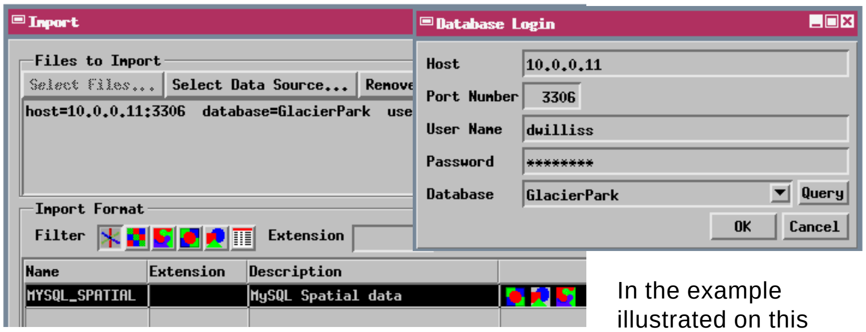
Linking to MySQL Spatial Layers

The MySQL relational database system includes spatial extensions that allow it to store geographic features that can be served to applications such as the TNT products. Following the specifications of the Open Geospatial Consortium, each MySQL spatial object (layer) is stored in a separate spatial table in the database, with one record in this table for each distinct spatial feature (element) in the layer. Ancillary tables store coordinate reference system information and other internal data. In the spatial tables, a geometry column stores the geometric information for each record, while conventional data columns hold associated attribute information for that feature. Point, line, polygon, multipoint, multiline, and multipolygon geometries are supported. The multi-element geometries allow the geometry field in a single record to store multiple elements of the same type, all of which have the same associated attribute set. Like a shapefile, however, each spatial table can store only one of these types of spatial element; a single MySQL geodatabase layer cannot include both attributed lines and polygons, for example.



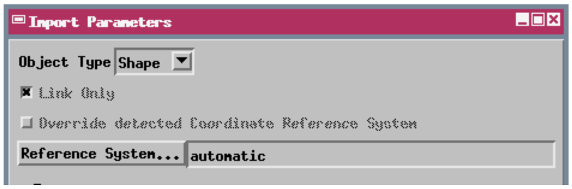
host=10.0.0.11:3306 database=GlacierPark use

Name	Extension	Description
MYSQL_SPATIAL		MySQL Spatial data

Host: 10.0.0.11
Port Number: 3306
User Name: dwilliss
Password: ********
Database: GlacierPark

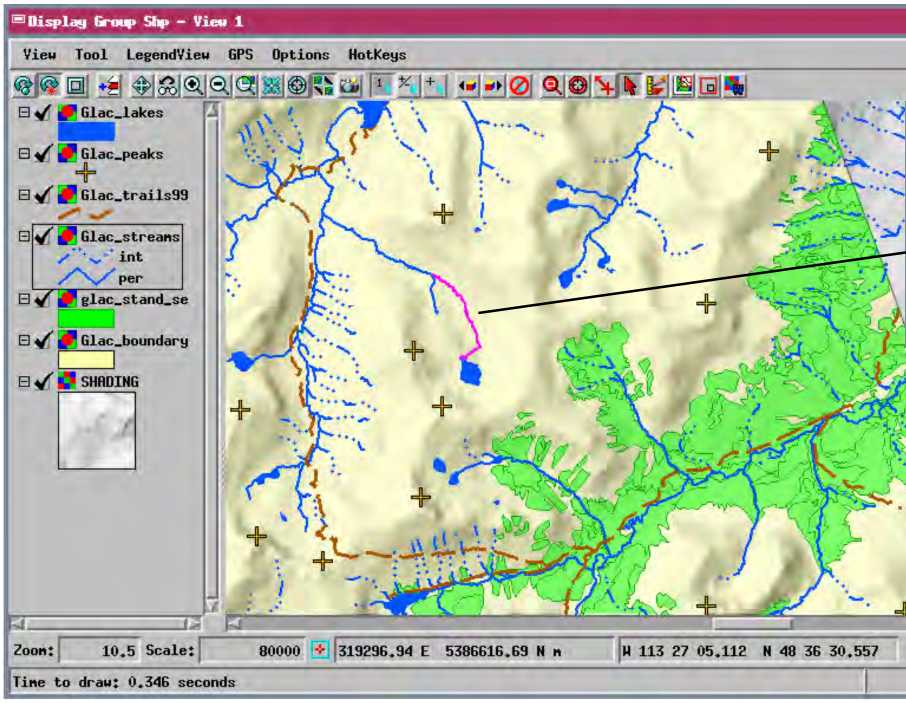
In the example illustrated on this plate, links are set up on a PC running Windows XP to geospatial tables in a MySQL database residing on another PC running Linux.

You can set up links to MySQL spatial layers in the TNT Import process. The import/linking procedure first prompts you to log in to the MySQL server and specify the desired database. You then choose the destination object type for the spatial data in TNT. To link to the MySQL layers, choose Shape as the object type. This choice creates in the destination Project File a shape object that represents and links to the elements and associated attributes that remain stored in the corresponding MySQL spatial table. (Alternatively, you can choose to fully import the MySQL layers to TNT objects by choosing either Vector or CAD as the destination object type.) Finally, you specify the destination Project File and select any or all of the MySQL spatial tables for linking/import.

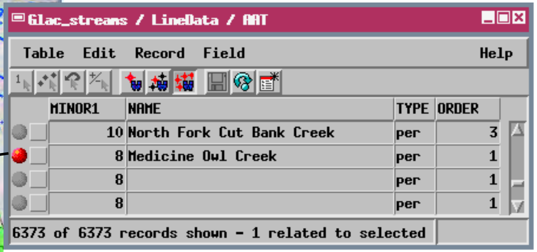


Object Type: Shape
☒ Link Only
☐ Override detected Coordinate Reference System
Reference System...: automatic

Linked MySQL geospatial layers can be styled using associated attributes in the MySQL spatial table and used for spatial display or analysis procedures. Since the TNT products and MySQL can run under all major operating systems, including Windows, Mac OS X, and Linux, combining these products provides flexible cross-platform GIS capability in an enterprise GIS environment. For example, standard GIS data could be stored in a MySQL database running on a central Linux server, but this data could be analyzed and used to generate specialized products in TNTmips on client computers running Windows or Mac OS X.



Zoom: 10.5 Scale: 80000 319296.94 E 5386616.69 N m W 113 27 05.112 N 48 36 30.557
Time to draw: 0.346 seconds



MINOR1	NAME	TYPE	ORDER
10	North Fork Cut Bank Creek	per	3
8	Medicine Owl Creek	per	1
8		per	1
8		per	1

6373 of 6373 records shown - 1 related to selected

Each of the shape layers in this View is linked to a geospatial table in a MySQL database. Display styling (including partial transparency for polygon fills and styling by MySQL attribute for streams) was set up and saved for each layer in the TNTmips Display process after linking. In this illustration the linked MySQL shape layers are overlaid on a relief-shading raster object created in TNTmips from a digital elevation model.