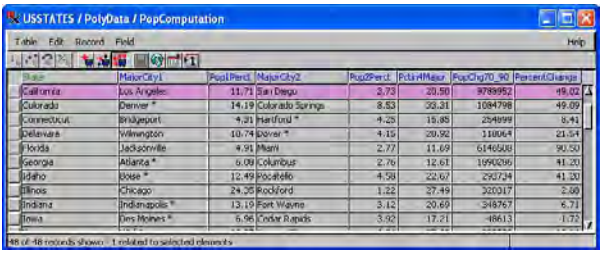
- ☑ click [OK] in the Query window and [Cancel] in the Table Properties window

Virtual, or computed, fields let you do computations with the information in a table and also bring in information from other tables. You added a table with computed fields in the exercise on page 18. We'll examine the expressions for those fields here. You create a new table with computed fields in the next exercise.

There are two types of virtual fields: those that incorporate string fields, which are referred to more specifically as *string expression fields* , and those that use numeric fields, which are referred to by the more general term *computed fields* . The PopComputation table also contains one field with directly entered data, the State field. The PopComputation table was created before the Implied One-To-One option existed. This option eliminates the need for any direct data entry as demonstrated in the next exercise. Names for virtual fields are shown in blue, which makes it easy to distinguish them from fields for direct data entry.

String expression fields may simply supply the information from a string field in another table or can be a concatenation of other string fields, such as LastName, FirstName from another table that contains first and last names in separate fields. Any operations available in database query can be used on numeric fields to generate the desired results.

You can also edit the expressions for individual virtual fields by right-clicking on the field in tabular view, selecting Field Options from the popup menu, and clicking on [Edit Expression].



Creating New Tables

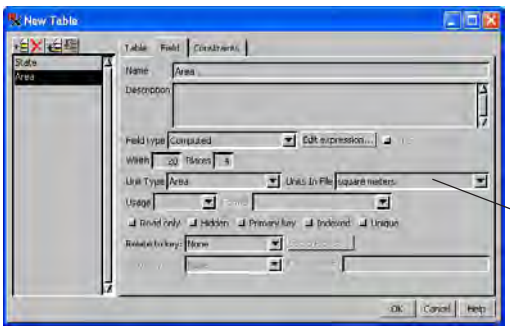
Choosing New Table creates a new, blank table. After you add and define the fields, there are two tasks to complete before the table can be useful in attribute retrieval for selected elements: data entry and establishment of a relationship between elements and records. Your job is greatly simplified if another table in the database contains a field that can be identified as a foreign key; you just set the attachment type to Related Only and the second task is completed. You still, however, have to enter the attribute values for the foreign key field in the new table and for any other fields that are not computed fields. You can get around data entry altogether if the Implied One-To-One option is appropriate for the record attachment type and the table is comprised entirely of computed fields as in this example.

When you add a field to a table, it gets a default name, which you change by selecting the name and editing it directly in the field list. The field list and associated icons are visible in the Table Properties window regardless of the active panel. Most of the work for this exercise is done from the Field panel, but the attachment type is set on the Table panel.

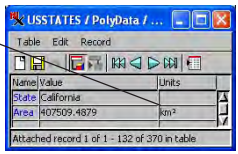
A units field is provided in single record view next to each field that has units set. A units selection window opens when you click on this field. The units used can be changed in tabular view by right-clicking on the field name and choosing Field Options.

STEPS

- ☑ right-click on the polygon row and choose New Table
- ☑ check that User Defined is selected, click [Next]
- ☑ enter StateAreas for the table name, click [Next]
- ☑ choose *One record per element, record number equals element number* from the menu, and click [Next] twice
- ☑ click on the Add Field icon, change the field name to State, set the Field Type to String Expression and Width to 16 (Field panel)
- ☑ click on [Edit Expression], click on the Insert Field icon, select StateCapitals as the Table and State as the Field, then click on [Insert]
- ☑ click [OK] in the Query window
- ☑ click on the Add Field icon, change the field name to Area, Field Type to Computed, Width to 20, Places to 4, and Unit Type to Area with Units in File as square meters
- ☑ repeat step 6 but choose the POLYSTATS table and the Area field, and click [OK]



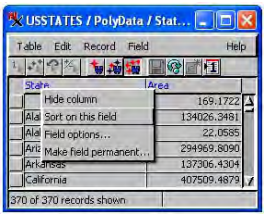
You can change the units for viewing if the units for a field are set in the table definition.



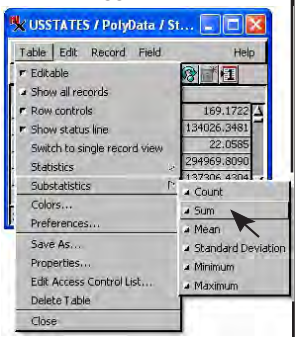
Statistics and Substatistics

STEPS

☑ click on the Switch to Tabular View icon if not in tabular view for the State Areas table already, click on the View All Records icon, then right-click on the State column heading, and choose Sort on this field



☑ choose Table /Substatistics and turn on the Sum toggle button



☑ choose Table / Colors and turn on the Statistics Row Colors toggle button

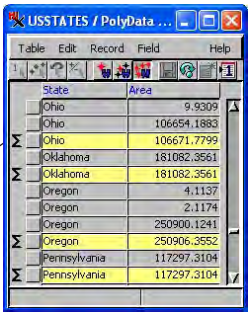
☑ click [OK] in the Table Colors window

Substatistics occur each time the attribute value in the sorted field changes. You could also add a count substatistic and easily see how many polygons make up each state.

The USSTATES vector object does not have a one-to-one correspondence between states and polygons; 48 states and the District of Columbia are represented by 370 polygons. About half the states are a single polygon, but states with intricate coastlines may be made of many polygons. (Florida has the most with 57 polygons.) The purpose of generating a table with the state name and area drawn from other tables is to get areas for states as a whole, then use these areas to compute population densities.

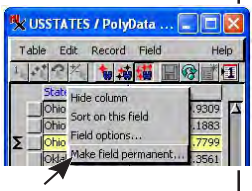
TNTmips offers six mathematical functions that can be applied to all records in a table (statistics) or to all records with the same attribute value in the field used for sorting (substatistics). These six functions are count, sum, mean, standard deviation, minimum, and maximum. Including a sum substatistic row in the table you made in the last exercise will provide total areas for each state.

A symbol identifying the type of statistic is placed at the left of the table. The Select Record box is active for substatistics rows and selects all records (and attached elements) included in calculation of the selected substatistic value. Turning on color for the substatistics rows helps to distinguish these rows from the data. Statistics and substatistics are turned off by returning to the menu and turning off the corresponding toggle.



New Tables from Summary Statistics

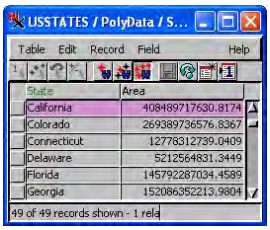
The Save As feature for database tables is useful in a variety of contexts. You can choose to save the data or the statistics for all records, selected records, or records attached to selected elements*. We will use this feature to create a table that contains state names and area totals. Each record in such a table is attached to all the elements that contributed to the substatistic calculated. The table created can be in RVC format, in dBASE III format, in a format you've identified by ODBC, or as comma separated or tab delimited values in a text file if you are running any TNT Pro product. Internal database format is the only option if you are running TNTmips Basic or TNTmips Free.



Because the table you are going to save substatistics from has an Implied One-To-One Attachment Type, which will no longer be the case after the Save As, you need to do a little database manipulation in preparation. The two computed fields need to be made permanent, which poses no problem for future flexibility since neither the state name or area are likely to change. The Attachment Type also needs to be changed.

You specify the format, which records, and what values to save.

When the table is saved to an existing vector object, you need to specify which database should include the table.



The resulting table has one record per state, which is related to all the state's polygons and provides the state's total area instead of just the area for the selected polygon.

* You can also save in other formats or create a CAD snapshot (the latter is available in TNTmips Basic and Free), which is discussed in the *Managing Geoattributes* and *Making Map Layouts* booklets.

STEPS

- ☑ right-click on the State field heading in Tabular View and choose Make Field Permanent, then click on [Yes] in the Question window
- ☑ repeat step 1 except right-click on the Area field heading
- ☑ choose Table / Properties and select StateCapitals.State on the Relate to key option menu (Field panel)
- ☑ set the Attachment Type to Related Only (Table panel), and click [OK]
- ☑ choose Table / Save As
- ☑ check that the Format and Records option menus are set to RVC and All Records, respectively
- ☑ set the Values option menu to Sum, the Units to As Stored, and click [OK]
- ☑ navigate to the USSTATES object you have selected for display and select it for the destination object, turn on the Polygon toggle in the Select window, accept the default name (State Areas1), and click [OK]
- ☑ click on the View Table icon for StateAreas1

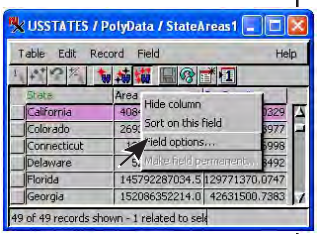
Modifying An Existing Table

STEPS

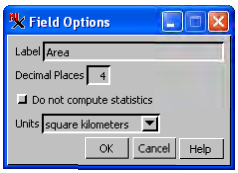
- ☑ choose Table / Properties in the StateAreas1 table
- ☑ highlight Area in the fields list then click on [Add Field]
- ☑ change the new field name to PopDensity
- ☑ choose Computed from the Field Type option menu
- ☑ set the Width to 6 and Places to 1
- ☑ set the Unit Type to None
- ☑ click on [Edit Expression] and enter the expression as shown at the right (you can use Insert / Field to get the names correct without typing if desired)

```
Population._1990 /
StateAreas1.Area *
1000000
```

- ☑ click [OK] in the Query window and [OK] in the Table Properties window
- ☑ right-click on the Area field name in tabular view then select Field Options



- ☑ set the units menu to square kilometers



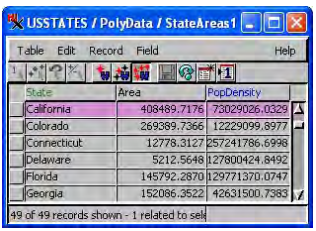
- ☑ click [OK] in the Field Options window

Now that we have the total area for each state in data form, we can calculate population density. (Statistics and substatistics fields cannot be referenced in a computed field expression because their values change depending on the records currently viewed.) The query shown will give population per square kilometer. Add an additional term to the query that divides by 0.3861 if you want the population density expressed as people per square mile.

Since the computed field expresses population density in square kilometers, it makes sense to also display the area in square kilometers. Recall that in the exercise on page 18 in which the initial StateAreas table was created, we specified that the area field was expressed in square meters (as it is in all standard attributes tables calculated in TNTmips), which enables you to change units for display. In single record view, the units option button is next to the field. You set the units in tabular view by first selecting the field (click on the field name), then choosing Options from the Field menu (or right click and select Field Options). You can also set whether or not to calculate statistics for the field if statistics are included in the table. If the field selected is a computed field, there is also an Edit Expression button that opens the Query Editor window with the current expression shown.

The District of Columbia does not have an entry in the State or PopDensity fields because the District of Columbia is not in the StateCapitals table.

Note that areas are shown in square kilometers at the right on this page and in square meters in the table on the preceding page, but are stored as square meters for both.



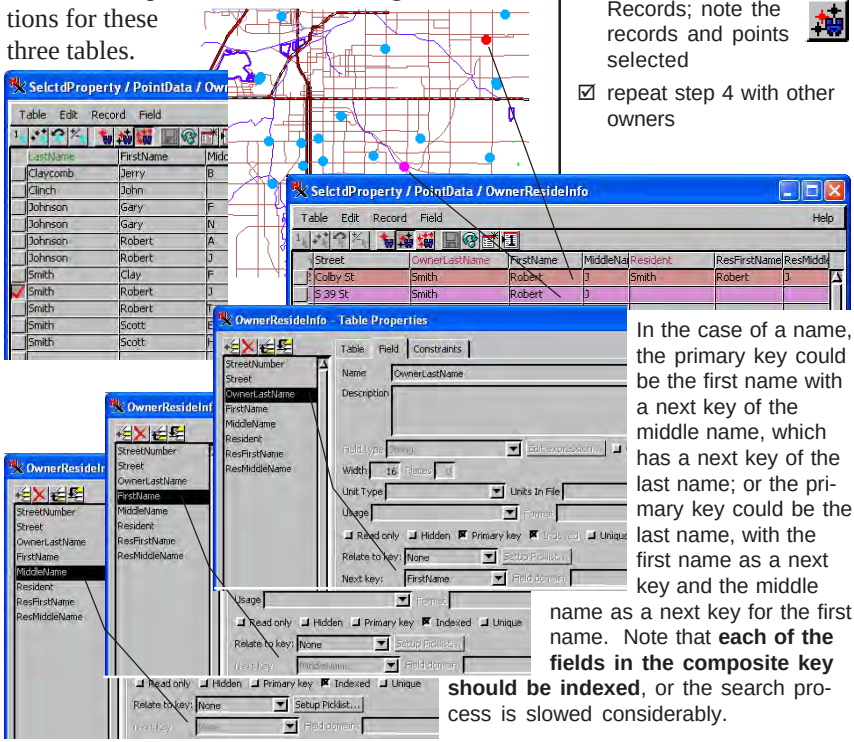
Composite Fields

Composite key fields allow unique identification of records that may not have any individual unique fields but have unique combinations of fields. In the property information database used in this exercise, property owner, resident, and street address are represented by composite fields. So while Smith is the most popular last name in this sample database, you can readily select only the properties owned by Robert J. Smith and not those owned by Robert T. Smith or Robert J. Johnson.

The tables in the point database use other important relational database features, such as multiple primary keys so related elements can be selected by owner name or by address and sorting on multiple fields so that addresses can appear in numerical order after sorting alphabetically on the street name. You should spend some time looking at the definitions for these three tables.

STEPS

- ☑ choose New / 2D Display and select the two vectors in the PROPINFO Project File
- ☑ expand the information for the SELCTD PROPERTY object to show the tables in the points database
- ☑ open the OWNERRESIDEINFO, OWNER VALUE, and OWNER tables
- ☑ click on the Select Record box for Robert J. Smith in the Owner table (View All Records), then click on the Exclusive icon
- ☑ set the other two tables to View Selected Element Records; note the records and points selected
- ☑ repeat step 4 with other owners



In the case of a name, the primary key could be the first name with a next key of the middle name, which has a next key of the last name; or the primary key could be the last name, with the first name as a next key and the middle name as a next key for the first name. Note that **each of the fields in the composite key should be indexed**, or the search process is slowed considerably.

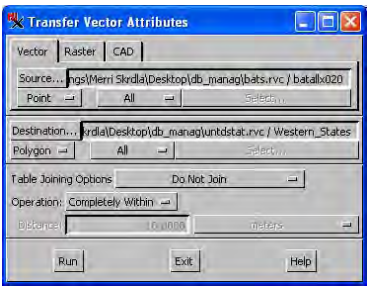
Other Ways to Add Tables

STEPS

- ☑ choose Geometric / Attributes / Transfer Attributes
- ☑ click on [Source] and select the vector object in the BATS Project File
- ☑ click on [Destination] and select the Western_States object in the UNTDSTAT Project File
- ☑ check that the destination element type is set to Polygon
- ☑ set the Table Joining Options to Do Not Join and the Operation to Completely Within
- ☑ click on Run
- ☑ add the Western-States layer to a new 2D group in the Display process
- ☑ examine its new database table
- ☑ click on the select tool, select the California polygon, and note that multiple records are attached (each representing a different bat species and/or county)

There are a number of ways to get database tables associated with objects in Project Files. Vector and TIN objects have a separate database for each of the element types. CAD, shape, and raster objects have a single database. Some TNT processes, such as Automatic Classification, generate tables automatically. The processes discussed here are those in which you select a source and destination database.

The Import process lets you bring a number of data types into Project Files as main level databases or as subobjects. You can link to many of these formats (more on import and linking in later exercises). You can use File / Copy Table in the Database Editor or Add Table on the right mouse button menu (RMBM) for a database in the Manager window. The Link to Data Source option on the same RMBM lets you add tables from established ODBC data sources. You can make a duplicate of a table from its RMBM in the Database Editor, which retains the same relationships as the original. The Transfer Attributes and Database Attach processes make direct attachments between elements and the records in the selected table by geographic position. There are also a number of processes that make specific calculations based on elements in a selected geometric object (see the Geometric / Attributes menu cascade). You can copy databases and tables from one TNT object to another using Project File Maintenance or Tools / Database / Copy.



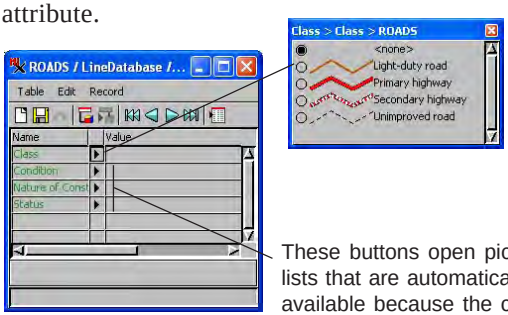
Transfer Attributes is particularly useful because you can move data from one element type to another. The most likely transfer is from points to polygons or vice versa, but you can also transfer point data to the nearest line (and vice versa) and line data to polygons. Another process (Tools / Database / Attach makes use of X and Y values in the database to attach records to the nearest elements in the selected geometric object.

Picklist Wizard

A picklist provides one-click selection of any of the allowable string values for a particular attribute associated with one or more elements in a geometric object. The TNT products provide a wizard to set up a picklist using the values in an existing table. You can also create a new table to provide the picklist using the picklist wizard. You can create picklists for databases associated with geometric objects in Project File format and also in any format supported for direct use or linking, such as shapefiles and PostGIS. Picklists are opened from the single record view window. Picklists are automatically available for any database table string field with a foreign key relationship to another table.

The picklist creation wizard is launched from the Table Properties window for the table you want to use the picklist with. When creating a picklist from an existing table, that table's primary key field is offered as the default choice for the field to provide the picklist. There is no reason to create a picklist from an existing table unless you will be adding elements to the geometric object with which it is associated or you will be reassigning attributes.

After creating the picklist table, you can automatically proceed to style assignment. The styles you set up will be shown in the picklist and will match the styles used for drawing if the same styles are assigned and the geometric object is being styled by attribute.

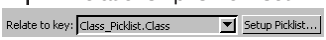


These buttons open picklists that are automatically available because the corresponding table is related-only with a primary key that is pointed to by the field in the table illustrated.

- ☑ choose Main / Display, create a new 2D group and select the ROADS object in CB_DLG.RVC
- ☑ expand the layer in the Display Manager, right-click on the CLASS table entry, and select Properties
- ☑ click on the Setup Picklist button on the Field tabbed panel

- ☑ accept the default table name by clicking [Next]
- ☑ accept the single field (Class) offered by clicking [Next]
- ☑ accept the values offered for that field by clicking Finish
- ☑ click on the Lines icon in the Assign Styles by Attribute window

- ☑ click on the Styles button and select the style object under the RoadClassStyle table
- ☑ assign the styles to the same named attribute values by clicking on the style name in the left column then on the double arrows for the corresponding entry in the right column
- ☑ note when you are done that the appropriate key relationship is now set



- ☑ open the class table in single record view and click on the arrow to the right of the class field

Field Constraints

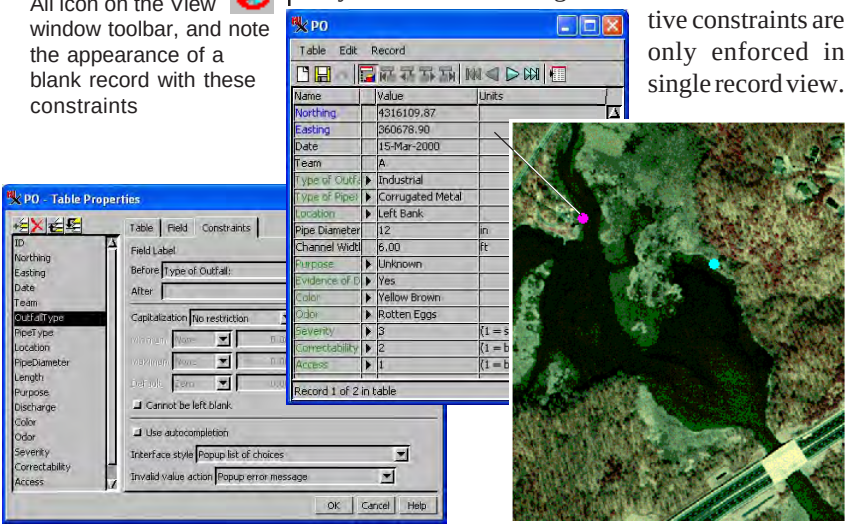
STEPS

- ☑ click the Open Display on the Display Manager toolbar, and select DATALOGGER from the CONSTRNT Project File
- ☑ click on the Select icon in the View window, then click on one of the two points in the pinmap overlay
- ☑ choose Properties from the Table menu in the PO table window
- ☑ select the OutfallType field, then click on the Constraints tab and note the options
- ☑ click on the Field tab and note that this field points to a primary key in another table
- ☑ check the constraints for other fields
- ☑ click on [Cancel]
- ☑ click on the Deselect All icon on the View window toolbar, and note the appearance of a blank record with these constraints

You may anticipate untrained personnel, such as temporary help, entering values into a database you have designed. You may also need to enter values without a keyboard under field conditions. Placing constraints on the values allowable for database fields lets you guide anyone entering data values to the choices acceptable to the database designer, which increases the accuracy and reproducibility of observations.

If all field values are constrained to be choices from a list, the need for a keyboard to fill in observations is eliminated. Other tables that contain all of the list choices must be in the database to enable you to pick from a list. There are eight tables that supply the list options for the PO (pipe outfall) table. The values you want to choose from need to be in a primary key field so it can be selected as a foreign key for the field that is constrained. You can also select values without the aid of a keyboard using picklists.

Choices provided by setting up constraints are available in both single record and tabular view whereas the picklists described in the previous exercise are only available from single recoved view. Restrictive constraints are only enforced in single record view.

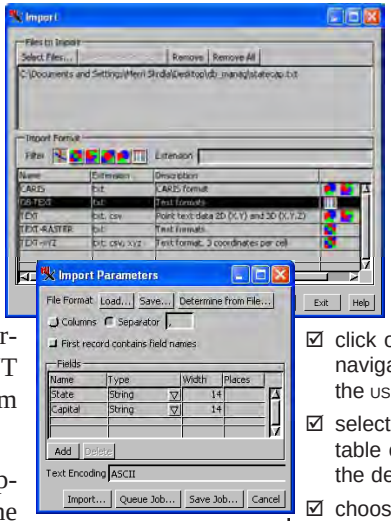


Importing Text Files to Internal Database

We are going to import a text file that lists states and their capitals, separated by commas. When you import from text you need to specify the format of the text file in relation to the information it contains. You can either save this information as a format file that can be loaded and used again or specify one for temporary use. One has been created for your use in this exercise (you can see how it is set up by clicking on the Edit button to the right of the temporary toggle, but leave the toggle off). Some database formats can also be linked to (Link button will be active), which means they are maintained in their native format and can be used by TNT products and the program that created them.

STEPS

- ☑ choose Main / Import in TNTmips or Tools / Import in TNTview
- ☑ click on [Select Files] and select STATECAP.TXT from the DB_MANAG data collection

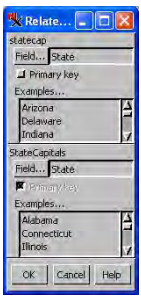


- ☑ select DB-TEXT in the Import Format panel and click on [Next]
- ☑ click on [Load] in the Import Parameters window and select STATECAP.FMT
- ☑ click on Import and navigate to PolyData in the USSTATES object
- ☑ select a new database table object and accept the default name
- ☑ choose Tools / Database / Edit and select USSTATES and polygons
- ☑ click on the Relate icon, place the cursor on the STATECAP box, drag to the State Capitals box, and release

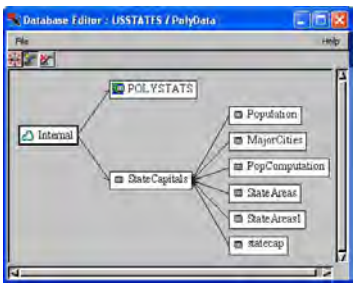
- ☑ check that State is selected as the Field for both tables, then click [OK]

Imported tables initially appear at the left edge of the Database Editor window with no attachments indicated. Once you set up the foreign key relationship, the table will be aligned with the others that use STATECAPITALS.STATE as a foreign key.

You need to select the fields in both tables because no prior relationships have been established.



Your database tree should have the same tables and attachments shown when you've finished the exercise.



OLE DB and ODBC

ODBC stands for Open Database Connectivity. **OLE DB** stands for Object Linking and Embedding for databases.

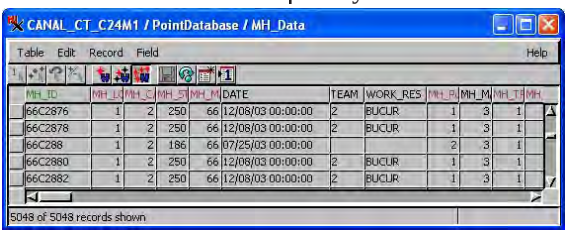
ODBC and OLE DB are both connection methods that provide acess to databases in a variety of formats. These connection methods are available only on Windows platforms.

ODBC is the older of the two connection methods and requires that you set up a data source using your Windows operating system (see the next exercise). OLE DB is more efficient, stable, and often several times faster than an ODBC connection. It does not require any initial setup. Additionally, OLE DB allows random access and relational linking to external database structures that lack a primary key.

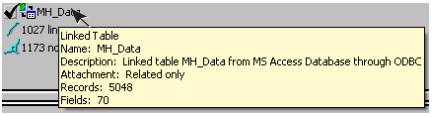
The next three exercises apply to ODBC data sources. No setup is necessary for OLE DB. For formats supported by ODBC and OLE DB, it is recommended that you choose the latter.

TNTmips supports import and linking to Microsoft Access and Excel files using either ODBC or OLE DB. Import and linking to Oracle and Import of Oracle Spatial layers is supported only through ODBC. Import and linking to SQL Server databases is supported only through OLE DB. The TNT products also directly support linking to dBASE III/IV, FoxPro, PostgreSQL and LiDAR point databases. As a result, these formats can also be imported or linked to on the Mac. Linking and import of MySQL files is also supported directly but is available only for Windows.

You can pinmap linked tables directly if they include spatial coordinate fields. Associating linked fields with elements in a geometric object requires that you establish a relational link between a table directly attached to the elements and the external table.



Tabular view of a linked Access database table.



DataTip for the same linked Access table.

Setting Up ODBC on Your System

ODBC is a cross platform interface specification that gives application software, such as TNTmips, direct access to database tables in many different formats. Vendors of host DBMS software (such as dBASE, Microsoft Access, and Oracle) provide ODBC drivers with their software. You need to define one or more ODBC Data Sources at the Operating System level before you can use ODBC data sources in the TNT products.

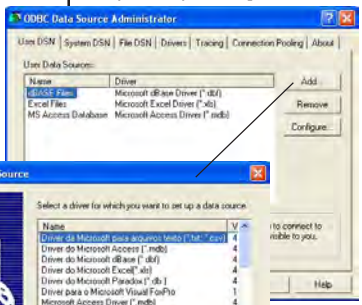
Windows 95 or later gives you ODBC configuration control through an ODBC control panel. The Administrative Tools folder in your PC's Control Panel contains an ODBC entry if ODBC drivers have been installed on your computer. Go back to the setup program for your DBMS software and look for the ODBC installation option if you don't have this control panel.

There is little point in setting up an ODBC data source for a dBASE IV or FoxPro database since TNTmips can link directly to these formats. The procedure is essentially the same for all ODBC drivers. It is likely your DBMS software has a sample data file you can use to run through this exercise, such as the Northwind Sample Database that comes with Microsoft Access, if you haven't already created one of your own.

You can use the Select button in the Setup window to have your choice point you to an individual file, however you will have to define a source for every file if you do.

STEPS

☑ select Control Panel from your Windows XP Start menu and double-click on Administrative Tools then on the DataSources (ODBC) icon [for



Windows Vista, double-click on the System and Maintenance icon to get to the Administrative Tools folder]

☑ click on [Add] in the ODBC Data Source Administrator window

☑ double-click on the Microsoft Access driver or another driver of your choice in the Create New Data Source window

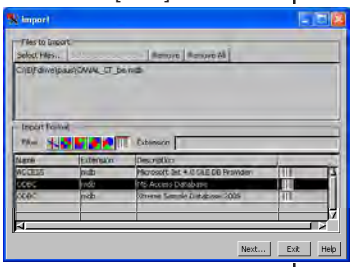
☑ type in the name and description for your data source, and click [OK] and [OK] again

Import into TNT Products Using ODBC

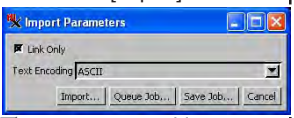
STEPS

☑ choose Main / Import from the TNT menu

☑ click on [Select Files], select the desired file from the ODBC Data Source you defined in the previous exercise (or some other data source of interest) and click on [Next]



☑ turn on the Link radio button in the Import Parameters window and click on [Import]

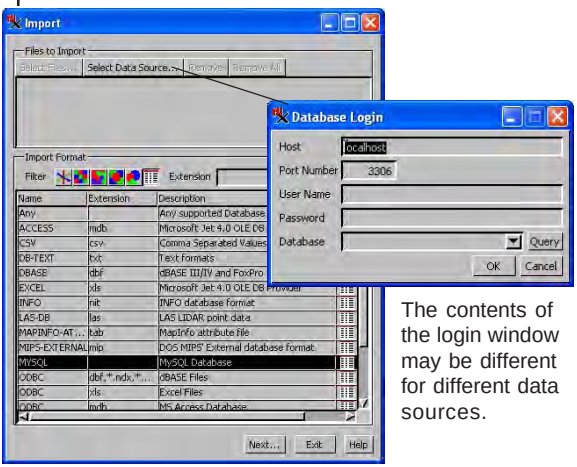


☑ create a new object or select an existing vector object then specify whether the tables will go in the point, line, or polygon database

After you complete the preceding exercise, you can copy or link to tables in the data source you specified. If your setup did not include selection of a specific file, you will need to select one. If your datasource is not file-based, you need to click on the Select Data Source button after you have specified the data source you want to import from in the format list. Clicking on this button opens a login window for the selected ODBC data source. The steps in this exercise are designed for file-based database systems.

You can copy or link to one or more tables from any ODBC data source. Linking lets you continue to manage the data in your external DBMS so that changes can be made in the external master file and viewed in the TNT products without having to import the data again or make duplicate changes in the database you copied. A linked table must have a primary key to be related to elements in a vector, CAD, or TIN object.

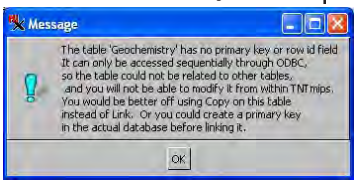
Some external database programs support field types not supported in TNTmips. The data in such fields will not be available in TNTmips.



The contents of the login window may be different for different data sources.

Link to ODBC Data Sources in Display

You can link or copy a table from an ODBC DataSource in the import process as described in the previous exercise or you can link to a table from an ODBC data source in the Display process or any other process with a View window and a layer manager.

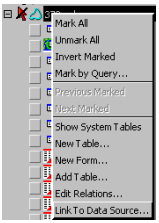


In order to be able to relate an ODBC table to vector elements or change table entries in the TNT products, the table must have a primary key or a row id field in the external format. The primary key is not shown as a primary key in the Properties window, which allows you to use the field to designate a foreign key and a Related Only attachment type for the linked table. You should use this indirect attachment method for linked tables.

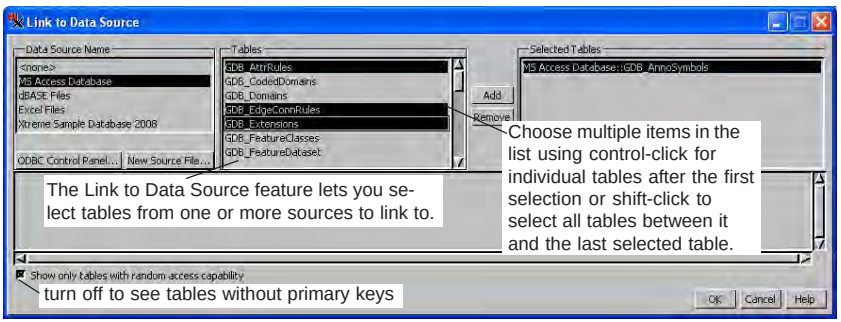
A linked table without a primary key can be useful in limited circumstances, such as providing values for virtual fields in internal tables. You could, for example, get the current price for a crop from a linked table and plug it into an expression that provides a dollar amount for the harvest of a particular field.

When working with linked tables, you need to adhere to the rules enforced by the external database software, which are generally more stringent than the rules in TNT.

- ☑ add a vector object that you want to link to an ODBC supported table to in a new group in Display
- ☑ expand the layer to show the databases
- ☑ right-click on the database in which you want to establish the link
- ☑ select the desired ODBC Data Source
- ☑ choose the tables you want to link to, click on Add, then click OK

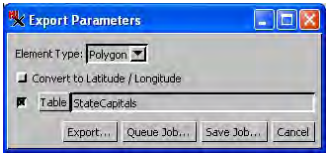


- ☑ right-click on the linked table name, and choose Properties
- ☑ select the key field in an internal table you want to relate to (same values as in linked table) and check that the table's attachment type is related only



Other Database System Requirements

TNTmips and TNTedit support export to the following spatial databases: Oracle Spatial, MySQL Spatial, and PostGIS (PostgreSQL Spatial). Unlike TNT vector objects, each spatial database can contain only one element type (points, lines, or polygons) and lacks the stringent topology required for accurate GIS work.



The export interface is the same for MySQL Spatial, Oracle Spatial, and PostGIS.

When you link to a table or a spatial layer in an external format, you automatically conform to any restrictions imposed by the native database system. When you are exporting from the TNT products to a database or spatial database, your data may have conditions that are not supported by the target software. Some of these may prevent you from exporting the data until you resolve the problem. Other issues may be resolved for you on export. For example, the structure of spatial databases dictates that there is one record per element. If the elements in the geometric object you are exporting have multiple records attached, only one record association is maintained by the Export process.

Some of the problems that will prevent export to Oracle Spatial are:

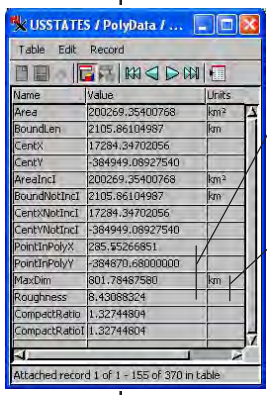
- multiple records with the same value for the primary key field,
- foreign key field values not found in the primary key field,
- mismatched field types in a primary/foreign key relationship, and
- a blank field with a cannot be left blank constraint set.

The first problem can usually be remedied by deleting duplicate records. The second problem can be eliminated using picklists. You cannot change field types after they have been added so the solution to the third problem involves using virtual fields. The last condition can occur if you enter field values in tabular view, which does not enforce this constraint. You just need to fill in values for each record. See the color plate entitled *Exporting Vector Objects to Oracle Spatial Layers*on MicroImages' web site for specific examples of error messages and how to fix them so the table and elements can be exported.

Other Tables Generated in TNTmips

You have looked at many ways to get database information associated with objects in TNTmips. Creation of tables and manual attachment is discussed in the Managing Geoattributes booklet. There are additional methods of getting database information associated with vector elements that have not been described here. TNTmips has six choices on the Geometric / Attributes cascade: Standard, Transfer Attributes, Fuzzy Properties, Raster Properties, Surface Properties, and Polygon Properties. Five of these create tables that contain the results of calculations for the vector elements or other objects that cover the same geographic area. The sixth, Transfer Attributes, was explored in an earlier exercise.

Standard Attributes

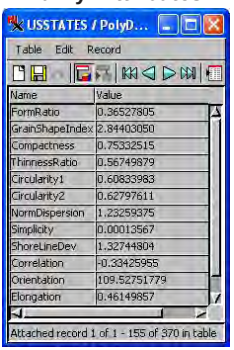


These fields are optional.

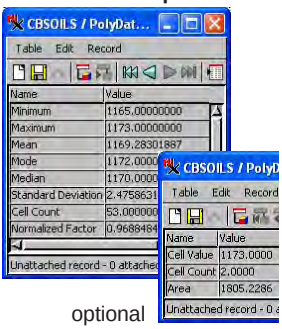
These fields are only generated in the stand-alone Standard Attributtes process but will be maintained in TNT's Editor if maintaining standard attributes has been selected for the object.

The Standard Attributes process computes this set of attributes for lines and polygons in a vector object in a single pass. Some of the attributes are optional and you can also set whether to update the table when the object is edited and whether to generate element ID tables. The Fuzzy Attributes process produces a table in the polygon database. You choose which of the 12 properties you want computed. The Raster Properties process generates statistics for the cells in the selected raster that fall within each vector polygon. You decide how to handle boundary cells, if you want a histgram record generated for each polygon, and whether to include islands. Surface Properties can be generated for all element types in a single pass. The contents of the table vary with the element type. The process uses an elevation raster to determine element properties, such as slope for lines and maximum slope for the lines in a polygon. The Polygon Properties process calculates statistics and properties from vector points or polygons and creates a table with this information for the polygons in the destination vector object. You select the attribute you want reported.

Fuzzy Attributes



Raster Properties



optional histogram table

Advanced Software for Geospatial Analysis

MicroImages, Inc. publishes a complete line of professional software for advanced geospatial data visualization, analysis, and publishing. Contact us or visit our web site for detailed product information.

TNTmips Pro TNTmips Pro is a professional system for fully integrated GIS, image analysis, CAD, TIN, desktop cartography, and geospatial database management.

TNTmips Basic TNTmips Basic is a low-cost version of TNTmips for small projects.

TNTmips Free TNTmips Free is a free version of TNTmips for students and professionals with small projects. You can download TNTmips Free from MicroImages' web site.

TNTedit TNTedit provides interactive tools to create, georeference, and edit vector, image, CAD, TIN, and relational database project materials in a wide variety of formats.

TNTview TNTview has the same powerful display features as TNTmips and is perfect for those who do not need the technical processing and preparation features of TNTmips.

TNTatlas TNTatlas lets you publish and distribute your spatial project materials on CD or DVD at low cost. TNTatlas CDs/DVDs can be used on any popular computing platform.

TNTserver TNTserver lets you publish TNTatlases on the Internet or on your intranet. Navigate through geodata atlases with your web browser and the TNTclient Java applet.

Index

Database Editor 5–17
- breaking links (unrelating) 8
- colors and text style 6
- copying tables 13–14
- delete duplicate records 17
- delete unattached records 17
- move table mode 7
- relating tables 9
- root table 8
- table joins 11–12
- table unions 10
- validate 16

database objects
- adding tables 18
- attachment type 16, 21
- combining tables 10–12
- composite fields 25
- computed fields 20, 24, 34
- copying tables 13–14
- creating new tables 21
- deleting duplicate records 17
- deleting unattached records 17
- editing relations 5–17
- editing tables 24
- field constraints 28
- foreign key 4, 27, 28
- implied one-to-one 20, 21
- importing 29
- indirect relationships 8, 9
- interleaved colors 19
- ODBC 30–33
- OLE DB 30
- picklists 27, 34
- primary key 4, 25, 28, 33
- related only 14, 21, 33
- Save As 22
- selecting elements by attribute 7, 19
- single record view 19
- spatial databases 34
- statistics as new tables 23
- statistics 22
- string expression fields 19
- substatistics 22
- tabular view 19, 22, 24
- terminology 4, 8
- units 21, 24

MicroImages, Inc.

11th Floor – Sharp Tower
206 South 13th Street
Lincoln, Nebraska 68508-2010 USA

Voice: (402)477-9554
FAX: (402)477-9559

email: info@microimages.com
Internet: www.microimages.com