



7 January 2009

TNTmips Job Processing

The DV2009 version of TNTmips introduces a new Job Processing (JP) concept to assist you in optimizing your geospatial analysis activities. It permits you to use the interactive TNTmips interface to set up large jobs, many similar jobs, and/or SML scripts to run during off-hours and to automatically use many cores, as a distributed system, and other setups.

How does Job Processing work?

When you start up TNTmips DV2009 it automatically starts the new TNT Job Processing feature. You see nothing new unless you choose the Job Manager from the TNT Tools menu.

Many TNTmips processes now have a new Save Job button where you find the other task disposition buttons, such as Run and Cancel. When you are ready to run your task, and would select Run in TNTmips 2008, you can now choose to Save Job. This will create an XML job file in a specified TNT Job folder and directory. For example, you select 100 Landsat scenes to import and losslessly compress with JPEG2000 into TNT raster objects. When you use the Run button, TNT immediately begins to run 100 sequential imports (i.e., serial one at a time). If you choose the Save Job button 100 XML job files will be written to the Jobs folder. The Job Manager window will open automatically to permit you to choose to immediately run these jobs in parallel on multiple cores or save them to run later. As each job is completed, its job file is moved to a Jobs Done folder where it remains until purged.

Running SML scripts can also create XML job files if they have no user interface or an interface that incorporates the Save Job button. Your script can write out an XML job file to be immediately executed or managed. Since any job file is humanly readable, the XML job file created by a TNTmips process or script could be used as a template and altered by your program to change some parameter(s) such as the geodata file name, path, and/or URL. For example, your program or web process could be monitoring some independent condition(s) and use the results to alter a job template and write the complete XML job file into the TNT Jobs folder for execution. A job file can even be created independently of TNTmips using a text editor or your program.

On a 1 real-core computer.

If you are using a computer with 1 real core, the Save Jobs option in processes or scripts can be used to determine when, and in what order, these TNTmips tasks should be automatically executed. This permits you to set up the tasks in TNTmips that are computationally intensive to run overnight and frees up your daytime computer resources for your use of other software. A typical application might be to set up the resampling and reprojection

of 100 images to be automatically started at 6 PM. Since you have only 1 core available these jobs will run sequentially but will be complete the next day.

On a 2, 4, 8, ... real-core computer.

When used on a single computer with 2 or more cores, you can designate how many cores are to be used by TNTmips to run your saved jobs. Your dual core computer might be set up for Job Processing to use only 1 core to sequentially import files in the background, thus reserving 1 core for your other daytime activities. In this mode your saved resampling and reprojection jobs might be set to run at once using 1 core since the results are needed as soon as possible but you have other things to do in the meantime. If the same saved jobs are to run overnight, TNTmips can be set to use both cores for these jobs.

Quad core desktop machines, even portables, are now readily available and a quad core processor will become common in desktop replacement computers in 2009 using Intel's new Core i7 chips. Using dual Core i7 chips 8 real cores will be economical for production units. For 4 real cores you could set TNTmips Job Processing to reserve 1 for your interactive use in other software or for setting up other TNTmips jobs and 3 for new background TNTmips job processing. All 4 could be set for overnight job processing. For 8 real cores a 2 and 6 approach could be used during the day and all 8 provided for overnight processing. This is typical of the way this process is being tested and applied at MicroImages.

Using SML scripts.

SML scripts can be used to set up jobs for TNTmips to process just as though the job was set up in an internal TNT process. The script is designed to interact as usual with the user to collect input and set up the custom job. However, the XML job file is saved to be run just as if it was created by a Save Job button in a TNTmips process.

On a distributed system.

Jobs set up by TNTmips processes or scripts do not have to be run on one or the same TNTmips computer. If you have multiple computers using TNTmips keys or a floating license to several TNTmips seats they can all be set up to run overnight to process a days accumulation of production tasks.

As a geoprovisioning service.

MicroImages sample geoprovisioning site at geospatialgateway.com uses Job Processing and SML scripts. An end user can select from a suite of web applications with different objectives. Each provides a different interface to collect the required input for an associated SML script. This input is sent to a PHP script on your web server that structures it into the XML job file and writes it to the TNT Job folder. In place of PHP you could use your favorite server-side scripting language for this such as ASP, Perl, or Python.

Each job file contains the identity of the SML script that created it and that TNTmips should run to process that job file. The TNTmips Job File directory can be located anywhere on the LAN. One or more TNTmips systems can be on running continuously on the same network computers and poll the Job folder to see if any job files are available. The geodata needed for any task can be accessed from a network drive, a file server, or the local drives if only one TNTmips is used. Each TNTmips logs out 1 or more jobs from this common queue according to the cores it has been set up to use as described above, the priority set in the script for this type of job, and other TNT Job Manager settings for each TNTmips station.

As TNTmips completes a job the script writes the resulting file(s) to an FTP site and sends the user an email informing them that their geoprovisioned job has been completed and providing a link to be used to download it. Various other approaches can be used to notify the user or other programs that the results of each geoprovisioning job are complete and available for use.

The operation of a TNTmips-based geoprovisioning service is described in more detail at www.geospatialgateway.com/75MemoGeoprovisioning.pdf and can be tried at www.geospatialgateway.com.

What is Asynchronous Multicore Processing?

TNTmips uses Asynchronous Multicore Processing (AMP) to run your saved jobs in the background or on the schedule you provide. AMP is a simple concept. You have always been able to manually start several similar or different TNT processes to run concurrently and share your core or cores. AMP automatically applies this same logic to processing all the jobs you have saved. Suppose you designate that 3 cores are available to process your TNT tasks in the background. Job processing will start 3 TNT tasks using the same or different processes each using a single core (i.e., multicore processing). When any of these jobs is completed and a core becomes available, a new job will be given to it (i.e., asynchronous tasking).

How does AMP relate to multi-threaded Processing.

Current literature is filled with articles discussing how complex it is to perfect multi-threaded applications with current compiler, language, and debugging tools. Converting processes to take advantage of multi-threading must proceed slowly and carefully while awaiting improvements or replacements for these tools. These are arriving very slowly. However, Multi-threading is useful in products like TNTmips in several areas: 1) improved user interface operations—like buffering tiles to support panning, 2) background network related tasks—loading image layers from the Internet while manipulating local layers, and 3) increasing task production – such as reprojecting and exporting many images. TNTmips 2008 addressed objective 1) and 2). Job Processing and AMP in TNTmips DV2009 address objective 3) in many processes without rewriting these batch-oriented processes to be multi-threaded.

AMP can use TNTmips processes that have been modified to use multi-threading. AMP is simply a combination of using TNT Job Processing to automatically accomplish multi-tasking. If you can use all your computer's cores at maximum capacity it makes little difference if this is accomplished by AMP or multi-threading or a combination of both. TNT Job Processing using AMP circumvents temporary multi-threading coding complications to use all your cores, often at maximum.

The Kakadu library used for TNT JPEG2000 compression/decompression is multi-threaded. You can link to a large JP2 file and export it to a new JP2 file to change its compression ratio (e.g., from lossless compression to 15 to 1 compression). This single task decompresses the image in segments and recompresses it using multiple cores. This is almost all continuous CPU activity as can observe from the use of your 1 or more cores using your OS task manager. If you let TNTmips use 4 cores for this single task, you observe

that all 4 CPUs or all 4 cores are operating at about 25% using multithreading with short interruptions to read and write. Using TNTmips 2008 you could manually, sequentially start 4 or more export processes of this type (large files) and see your CPU time approach 100% for all 4 cores. TNTmips DV2009 simply automates CPU usage using AMP. Simply select the same 4 or more files and export them all as one job via the Save Job button. The AMP approach in job processing will use all 4 CPUs at about the 100% level. In this case you do not know or care if multi-threading or AMP is at work to maximize core use since the result is the same.

Job Manager process.

Many different kinds of jobs and scripts can be queued up for processing via the TNT Jobs folder. They can come in from TNTmips processes, SML scripts, manually prepared XML files, an HTML web page, a web application, your program, a geoprovisioning site, ... The Job Manager processor in DV2009 can be opened at any time from the TNT Tools menu and opens automatically when the Save Job button is used in any process. It provides the interface needed to manage and monitor how and when each XML job file will be processed. This window is where you set the number of cores that TNTmips can use when it is running jobs. It also provides a toggle to enable all new jobs to run immediately when added to the queue or to hold new jobs so you can set their status and priority. Many more features are planned for this new process in addition to those already available on the following tabbed panels.

Job List Tabbed Panels.

The Job Manager interface provides Queue, Done, Failed, and Settings tabbed panels. The Queue panel lists the status, priority, ID, name, and creation time of each job currently in the queue as well as additional metadata about each job such as a process ID, name, The job status of each job can be run, running, hold, delete and other states. The job priorities are numbers 1, 2, 3, ... You can select and change the status or priority of a job or group of jobs in this panel. The Done panel contains the metadata about the jobs that are done in the order they were completed. When a job is listed in this panel its XML job file has been moved to the Jobs Done folder. The Failed panel contains the metadata about jobs that did not execute correctly and are still in the Jobs folder and have been skipped.

Settings Tabbed Panel.

This panel lets you set the path where the Jobs folder and Jobs Done folder are located. These locations can be set to anywhere on your network and set to be the same file and location in several TNTmips systems. You can set the delayed time for running all jobs from this panel and other control parameters.

Processes supporting Job Processing.

The following are TNTmips' processes that have been modified to provide Save Job buttons. If a process has an interactive capability, it will not provide a Save Job button. For example, if a polygon is used to define an area the Save Job button will not appear and the task will run immediately. Your tasks that require interaction to outline a specific area, select features, ... are usually performed one at a time. At the present time coding effort is focused upon improving the Job Manager. However, if you have a justification for adding a

Save Job button to some TNT process not listed below please communicate it to MicroImages for possible implementation.

Processes already modified to use Job Processing.

| | | | |
|----------------------|------------------------|------------------|---------------|
| SML | Raster Resample | Vector Reproject | Pyramid |
| Principle Components | Topographic Properties | Decorrelate | Convert Color |

Processes being modified to use Job Processing.

| | | |
|--------------|---------|--------------|
| Buffer Zones | Extract | Autoclassify |
|--------------|---------|--------------|

Import/Export formats that use Job Processing.

| | |
|---------------------------------------|--------------------------------------|
| ArcBIL/BIP (.bil, .bip) | IRS |
| Arc-Grid (.adf) | JP2, GeoJP2 (.jp2) |
| Arc-Shapefile (.shp) | JPEG (.jpg, .jpeg) |
| Arc-Text | Landsat-NLAPS (.hd, .hi, .dh, .h1) |
| ASTER-HDF (.hdf, .dat, .l1r, .l1g) | Landsat7-HDF (.hdf, .dat, .l1r, l1g) |
| BMP (.bmp) | LAS, LAS-DB (.las) |
| CADRG | MODIS-HDF (.hdf, .dat, .l1r, .l1g) |
| CCRS (.tm) | MrSID (.sid) |
| CGM (.cgm) | NetCDF (.nc, .cdf) |
| DCR (.dcr) | NetCDF_GMT (.grd, .gmt, .nc, .cdf) |
| DEM (.dem) | NITF, NITF2.1 (.ntf) |
| DEM-GTOPO30 (.dem) | NTAD (.geo) |
| DGN (.dgn) | NTF-vect (.ntf) |
| DIMAP | PCI (.pix) |
| DOQ (.doq) | PCX (.pcx) |
| DTE (.dte) | PNG (.png) |
| DWG (.dwg) | PRISM |
| DXF (.dxf) | RADARSAT |
| ECW (.ecw) | SDF (.arg, .pnt, .pgn) |
| ER-Mapper (.ers) | SDTS-DEM (.dbf) |
| ERDAS-IMG (.img) | SRTM (.hgt) |
| Geosoft-XYZ (.xyz) | SunRast (.ras) |
| GIF (.gif) | Surfer (.grd) |
| GIRAS | TIFF, GeoTIFF, BigTIFF (.tif, .tiff) |
| Google-Local (.jpg, .jpeg, .png) | TIGER (.rt*) |
| GRASS-Raster | TM-Fast-L7A |
| HDF4-Generic (.hdf, .dat, .l1r, .l1g) | |

Import/Export formats that do not support Job Processing.

| | |
|-----------------------|----------------------------|
| Access (.mdb) | DB-Text (.txt) |
| ADRG (.thf) | DTED (.dto, .dt1, .dt2) |
| Arc-Coverage | ENVI (.hdr) |
| Arc-E00-Grid | ERDAS-GIS/LAN (.lan, .gis) |
| Arc-E00-Vector (.e00) | ERS-SAR |
| Arc-Generate | Excel (.xls) |
| AVIRIS (.img, .fl) | Geosoft-GRD (.grd) |
| CARIS (.txt) | Geosoft-GXF (.gxf) |

GML (.gml)
IDRISI (.rdc, .img)
ILWISR (.mpr)
JERS1
KML (.kml, .kmz)
Landsat-CCRS
MapInfo, MapInfo-Attrib (.tab)
MIF (.mif)
MOSS (.mos)
My_SQL
MySQL_Spatial
ODBC (.dbf, .ndx, .mdx, .xls, .mdb, .dbc)
Oracle
PostGIS
PostgreSQL
Simple-Array
SPOT (.dat)
SPOTView (.hdr)
SQLServer
Text-Raster (.txt)
Text-XYZ (.txt, .csv, .xyz)
TM-Fast
User-Defined