



2 March 2007

Provisional Summary of New Features for TNT 2007:73

The official release of version 2007:73 of TNTmips, TNTedit, TNTview, and TNTatlas/X will be in mid-March.

This memo summarizes the features available now for your testing in the development version 2007:73 of the TNT products. Note: features in italics have not yet been implemented and may or may not be in the official or any release of 2007:73.

You do not need to have purchased version 2007:73 of your TNT product to try it now. Any client using version 2006:72 of the TNT products can immediately begin to experiment with and use this development version of 2007:73 by downloading it from www.microimages.com/downloads/tntpatch/DV73.htm. Installing these new products now will not affect or change the current installed version of your TNT 2006:72 product(s). It will be installed on your computer as a totally separate product. This will permit you to test this newest version's features, submit suggestions for final adjustments, report errors, and even begin to employ these advances in your activities while continuing to use your current version for your production activities.

If you have purchased version 2007:73 via annual maintenance, you can simply continue using and updating the official release of version 2007:73. If you have not purchased version 2007:73, you can continue to use it for 30 days beyond its official release (i.e., 30 days after it ceases to be the development version).

Even if you are using a version of a TNT product released before 2006:72, you can also try the 2007:73 development version of the TNT analysis products now using TNTlite. Simply download the development version from www.microimages.com/downloads/tntpatch/DV73.htm, and run the product in TNTlite mode without any software authorization key.

Technical Guides (formerly called color plates) describing the new features in version 2007:73 are being introduced in the daily NEWS on the home page of microimages.com. All these new TechGuides for this version will, as usual, be posted at www.microimages.com/documentation/platecatalog.htm.

System

X Server for Windows Desktop Mode.

A translucent curtain added between the TNT windows and all other windows (focuses your attention on the TNT windows currently in use).

Set color and transparency level of this curtain or shut it off.

- Clicking the mouse anywhere else outside a TNT windows hides curtain.
- Opening or clicking a TNT window restores curtain on Windows desktop.
- “Keep on Top” option available on title bar menu for each window.
- Auxiliary window(s) by option can stay above their parent window.
- “Snap To” option to adjoin a new window to nearest window or screen edge.
- X server for Windows updated to be fully compliant with X11R7.1.

System.

- Improved management of free space (i.e., released space) in Project Files.
- Set TNT processes to reference local time.
- Screen snapshot automatically saved just after any TNT error message.
- New “Aligned,” “No Cache,” or “Default” (unaligned) buffering options and “Sequential” or “Scattered” access options in Operation Timings process to evaluate your system’s performance (e.g., drive fragmentation?).
- JPEG2000 compression and decompression is multi-threaded everywhere.
- Thread pool (multithreading) introduced into Graphic Rendering Engine (supports asynchronous use of Web Map Service objects in TNT views).
- Option to set maximum number of threads used in TNT processes.
 - As TNT processes are released with multi-core support, you can use this option to temporarily force them to use only 1 thread. You can then compare your multi-core system’s performance in your TNT product against a simulated single-core system. Examination of these results may indicate that some other system component is limiting your performance, such as a slow drive, slow bus, slow network access to geo-data, using same drive for read and write operations, older operating system’s thread queuing, ...

DataTips.

- Set up a complex DataTip combining graphs, images, attributes, and text.
- Layout components in this integrated pop-in window as previously.
- Alternatively design the DataTip layout and contents using HTML.
 - Using HTML to layout a complex DataTip means that the same appearance can be duplicated in the Info window (the DataTip equivalent) in TNTmap. This Info window pops in when the feature is selected in TNTmap Open or Viewer to show the same DataTip contents or for a WMS server layer added to TNTmips or other TNT products.

Direct Use of Geodata Files.

- New geodata raster file types directly selected and used as objects:

ERDAS (*.img, *.lan)	NetCDF (*.nc, *.cdf)
Arc Binary Grid (*.adf)	NetCDF GMT (*.gmt, *.grd)
Windows bitmap (*.bmp)	VTBuilder Binary Terrain (*.bt)
GIF (*.gif)	BSB Nautical Chart (*.kap)
USGS ASCII DEM (*.dem)	Netpbm (*.ppm, *.pgm)
DTED (*.dto, *.dt1)	Swedish grid RIK (*.rik)
DOQQ (*.doq)	Envisat (*.n1)

USGS SDTS (*.ddf)	Raster Matrix (*.rsw, *.mtw)
ILWIS (*.mpr, *.mpl)	PCRaster (*.map)
X11 Pixmap (*.xpm)	GRASS (*.hdr)
others (via GDAL *.hdr and *.mem)	

Project File for Rasters.

Multifile raster linking no longer creates a subobject for each linked file (i.e., much smaller link file, links faster, and loads faster).

Kakadu library for multithreaded support of JPEG2000 updated to V5.2.

Multi-threading used in JPEG2000 compress/decompress operations (i.e., simultaneous use of all your processing cores: 2, 4, ..., more than 3 times faster for a 4 core system).

GeoExpress SDK for MrSID updated to LizardTech V6 (does not support multi-threading).

Use Windows OS calls to synchronize access to Project Files (eliminates most creation/use of lock files if local drive is being used).

A larger buffer is used during the allocation of space for a raster object (speeds up creation especially for very large raster objects).

Long names and descriptions supported for styles and patterns.

Pyramiding a raster object will automatically delete existing pyramids.

Smaller Raster Objects.*

Tiles in object that contain only null cells are not created (includes null tiles that represent holes in real data).

Extents of irregularly-shaped raster data no longer set the size of the raster.

Large reduction in object size if real data area is unusual in shape (e.g., lossless raster object of Florida or South Island of New Zealand with water as null are reduced to ~1/5 of previous size).

Greatest effect when raster object is uncompressed (lossy or lossless JPEG2000 compression in raster yields similar results).

*This is a significant improvement since more and more of your projects do not involve single, rectangular images as in the past but larger, irregularly-shaped project areas that result from mosaics, restricting analyses to areas of interest by extraction, flight lines not north/south, and so on.

Multi-Threaded JPEG2000 Compression/Decompression.

2, 4, 8, or more cores now supported for faster JPEG2000 applications.

Achieve 60 to 80% use of each core (drive and data bus dependent).

Especially useful when JPEG2000 is applied to tasks using large rasters (e.g., import/export, extract, mosaic, creating multifile raster object ... specific example is exporting directly from linked *.sid to *.jp2).

Typical example: Mac OS X with 4 cores and SATA II drives can be 3 times faster in the JPEG2000 compression action in extract or mosaic.

Manual Assembly of Multifile Raster.

Rasters used as tileset no longer need to match in row, column, or size.
(limitations: null edges are not detected and CRS must be uniform)

*Use Project Files (1 raster each) as tileset
(secures rasters in a locked TNTAtlas against any other use).*

Assemble a multifile raster from 1000s of georeferenced rasters in a minute.

Create a multifile raster from tiles of one format but of varying size
(e.g., select directory of 100s or 1000s of DOQQs of dissimilar sizes)

Use any supported georeferenced raster format with matching CRS as tileset.
(e.g., use a collection of DOQQs in *.sid format)

A few large rasters such as 100 *.sid or *.jp2 files of counties or other provinces can be used as a multifile raster as long as they have the same CRS. However, if the collection of files is to be used over and over, maximum viewing performance will be achieved if these geodata files are mosaicked. This mosaic can then be automatically broken into a multifile raster with many small tiles using the new process for the automatic assembly of a multifile raster.

This process is not a virtual mosaic and is designed for maximum display performance, thus null edges are not detected or blended. The process uses the last value it reads for a common cell in the overlap area and this value can be null. If your rasters contain null areas, different CRSs, or require other adjustments then they should be mosaicked first and then auto-converted to a multifile raster.

Automatic Assembly of Multifile Rasters.

This new Create Tileset process automatically builds a multifile raster object and other types of tilesets
(raster object has same limitations as for Manual Assembly noted above).

Select huge single raster or multifile raster.
(currently supports input and output of *.rvc, *.jp2, *.png, *.tif, and *.jpg).

Outputs tileset in a directory structure containing uniform tiles of the size and format selected.

Outputs a tileset meeting OSGeo proposed Tile Map Specification (1.0.0) (this tileset parallels the tileset structure used by Google Maps and Earth).

Automatically links-up tiles in directories into a single multifile raster object.

This Create Tileset process is not a substitute for mosaicking your miscellaneous images of different scales, formats, projections, colors, ... It simply merges the input objects when writing each tile with no options to control how they are merged. Your collection of raster objects or geodata files may be varied as noted. Use mosaic if you want more control over handling mixed projections, overlap methods, and other parameters. Then use its resulting raster as input to the CreateTileset process to create your multifile raster object or tileset.

Project File for Shapes.

Use MySQL spatial tables directly as shape objects.

Use PostGIS spatial tables as shape objects.

Export geometric objects to PostGIS tables.

Project File Maintenance.

Georeference model of object now shown in object information list.
Compression method and ratio is shown for linked raster file types.

Coordinate Reference Systems (CRSs) added.

Florida Albers supported for NAD83 and NAD83(HARN) datums.
Cassini projection updated for Malaysian applications.
Putrajaya added to Cassini / Malaysia list.
Two more Local Stereo 70s supported for Romanian cities.

GPS Support.

Use Garmin's proprietary support of USB ports to read coordinates.
Use log files in the GPX (GPS Exchange Format) including multiple tracks
(e.g., use them for geotagging photos).
Import GPS data from a formatted text as a GPS log.

TNTsim3D for Windows.

Automatically pause process when minimized
(releases processing resources to other applications).
Use Windows codec to record new movie formats (*.wmv, *.mov, and mpeg).
Record movies of other view windows besides Main View.
Combine multiple view windows into a movie
(e.g., Main view, Map view, ... synchronized in a single movie).
Specify that movie frames of view are rendered at desired resolution
(i.e., 640 by 480, 1024 by 768, 1080i, ...).
Set the pixel resolution of movie made from view (e.g., 640 by 480 pixels)
Reorder layers in the view without providing offsets.
Allow layers to have transparency.
Select CAD and Vector objects and drape styled features onto views.
Show label text as billboards.
Geospatial scripts can use many new formats (see import/export below).

Contact Support Form.

Movie recorder automatically installed as part of your TNT product.
Activate recorder from the TNT Menubar.
Activate recorder from Contact Support dialog.
Review movie from dialog and include it as part of your report.
(Sends screen snapshot saved after any TNT error message.)
Include screen snapshot when use of form is from Help menu.
Capture screen shot from within Contact Support dialog.
Save report/attachments to send later when Internet access is available.
Attachments for report put into MicrolImages subdirectory in user folder.

Show size of report including all attachments before saving or sending.
Open a saved report for review before sending or for later review.
Automatically checks email account being used and reports any problem.
Patch date of TNT product is included in report (see Help/About TNTmips).
Add information to the subject line of report.
Automatically include tntproc.ini and tnthost.ini files and path to shellex.dll.
Key number reported for use of form in TNTlite is "TNTlite."
Record environment variables to show where temporary files are going.
Include directory structure and names of all files in TNT programs folder.

Licenses.

FLEXnet replaces FLEXlm to manage floating licenses
(supports installing license manager on Intel Mac and XP64).
Microsoft's Windows Vista is fully supported in this version.
Apple's Mac OS X 10.5 supported when it is released.

Visualization

GeoDisplay Properties.

Complex DataTip can now integrate text, graphics, and images (e.g., photos).
Separate scale ranges for auto generated labels of points, lines, and polygons.
Show object name in selection dialog when prompting for missing file/object.
Extents determined rapidly for all objects after layers selected
(significantly faster if many objects [100s to 1000s] selected).
Closing active Display window automatically makes earlier Display active.
Show message in Display Manager window after saving Display.
Display Manager window now defaults to open at previous size.
Sketch tables are automatically named by default.
Expand/contract layer list for group or layout in Display Manager window.
Uses object name as group name in Add/Saved Groups.

Image Fusion.

Mix images of varying non-integer cell sizes using automatic reprojection.
(reprojects low-resolution images to match high resolution images)
16-bit data range for output to each 16-bit separate RGB image.

Integrated Viewing of Internet Images.*

View images from any Web Map Service (WMS 1.1, 1.3, and ISO 19128).
WMSs can be located on an Internet site, WAN, LAN, or VPN.
Use new layer selection window to browse Internet/Intranet for WMS layers.
Add layer(s) from any WMS to your composite view in your TNT product.
Combines layers from WMSs with local geodata files and TNT objects.

Manages delays and errors that can result from using images from other sites.

Make measurements or build sketch object using WMS layers.

View WMS layers in the Editor.

Even FREE TNTAtlas supports access to and use of WMS layers.

**Understanding what you can and cannot do with WMS layers!*

A Web Map Service (WMS), including that provided by a TNTserver, will only deliver an image to your TNT product! This image is compressed, usually as a PNG or lossy JPEG file, and deliberately set by the WMS to not exceed a maximum pixel size (e.g., 2048 by 2048 pixels). All raster, vector, CAD, and other geodata formats offered by a WMS are automatically converted by that service into images before being sent to your TNT product for compositing into your view.

It is important to understand that by these limitations a site publishing geodata via a WMS is offering it for you to view, not analyze. Version 2007:73 of the TNT products lets you locate and select WMS layers. These WMS images are then combined by your TNT product into a composite view with your local, directly-supported geodata files and TNT objects. You can apply many of the tools provided by the TNT display process to these WMS layers and the composite views in which they are incorporated. For example, you can use the sketch tool, appropriate edit tools, measurement tools, and so on. However, the images in your view pulled from a WMS are not directly available for other automatic analysis.

When you include any WMS layer in TNT display, you are accessing material that may or may not be available from the remote site. You are also requesting an image that may be returned quickly or be delayed depending upon the current load on that remote WMS, on your web access bandwidth, and so. Creating a view that uses only local geodata files and TNT objects is totally under the control of the TNT process and the local operating system. The layer buffers are all filled and only then does the composite view show. If a layer is not available, this is immediately known and you are prompted to locate it. But TNT display doesn't know when a WMS layer will be available in its local buffer. As a result, when all the local layers are buffered your view will be shown and will later be redrawn when the WMS layer is available. This uses a new multithreaded capability incorporated into the TNT Graphics Rendering Engine (GRE). Using this capability your viewing activity will not "hang up" waiting for a WMS layer that may never arrive and you can also use all the tools and actions as if the final view is complete.

TNTAtlas Goes Internet.

*Now an advanced FREE viewer for geodata published via WMS.**

Use the tools in a TNTAtlas with WMS layers, sketch, measure, ...

Use with GPS and mobile wireless devices that support Mac OS X or Windows.

Synchronize TNT Views with Google Earth (planned: not yet implemented).

Launch Google Earth with vertical view and zoom matching TNT view.

Geolock Google Earth's vertical view to a TNT view.

Move Google Earth's oblique view to maintain same nadir point as TNT view.

Launch Google Earth using selected layers as separate Temporary Places.

Render to KML now keeps raster, points, lines, polygons, ... separate.

Integrated Geotagged Digital Photos (photos with lat/lon coordinates).

New simple Photo View window available with pan and zoom.

Directly use JPEG image with the geotag recorded in EXIF tag.

Examine photo content and use to plot it on map/image to geotag its position
(as accurate as GPS if satellite or orthophoto print is used in field to record a feature at the camera position and then used later to plot it on view).

Add geotags to photos from time correlated GPS logs
(an SML script with similar capabilities is also provided for this for your modification/extension).

Manually assign positions of photos (i.e., geotag them) by plotting in view.

Manually refine positions of geotagged photos by dragging them in view.

Insert geotag information into photos' EXIF headers.

Photos can be stored as JPEG images (i.e., Binary Large Object [BLOB] fields) as image fields in a TNT internal, MySQL, or PostGIS database.

Create description field (i.e., name) for each photo.

Thumbnails of photos are automatically made in a second blob in same record.

Attributes of photos also stored in database.

View photo in blob directly in Single Record View.

Image field in tabular view shows iconized (tiny) representation of its image.

View DataTip of linked or blob photos when cursor over image field in record.

Open new Photo View window of image by clicking on its image field or name.

Open list of photo (EXIF tag contents) from Photo View window.

DataTips for features in view can contain a photo
(e.g., nearest photo, photo associated with feature, ...).

Pinmap Geotagged Photos.

Pinmap layer can now symbolically represent digital photo positions.

Any point symbol can be used to represent photos in a pin map layer.

Tiny thumbnail representations of photo can also be used as pin map symbol.

Cursor near pin automatically shows DataTip (i.e., image) of photo.

Click on pin to show image in new Photo View window.

DataTip for pin can be a small photo or more complex by adding graph and text.

Open a view of all acquisition parameters stored in photo's EXIF record.

Open Tabular View of records with imbedded photo stored as blobs.

Profile Display Window.

Display coordinates in CRS of group instead of in object coordinates.

Set-up vertical scale start/interval always divisible by 1, 2, 5, 10, 20, 50, ...

Unused space in window above/below profile is automatically deleted.

Save map coordinates of horizontal axes if raster is georeferenced.

Editing Spatial Objects.

Quick Snapshot is now available via an icon on the toolbar.

Edit an object and immediately launch view into Google Earth.

Google Earth for Feature Capture.

Render contents of a TNT view into Google Earth.

Interpret these TNT overlays in Google Earth view using SketchUp.

Save interpretations in SketchUp as KML layer.

Import KML layer into CAD object.

Use CAD object in TNT view, spatial data Editor, convert to vector, etc.

Vector/CAD/Shape Operations

Import Geometric Objects.

Oracle Spatial as a CAD object.

KML as CAD object.

DGN as CAD with Mac and Linux.

Labels from shapefile's text elements now have angle and size fields.

LIDAR datasets imported from *.las as 3D points with database records.

Updated to use current options/interface for

MOSS, SDF, SDTS, CARIS, and NTF-Vect, and MIF including attributes.

Export Geometric Objects.

Multiple shapefiles, DWG, or DXF in one setup.

DGN format using OpenDesign libraries.

DWG and DXF using OpenAlliance libraries.

Label text strings exported to shapefiles with angle and size fields.

MOSS and MIF updated to use current options/interface.

Export to KML.

CAD, vector, and shape objects exported as graphic features.

Raster objects exported as linked raster files.

Styles of features are exported (except bitmap patterns).

DataTips exported as information for each feature.

Tables exported as Comma Separated Values

(concept of database tables not incorporated into KML).

Render Objects in View to KML.

Render points, lines, polygons and associated styles into KML graphical features
(note, bitmap patterns can not be rendered as not defined in KML).

Render rasters into linked rasters.

Database tables converted into Comma Separated Values

(note, tables are not defined in KML).

Labels and styles converted to KML label features.

DataTips converted to information features displayed by selection of feature.

Merging Vector Objects.

Warning shows if input objects mix global and local engineering georeferences.

Vector Grid Generation.

Position a grid and define its intervals in latitude and longitude.

Sample Point Generation.

It was recently concluded that this activity should be moved from the Vector Grid Generation process into a new process to allow for more flexibility in its further development. This new process's features will be itemized as part of the official release of 2007:73.

Database Operations

Make note of this important new technical book on this topic.

Spatial Database Systems: Design, Implementation and Project Management. 2007. by Albert K.W. Yeung and G. Brent Hall. Springer, PO Box 17, 3300 AA Dordrecht, The Netherlands. (ISBN-10 1-4020-5393-2 (PB)). 553 pages.

Part 1: Introduction

Chapter 1: The Current Status of Spatial Information Technology

Part 2: Database Principles and Architecture

Chapter 2: Concepts and Architecture of Database Systems

Chapter 3: Database Models and Data Modelling

Chapter 4: Spatial Data and Spatial Database Systems

Chapter 5: Spatial Data Standards and Metadata

Chapter 6: Spatial Data Sharing, Data Warehousing and Database Federation

Part 3: Spatial Database Implementation and Project Management

Chapter 7: User Education and Legal Issues of Spatial Database Systems

Chapter 8: Use Needs Assessment and Multi-user Spatial Solutions

Chapter 9: Project Management for Spatial Database Implementation

Chapter 10: Web-enabled Spatial Database Systems

Chapter 11: Spatial Data Mining and Decision Support Systems

Part 4: The Future

Chapter 12: Trends of Spatial Database Systems

Glossary of Terms

Index

Database Editor / Operations.

Auto detect changes to dBASE, Access, or MySQL database tables.

Any redraw will detect and use changes in these tables.

An automatic redraw occurs when TNT internal tables are changed.

MySQL support for all platforms (Mac, Windows, Linux, and Sun).

Write to PostgreSQL tables via OLE DB.

Create new table(s) in PostgreSQL.

Table union and other combinations support long field names.

Image BLOB fields can be viewed in a single record view.

DataTip of image pops in over image BLOB fields in a tabular view.
 Field description pops in as ToolTip over column heading.
 View technical properties of any database table, not just user-defined tables.

Geotagging Database Records (geospatial script in SML).

Adds GPS coordinates to existing records.
 Interpolates GPS logs for position based on date/time fields in record.
 Adds interpolated latitude and longitude fields to the record.
 Opens view of the modified records as a pinmap.

Raster Operations

Import Raster Objects.

AVIRIS import module updated to use current options/interface.
 Google Maps JPEG and PNG files imported as georeferenced
 (uses grid position tile naming convention to establish georeference).
 Surface properties computed automatically for common DEM formats.
 Import IPTC tag with JPEG files and store in metadata and use geotags
 (IPTC = International Press Telecommunications Council).
 Import EXIF tag with JPEG files and store as metadata or in record
 (EXIF = EXchange Image Format).
 Support upgraded for GIRAS format from USGS for LULC data.
 Create multifile raster that has incomplete or missing tiles.
 Import ECW files under Mac OS X and updated support for all OSs.
 No layer table created when DXF file imported into separate objects.
 Use ESRI's *.aux file for CRS, projection, and transformation.

Export Raster Objects.

Rasters can be exported with contrast enhancements
 (i.e., data values converted if format has no way to record contrast).
 Default for TIFF export is 300 dpi resolution to match common printers.
 Export ECW file under Mac OS X and support updated for all OSs.
*Create ESRI's *.aux file with CRS, projection, and transformation.*

Render Objects in View to Rasters.

Save georeference in KML file whenever possible.
 Option available to render to JP2 and GeoJP2 formats.
 PNG files get same global transparency as object they are derived from.
 Render to Google Earth option is available.

Geotagging Digital Photos (geospatial script - SML).

Adds GPS coordinates into JPEG photo's EXIF tag.
 Interpolates GPS logs for photo positions using date/time in EXIF tag.
 Assigns nearest GPS position if discrete GPS positions recorded.

Directly enter coordinates of photo

(e.g., read from TNT view, Google Maps used via TNTmap, ...).

Click to assign coordinates of cursor position in view to a photo's EXIF tag.

Creates vector object with point element for each photo.

Builds table with a record for each point containing all data in EXIF tag.

Opens a pinmap view of the vector object, add any other layers.

Linear Combinations.

Output raster(s) can now be compressed in any supported format.

Use min/max cell values to auto-scale output raster object(s).

Automatically set number of output rasters to match number selected.

Allow the number of output rasters to exceed the number of input rasters.

Add "Clear" or "New" options to reset an input array back to the default state.

Compress output rasters using any method supported.

Predefined Combinations.

Output rasters can be auto named from input objects.

Adding algebraically can optionally treat any null input cell value as a zero.

Morphological Operations.

Dilation/erosion can use weighted values for cells in grayscale images.

Pyramids can now be sampled or averaged.

Compress output rasters using any method supported.

Statistics by Category.

Use mask or region to define area to compute properties.

Any integer raster can be used as a category raster.

Contrast Matching.

Contrast table(s) are transferred from reference raster to output raster.

Terrain Operations

Topographic Properties.

Shading is cached and reused automatically as long as window is open.

No DataTip is shown for shaded layer.

Coordinates for sun angle calculation default to center of DEM.

Output raster(s) can now be compressed in any supported format.

Contours to DEM (geospatial script - SML).

The MorphContour script implements a method of interpolating an elevation raster from contour lines using morphological functions, as described by William Barrett, Eric Mortensen, and David Taylor in the 1994 paper entitled An Image Space Algorithm for Morphological Contour Interpolation (<http://citeseer.ist.psu.edu/178773.html>).

The script transfers elevation values from contour lines in a vector object to corresponding cells in the output raster object, leaving the inter-contour areas initially blank. It then applies raster morphological functions including dilation and erosion to the inter-contour areas to find the midline between each pair of contours and assign those midline cells the average value of the two bounding contour elevations. The original contours and the new set of midlines (intermediate contours) then define a new set of inter-contour areas that are processed as before to find an additional set of finer midlines. This procedure is repeated until all possible midlines have been found.

Viewshed Analysis.

New, improved user interface.

Standard view Options menu available for Viewshed Analysis window.

Select colors for viewpoints.

See and alter all viewpoints and parameters in grid layout.

Create or add to a viewpoint list from points selected from a vector object.

Manually add or edit positions in a viewpoint list.

Save current viewpoints in list as points in a vector object.

Watershed Geomorphology Analysis.

Much faster performance but still a very computationally intense process.

Process larger DEM rasters

(tested on 100,000 by 50,000 cell DEM, but will take many hours if all output options are selected).

Much faster algorithm to fill all depressions.

Filling all depressions no longer memory based for use on very large DEMs.

Step for deleting pits is removed and is now part of depression filling.

Input integer or floating point DEMs.

Faster specialized integer algorithms used for integer DEMs.

Faster tracing of flow directions.

Initialization of outputs made after process is started, not during setup (i.e., much faster setup of the input parameters without delays).

Output rasters can now be compressed in any supported format.

Flow accumulation raster is 32-bit and is monitored for overflow.

Insert location of dams into process.

STREAM_ORDER table renamed as Stream table and adds MinZ, MaxZ, Elevation Drop, Length, and average slope.

Messages reported after each major step to monitor progress.

DataTip prefixes for each temporary object identify values reported.

Support mouse wheel for scrolling view of attributes.

Specific Catchment Area (SCA) raster is now an output option.

SCA is the upslope contributing area per unit width of flow perpendicular to the flow direction. The Watershed process computes it from the flow accumulation, cell area, cell width; it is used in modeling of overland flow, erosion, and soil loss.

Additional attributes computed for flow path lines:

Stream Properties table with length, elevation drop, average slope, and sinuosity ratio.

Stream Distances table with minimum, maximum, and average upstream and downstream flow distances.

Attribute tables now spatially relate flow path lines and basin polygons.

These tables can be used to identify the flow path lines that traverse each basin polygon and all the polygons that contain a specific flow path or set of paths.

Attribute table added for extrema and flow direction rasters.

This table contains the text equivalent of the raster value code schemes used (e.g., extrema type field = local maximum, flat, etc. and flow direction field = upper left, upper right, etc.). These text attribute fields are automatically set to show as the default DataTip for these rasters.

Compound Topographic Index (CTI) raster is now an output option:

CTI = natural log of (upstream contributing area / tan (slope)). This property is computed for each cell in a depression filled DEM. It is used in surface water hydrologic models and also is used to predict local depth to the groundwater table. Slopes of 0 are detected and recorded as maximum value in this CTI raster.

Maximum Drains To and Maximum Drains From rasters are now output options.

Maximum Drains To raster contains the maximum upstream flowpath length from the watershed boundary to each cell.

Maximum Drains From raster contains the maximum downstream flowpath length from each cell to downstream watershed outlet or boundary.

Create output rasters with bit-depth only as large as necessary.

The output raster objects created with integer results (such as watershed raster, flow accumulation) use a 32-bit range to accommodate large datasets. However, when processing a smaller watershed, these results may have a smaller range (e.g., only 200 watersheds) and bit-depth of final output rasters is reduced accordingly (to 8-bit in this example) to speed up processing and reduce raster sizes.

Surface Modeling.

This complex process has been completely recoded.

User interface redesigned but this is still a complex, multifaceted process.

Eliminate redundant ways to select input objects.

Set search radius units in output cells or ground measurement units.

Designate raster units for output (feet, meters, lat/lon, ...)

Align output grid cells to integer multiple of cell size.

Compress output raster using all supported compression methods.

Geospatial Modeling Language

SML Operations.

Encrypted scripts can be created to run in TNTlite.

Status window displayed when a script is run from the menubar.

Function names and descriptions now in columns for easier reading.

Variable type "string" and class STRING now treated as same thing.

SML Editor.

New viewer for documentation

(requires specific authorization of TNT key and license).

Search documentation for classes with many methods and/or members.

Class list of members and methods now in alphabetical order.

Create icon bar for query editor and add icons for "Insert" accessories.

Alphabetical re-sorting of function groups on interface language change.

SML Functions and Classes.

EXIF() class will create an EXIF tag.

Get compression ratio of a raster via new methods in RVC_RASTER class.

Morphological contour method for surfaces via new class.

ImportRaster() can import multifile raster objects.

RVC_RASTER has methods to test raster for compression.

SetVisibleView() method for GRE_GROUP class.

Timer class added for use in MGUI.

MieHANDLE-derived classes have new import and export methods.

VisibilityInView Get/Set methods added to GRE_GROUP.

TIFF export can be forced to write CRS as "user-defined." CONTEXT::CurrentLineNum member added.

DBTABLEINFO has a member to set the encoding for the table.

Compute the determinant of an NxN matrix.

Compare two vector objects.

GetOutputFileName() creates a new RVC file if destination does not exist.

Select a variable number of input TNT objects (also retrofit into 2006:72):

DlgGetObject permits selection of a single object,

DlgGetObjects selects many objects (number unspecified),

DlgGetObjectSet selects a specified number of objects.

New << operator for class STRING.

Use Windows codec to generate movies (*.wmv, *mov, and mpeg).

Use Mac's Quicktime codec to generate movies (*.mov and mpeg)

New database classes for Raster, vector, CAD, shape, and TIN objects.

TNTserver

Some of the new features introduced into TNTserver are based upon the completion of features added to TNTmips. As a result the official release of the TNTserver 2007:73 will be later than the release of the TNT analysis products. For example, SML has been modified to be used by TNTserver to process WMS layers and the use of WMS objects just added in TNTmips etc. may require changes to TNTserver 2007:73.

Web Feature Service (OGC V1.1 = ISO 19142).

This release supports all mandatory WFS operations:

Get Capabilities.

Obtain metadata of the available list of feature types, layers, and database tables and operations supported.

DescribeFeatureType.

Return XMLSchema describing the specific feature type requested.

GetFeature.

Retrieve the features requested in the format specified by the “DescribeFeature-Type” operation.

This release may not include support for the optional WFS operations:

Transaction.

Operation to modify features using primitives of create, update, and delete (i.e., edit the layers and their database records).

GetGmlObject, LockFeature, or GetFeatureWithLock.

TNTserver Manager has been modified to support the use of a WFS.

Web Map Service.**Publishing WMS Layers with Styles.**

Virtual WMS layers can now be published by a TNTserver that are “styled” (i.e., computed) when requested via a GetMap request from the real objects and files. Since TNTserver is built on an embedded TNTmips, these styled layers are processed (i.e., styled in web service terminology) using the functionality in TNTmips. For example,

- *a single DEM layer can be published with multiple styles each style applying a different color map;*
- *the same DEM could also be published to appear as a layer styled into shaded relief;*
- *an image could be published with multiple styles each style applying a different contrast enhancement;*
- *a user could input data into style entry to use to build the URL for the GetMap request, and so on.*

The WMS specification uses the concept of dimensions to convey needed and/or optional parameters to and from a WMS. The WMS specification permits the addition of custom dimensions to be conveyed to and from the WMS. The client program, such as TNTmap, can then be adapted to make use of these custom dimensions to enhance its capabilities and interactions with the end user. Any other client program that does not support these dimensions would simply ignore them since it can not use them.

Using Dimensions for a Styled Layer.

The response to a GetCapabilities request uses dimensions to inform the client program about the layers and their parameters available from a WMS (i.e., area covered, formats, maximum image size available, and so on). A GetCapabilities request to TNTserver can now respond with dimensions to inform TNTmap that it expects additional input parameters from the user to dynamically produce the styled layer rendered

as the image (e.g., sun angle and azimuth for shaded relief layer). The GetMap request for that layer then conveys those dimensions back to TNTserver to act on.

For example, the reply to a GetCapabilities request from TNTmap for a DEM layer includes a set of dimensions for that layer, if the layer has shaded relief enabled. These would be listed as the names of the parameters that could be applied to the image returned for the DEM. Subsequently, when the layer is viewed, the GetMap request for this layer may contain the dimensions requested to alter the shaded relief image returned.

Using Dimensions as Parameters to Style a Layer.

A more complex example might be that TNTmap permits the DEM to be rendered as a shaded relief. Under the DEM layer in TNTmap Builder, there would occur the styled virtual layer of Shaded Relief. This is created from the response to the GetCapabilities request for the layer that responds with the DEM and that any GetMap request for it must also contain the sun angle and azimuth dimensions. A user selecting this virtual layer would see a pop-in window with data entry boxes for sun angle and sun azimuth. The values entered would then be included as the optional or required dimensions in the GetMap request. TNTserver using the underlying TNTmips code would use these to compute the shaded relief and return it as the requested image.

This examples does not require that the shaded relief computations be created as they already exist in TNTmips. Using these and other TNTmips built-in capabilities via TNTserver are primarily dependent upon implementing the proper interfaces in the web application client to use them. However, remember that an installed TNTAtlas has many of these user-interaction interface components already available and is being adjusted to permit the use of a layer derived from a WMS.

Using SML Scripts to Style Layers.

For more complex virtual layers, you will be able to implement your own styles for the layer from a WMS using your knowledge of SML. Development is underway to permit SML scripts to define the needed dimensions for a GetCapabilities request, intercept these dimensions in the GetMap request, use them as input parameters to the script, process the layer, and deliver it via a TNTserver as the virtual WMS layer and eventually as a virtual WFS layer. When complete, the styles shown under the layer in TNTmap could pop in a dialog to be used to set parameters required by the SML script associated with it in TNTserver. For example, if the layer is a DEM then selecting the style name supplied by an SML script would pop in the dialog window to collect the parameters to be sent to the SML script via the GetMap request. TNTserver would then use the SML script to create and return that image. A more advanced SML script might perform complex processing on the real files and objects associated with the SML styled layer and when finished, send back an email to inform the user where the resulting file will be located for downloading.

TNTserver Manager.

Managing WFSs.

Alterations to the TNTserver Manager are required to accommodate its management of different kinds of OGC services. A new Service tabbed panel has been added for this purpose. It provides you with the ability to enable or disable specific operations of your TNTserver. For example, if you want to set up a TNTserver to only respond to WFS re-

quests, you can disable its operation as a WMS and WCTS. If you do not want to allow clients to have access to the WMS GetFeatureInfo operation, that operation can be disabled individually.

Publishing Virtual WMSs.

A web site can use separate CGI scripts to publish as many different access points (URLs) to its content as desired. Using this approach separate CGI scripts can access a single TNTserver and make it appear as multiple, different virtual WMSs. Each virtual WMS accessed by its URL/CGI can respond to a GetCapabilities request with a different title and content. Each CGI script you create can be used to define the layers it publishes, how they are published, if they are public or private, and so on. TNTserver Manager is used to manage and publish each CGI associated with each virtual WMS.

Since only one TNTserver is actually being used, these CGIs can control access to a common subset or separate set of geospatial data. For example, a layout might contain a collection of layers, some that are public and some confidential. One CGI could publish the general purpose layers in that layout for general public access. A second CGI would provide access to all the layers but require that TNTmap or your client program authenticate access via a user name and password. If you are using a farm (collection of computers) each running a real TNTserver, you could set up your load balancing using these CGIs to assign a client request for a particular virtual TNTserver to the next available or least busy real TNTserver.

Maintaining Synchronization with TNTserver.

TNTserver Manager can not be used to manage a TNTserver with altered or added features. When TNTserver is updated, TNTserver Manager is always updated in tandem. These matching version are provided together. TNTserver Manager now checks to make sure that it matches the version of TNTserver that is running. When both the "Start" and "Status" icons on the Control tabbed panel are enabled, it means that a different version of TNTserver is running on this system. Pressing the Status icon (the question mark icon), will provide information on the version of TNTserver that is currently running. Pressing the Start button (the VCR play button) will bring up a dialog asking if TNTserver Manager should stop the currently operating TNTserver and start the version of TNTserver that corresponds with the version of the TNTserver Manager that you are running.

Using Abstracts.

The WMS specification permits the response to a GetCapabilities request to contain an abstract for each layer, group, and layout. These abstracts can be used by the client program to show the user additional metadata about each available layer. TNTmap Builder already uses a pop-in to show this metadata, if it is available, when the cursor is positioned over the title of each layer, group, or layout name. The text content of the abstract for each layer, group, or layout is entered using the TNT text editor accessed via the Quick Edit buttons on the Layers tabbed panel. The WMS specifications do not define any text markup to format the appearance of this abstract in the client program. The use of HTML to style these abstracts is being investigated. (This feature has also been added to version 2006:72 for layouts.)

Altering Published Content.

The CGI setup wizard leads you step-by-step through the setup of the components you are publishing via a specific CGI with your TNTserver. However, this can be a tedious procedure for subsequently changing or adding to the content you already have published. Quick Edit buttons are now available where appropriate to let you directly change and add entries without repeating all the steps in the wizard:

- on the Web tabbed panel to permit you direct, easy editing of the individual virtual WMS titles associated with each CGI;
- on the Layers tabbed panel permits the direct editing of the Valid CGI List entries for layers, groups, and layouts;
- on the Layers tabbed panel lets you create or edit the abstract for a layer, group, or layout; and
- on the Layers tabbed panel allowing you to change the logo file you are using for a layout.

An individual layer in a group can now be turned off using its check box on the Layers tabbed panel. The layer will then not be included in the layer list returned to a client in the response to a GetCapabilities request for that group or the layout containing the group. Similarly a group can be turned off using its check box.

Miscellaneous.

“View Latest System Log” and “View Latest Status Log” buttons added to the Log tabbed panel provide for a quick and direct review of these logs.

An entry for “Hours of Service” and a button for contact “Instructions” have been added to the information entered on the Contact tabbed panel.

TNTmap

TNTmap has provided an excellent model for the incorporation of the use of WMS derived layers in other TNT products. All the necessary concepts and functionality were developed over the past 18 months in TNTmap Builder for the layer browse and selection and TNTmap Viewer for the layer retrieval and management.

TNTatlas now incorporates access to WMS layers. As a result it is now an alternative installed client for any WMS. TNTmap is designed for general public use while TNTatlas can provide more complex tools for using web published geospatial materials and can be customized by you using SML scripts. Even TNTlite is now a potential client for WMS layers that can perform complex activities combining geodata from local, VPN, and Internet sources.

Version 2006:72.

The features in this version of TNTmap have not yet been finalized, so it is still changing almost daily. Adjustments to TNTmap Builder continue to improve the logic of its interface. TNTmap Open is still being modified to provide the same features as TNTmap Viewer. Samples of using only server-side HTML to set up the user interface for TNTserver access are being prepared.

When finished, the widget version of TNTmap will be submitted to Apple for possible public distribution. It is already a far more complex widget application than any of their other 2500 public widgets.

Version 2007:73.

The HTML layout of DataTips will be sent to and used to style the test in feature information window.

A gadget version, based on the Mac OS X widget version, will be subsequently released for use in Windows Vista (gadgets are similar to widgets).

A Microsoft Live viewer has already been implemented in TNTmap and will eventually be released as an additional viewer.

Technical Developments

Raster Pipeline Coding Primer.

Many of the raster processes in the TNT products are being recoded to use the concepts of a pipelined code stream. This primer section introduces this approach to coding to help you better understand the subsequent sections on the status and use of these revised processes.

Introduction.

From Wikipedia (search PIPELINE SOFTWARE).

"In software engineering, a pipeline consists of a chain of processes or other data processing entities, arranged so that the output of each element of the chain is the input of the next. Usually some amount of buffer is provided between consecutive elements. The information that flow in these pipelines are often byte streams or bit streams."

In pipeline coding the operational or processing modules are called stages and the connections between them are the pipes. This creates an analogy with physical water systems where stages (e.g., a pump in a lift station or a valve) act according to upstream or downstream actions and the pipes act as conduits of the results and any requirement for further action (e.g., put in storage, divide flow, stop flow).

Stages.

Each component in the pipeline is called a stage.

There are three types of stages:

Source. A source stage has no pipeline inputs and one output. A source can access an image from an external file, device, memory or even a mathematical function.

Filter. A filter stage has one or more inputs and one output. A filter will usually derive some of its properties from its input stage(s). Most filters will have the same dimensions and georeference as their input stages.

Target. A target has one input and no pipeline output. A target is designed to write an image to external file, device, memory, or discard it entirely. A target stage derives its properties from the input stage it is connected to.

Every TNT stage has the following properties:

Dimensions. Its total rows, columns, and tile rows and columns.

Pixel. Its type (i.e., RGB, grayscale, CMYK, etc.), number of samples (pixel components), and sample properties (i.e., sample type, data type, and precision).

Georeference. Its transformation to non-image coordinates with the required model and parameters.

Connecting Pipes.

A valid pipeline has exactly one target stage, connected to the stage it pulls from. A minimal pipeline would connect the target directly to a single source. More complex pipelines may have many sources and filters, but still only one target. Note however that a source or target is not limited to using a single raster object. Conceptually, these would read or write a complete image, which may be made up of one or more raster objects (e.g., separate RGB, hyperspectral, etc.), all having the same dimensions and geo-reference.

Pipeline Flows.

Once the pipeline is constructed, a single function is called on the target stage to control the entire processing, pulling data from its input stage and writing it to its destination. This processing is done in sections, where a section specifies a rectangular portion of an image. The sequence of processing is determined based on a negotiation between the target and its input stages, and can be by tile, by strip or line-by-line, depending on what is determined to be optimal and what the stages are capable of doing. For example, if the target is an untiled JPEG2000 image, processing is done in strips, rather than tiles, because the image must be written in that order.

TNT Pipeline Coding.

The raster processes in the TNT products are being converted to use a pipeline coding paradigm. This has been completed in various processes and recoding will be continued in version 2008:74 of the TNT products.

Pipelining the TNT raster processes is a prerequisite for subsequently using multi thread coding to take advantage of the multi-core computers you will soon have on your desk. Note that multi-threaded processing of rasters does not automatically occur by simply recoding to use the pipeline approach (i.e., it does not subdivide the image into segments and send these to the various cores). Each stage used in TNT raster processing must be analyzed to determine if, and how it can use multithreading to increase its throughput and performance. (FYI, TNTserver 2006:72 is already multithreaded.)

MicroImages pipeline development efforts in version 2007:73 focused on converting the portions of all processes that communicated with any raster geodata files or raster objects to use a common set of RVC classes. This portion of the conversion has been completed and enables modifying for version 2008:74 to use multithreading where appropriate to improve performance and enable new features. New features you have requested were also added to some of these processes as they were converted to use stages. For expediency some processes were only pipelined without addressing the addition of any new features or changes.

Benefits of Pipeline Coding.

Recoding to use the pipeline approach has benefits other than providing the framework for multi threading. The generic stages created for use in our pipeline coding approach are much easier to add into a new or existing process where appropriate (e.g., the new Tile process). For example, a TNT filter stage is now available to convert from one CRS to another CRS and is used in the display process. This stage can subsequently be inserted into other process's for this purpose without modifying the stage (but the processes user interface must still be adjusted).

The image compressions supported in the TNT products are now implemented in stages. The stage using the Kakadu JPEG2000 library licensed by Microlmages has already been upgraded to use multi threading for both compression and decompression. As a result, the JPEG2000 compression/decompression stage in the TNT products has been automatically converted to be multithreaded in all processes reading and writing JP2 files as well as in the use of JPEG2000 compression in a raster object (e.g., import, export, extract, mosaic, and display). As a result an export to or from a raster object in the JP2 format is 3+ times faster using a system with 4 cores. Stages are also now available for use in SML processes. Many stages will connect together automatically but the script will still require the design and implementation of the user input/control of the process.

Thread Safe Project File Management.

The use of a single integrated Project File is the foundation upon which the TNT products have been built for over 20 years. The evolution of the capabilities of the Project File over those 20 years has been as transparent to you as possible. In the early years it had only 1 user, you, and early computers could not multitask. When your computer advanced to support multitasking, the entire Project File was locked when it was in use by a process to protect its contents. Now this evolution in Project File management must continue to support the effective use of your multiple cores and the even more interactive multitasking, parallel processing, and the user-friendly features they enable. To accommodate this the objects in the Project File management must become thread safe.

A thread safe Project File management must support the interaction with multiple compute threads, each of which can be doing something different to an object in an unpredictable order. A simple example would be for a TNT image analysis operation that starts 4 separate threads to process 4 separate pieces of a single raster object. Each available core may be called upon by the operating system to service one of these threads but it can be interrupted by other tasks from other threads assigned to it with a higher priority by the operating system. There is no guarantee that all their processing of the raster pieces will begin or finish at the same time. The Project File management must safely interact (i.e., be thread safe) to permit coding the TNT processes to use multithreading and to use the multiple cores effectively in the TNT processes. In summary, when the Project File management is thread safe, TNT process(s) can create and use multiple threads.

The coding to change the Project File to be thread safe has been completed for some time. In fact, it was used in the TNTserver 2005:71 product. Obviously its utilization is transparent to you since Project Files created in TNTmips 2005:71 and 2006:72 were usable in TNTserver 2005:71. However, the thread safe version of the Project File management code has not been incorporated into version 2007:73 of the TNT analysis products (TNTmips, TNTedit, TNTview, or TNTatlas) since some of the processes have not been converted to use it. The conversion of all processes to use this new RVC code will be completed about the time of the release of 2007:73, but it would be unsafe to release it into the TNT analysis products this late in the development of the 2007:73 release. Thus the thread safe Project File code will be in the first release of the 2008:74 development version of the TNT analysis products in a few months. At that time multithreaded coding will be introduced into the TNT libraries and processes to enhance performance and permit more interactive concurrent operations in the TNT user interface

and interaction between TNT processes and concurrently running independent programs.

Why Use Separate Null Masks?

There have been questions about why null masks were introduced into the TNT products. In part these were questions prompted by the fact that null masks were not fully implemented in all possible processes. The storage cost of using null masks is very small. They are binary rasters that are very quickly and highly compressed and only created at all if the raster has null cells. Null masks are more versatile than null values and are designed to address the following issues.

Computed Value Can Match Null Value.

Values from raster operations on non-null cells can result in a value for the cell that matches the null value of a raster (e.g., 0, 255, or ...). Prior to using null masks, a test of the value resulting for the computation of every cell had to be made, and if the value did match the assigned null value, an adjustment was made to that result for that cell. This adjustment resulted in numerically incorrect, different values for some cells, and made the process take more time to run.

No Value Has been Reserved for Nulls.

One of the strengths of the TNT products is in their creation and management of irregularly shaped raster objects including raster objects containing holes. Processes such as raster extraction, reprojection and mosaic can easily result in rasters having null cells. If the source did not specify a null value, one was assigned, and then all valid cells had to be tested for the null value and adjusted if they matched, since there is no guarantee that there are any unused cell values in the entire data range.

Combining Rasters with Different Null Values.

You can freely select raster objects and other geodata file formats from different sources for use in a TNT process. You can not ensure that each uses the same null value. This can result in combining rasters with differing null values, or some not having them defined at all. In many cases you may not be aware of what null values are used or of what the resulting range for the operation may be, making the choice of a null value difficult. Converting null cells to a null mask ensures that the null cells are correctly combined and a null mask is created for the result.

Lossy Compression Changes Cell Values.

If lossy compression is used, it is not possible to ensure that null cells retain their original value, nor that non-null cells are not altered to the null value. Use of a null mask, which is never lossy compressed, will ensure that null cells remain null and vice versa.

Null Values Still Used For Direct Linking.

Using null masks does not eliminate the TNT support for having a defined null value. In the case of an import, if a null value is defined, TNT will set the null value for the raster instead of creating a null mask. It is more efficient in both time and space and required when linking to the raster whether manually via the import process or automatically where supported. The only case where this might be a problem is if you alter the cells of the raster in-place after importing, which is unlikely except in TNTedit.

Miscellaneous Benefits.

Simpler User Interface. The use of null masks simplifies the interface as you do not need to specify null values except in the import and export processes.

Simpler Code. Null masks simplify the TNT code resulting in improved reliability.

Faster Results. Processing is faster using null masks as the validity tests, some of which are noted above, are simplified and in many cases can be skipped entirely.

More Accurate Results. Using null masks means that no valid raster cells need to be altered to prevent them from being interpreted as null.

Identifying a Null Value.

Ideally, the only places where a null value should be of concern are when using raster geodata files produced outside of your TNT product and when exporting rasters. Unfortunately, geodata is sometimes imported or used from other sources without regard for, or knowledge of, their incorporation of a null value. For this reason, the "Set Null Cells" utility is also provided to manage or adjust the null value of existing internal raster objects.