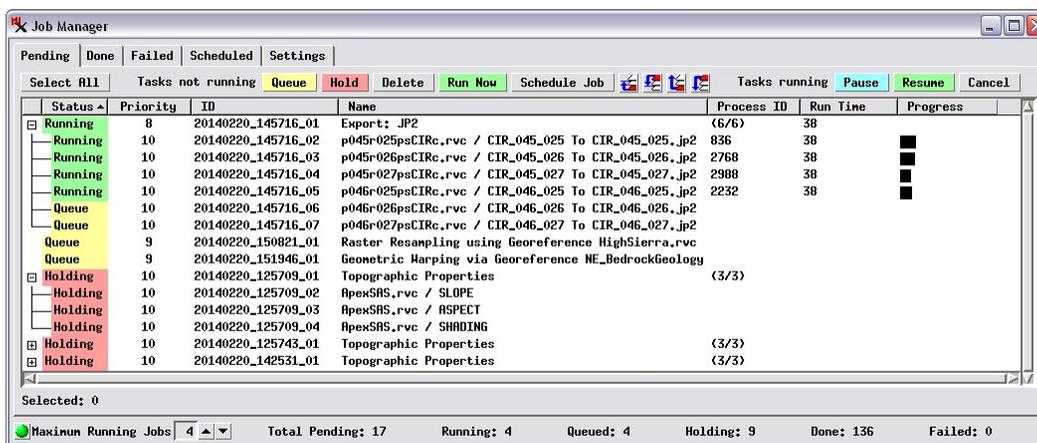


# TNTmips Job Processing System

The TNTmips Job Processing System (JPS) and the associated Job Manager are designed to significantly improve throughput of your production projects, exploit your computer's multiple cores, and interact with other software products and the Internet. Tasks can be defined by interactive TNTmips processes, geospatial analysis scripts (i.e. SML), web applications, and your own programs. Each task is an XML job file that defines its input and output parameters and the TNT process, SML script, or other program it controls and executes. When TNTmips is running, the TNT Job Processing System (JPS) is automatically activated in the background to detect, prioritize, and run multiple TNT tasks concurrently. If no tasks are being executed, this background process uses almost none of your computer resources.

The XML job file for every task you create is written into a predefined Jobs directory. The TNTmips Job Manager provides the interface to manage all the current tasks in this directory. You can use the Job Manager to set the priority of each job, how many tasks are to be executed in parallel, schedule when tasks are to be executed, review the status of every task, and so on (see the Technical Guide entitled *System: Managing Job Processing*).



The TNTmips Job Manager window opens automatically (if not already open) when new job files are created. It allows you to select pending jobs and change their status (hold, release, delete, or set to run with the next available processor thread) and shows run times and progress bars for current running jobs. Lists of completed and failed jobs are shown on separate tabbed panels. You can also set up time and day-of-week scheduling for jobs.

JPS permits you to run many identical or mixed TNT processes concurrently in the background. Even though these JPS tasks are running in the background, you can continue to interact with other TNT processes or other applications. To exploit your computer's multiple cores you can specify a maximum number of tasks that can run at the same time. If you have only a single core machine, JPS can still increase your throughput by running several tasks concurrently, especially if you program them to start when you are not actively using your computer.

## Using Interactive TNT processes

Many TNT processes support job processing and are provided with several execution buttons. In addition to the standard Run button (to bypass JPS), these windows have Queue Job and Save Job pushbuttons or icon buttons. When you select the Queue Job or Save Job button the task or multiple tasks you have defined in that TNT process creates one or many XML job files in the Jobs directory. Queue Jobs adds them to the task queue to be executed as soon as an open job slot becomes available. Save Jobs adds them to the task queue with a status of *holding*. Choosing either of these buttons automatically opens the Job Manager window, if it is not already open, to permit you to manage the order and priority of the new task(s) relative to the other tasks waiting or underway.

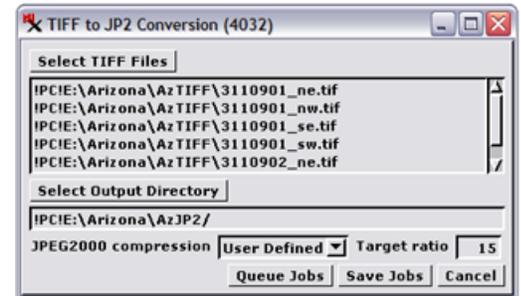


TNT processes that support batch processing, such as the Import and Export processes, set up a job file for each complete process operation and a group job file to allow you to easily manage the entire set of jobs. For example, if you export 1000 MrSID files to JP2 files, then 1000 XML job files are created with a status in your job queue of *holding* or *queue*, along with an Export group job file. A simple example of using Job processing is illustrated in the Technical Guide entitled *Export: Concurrent Exports Using Job Processing*.

(over)

## Using Paired Scripts

You can use paired SML scripts (an Input/Output or I/O script and a Process script) to set up many job files to accomplish unique production tasks. Your I/O script provides the interactive user interface to set up one or many job files. An SML class is available to write out the XML job file from within your script. Each job file created by the I/O script contains the input and output parameters for the task and the identification of the P script to be executed. A typical application might be a P script to convert thousands of multiband GeoTIFF images of more than 3 bands to equivalent multiband GeoJP2 images. The I/O script permits batch selection of the GeoTIFF files and the naming or autonaming and location of the GeoJP2 files. Sample I/O and P scripts for this application are documented in the TechGuide entitled *System: Custom Job Processing with Geospatial Scripts*.

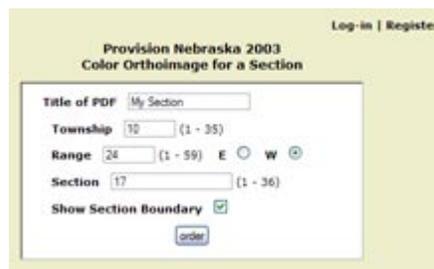


## Creating Jobs from External Sources

Each job file contains the identity of the TNT process or SML script that the JPS will execute and the input/output parameters to be used. The input and output parameters in an XML Job file can be created by web applications or custom programs written in any language (e.g., BASIC, Java, ...) and these applications can even reference dynamic sources such as sensor or other data collection systems. Any of these methods of creating a job file can result in queued or scheduled execution of the task specified by the SML script or TNT process. Your geospatial application might be quite sophisticated. For example, it might monitor a sensor network or fleet of vehicles and update displays; create other job files; or provide custom reports and maps based on user interactions and queries. The proper format of an XML Job File is documented in the TechGuide entitled *System: What Are Job Files?*.

## Provisioning Geodata

MicroImages' sample geodata provisioning site uses job processing to execute each user request. The various web clients illustrated here use free MySQL and PHP installations to create job files. These job files then call the appropriate SML script to complete the geoprovisioning task with the input provided by the HTML form or web application. The processing script also sends an e-mail message notifying the user that their results are available. Try the provisioning site at [www.geoprovisioning.com](http://www.geoprovisioning.com).



## Launching Other Applications

An SML script can call a simple function to launch another application program and pass command-line parameters to it. Thus, a script that is run by a job process can create geospatial or other data and pass it to another application. For example, a job file could run an SML script to create a KML file and then use it to launch Google Earth. Or a script could write tabular data to a CSV file and use it to launch a spreadsheet program or pass it to a custom local program or web application for processing or dissemination. A TNTmips user can thus combine TNTmips Job Processing, SML, and custom local or web applications to carry out complex geospatial analyses, transmit information into other commercial programs such as spreadsheets or databases, and create analysis results in popular formats such as PDF or SVG.

