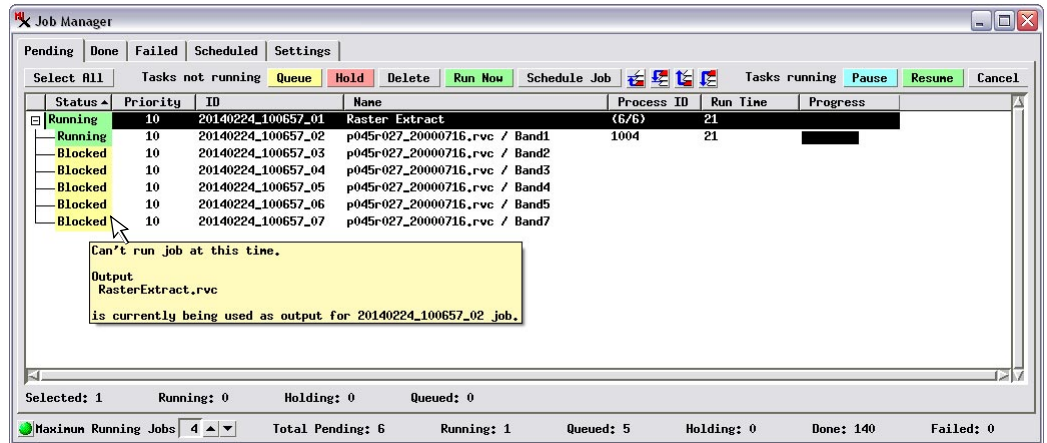# Directing Outputs to Enable Concurrent Job Processing

One of the main benefits of the TNTmips Job Processing System (JPS) is that it allows you to easily run multiple concurrent processes to maximize the use of your computer's multiple processor cores. The JPS Job Manager allows you to specify the maximum allowed number of concurrent jobs and to manage and monitor the progress of these jobs (see the Technical Guides entitled _System: Managing Job Processing_ and _System: Managing the Job Queue_).

Multiple jobs can be run concurrently using the JPS, limited by the value you set for the Maximum Running Jobs parameter. However, only one TNTmips process (and thus only one job) can write to a specific output file at a time. If you start a series of jobs that direct their outputs to the same file, the JPS will automatically run them one by one, rather than concurrently, regardless of your Maximum Running Jobs parameter setting. Jobs that are currently blocked from running due to contention for the same file remain in the job queue with their status set to _Blocked_. These blocked jobs are run in the order shown in the Job Manager's Status column, which you can control by changing their priorities as discussed in the TechGuide entitled _Managing the Job Queue_. If blocked jobs are followed in the Queue by other jobs that do not require a file that is currently being written to by another job, the JPS will automatically skip the blocked


All of the jobs shown above in the Job Manager's Pending tabbed panel specify the same Project File for their output. The currently running job has placed a write lock on this output Project File; each of the remaining jobs needing to write to this file is blocked from running until all of the preceding jobs have completed. Queued jobs blocked by a file lock are indicated in the status field as _Blocked_. Hovering the mouse over the status field also reveals a DataTip that explains the lock status.

jobs temporarily and run the next nonblocked jobs and thus proceed with concurrent processing up to the Maximum Running Jobs value.

You can avoid the JPS file contention issues outlined above by setting up jobs so that each one writes its output to a separate file. Tools and strategies for directing outputs to different files are discussed below for different types of TNTmips processes.

Processes Making a Single Output Object (Single Job File)

Some TNTmips processes that support Job Processing produce a single output object from one input object (or a set of input objects). An example is Vector to Raster Conversion (Geometric / Convert / Vector to Raster), which has a number of data-specific process settings and thus allows you to select only one input object at a time. Pressing the Queue Job or Save Job button in this process creates a single job file that launches one job process utilizing that input/output pair of objects. Setting up a series of such jobs requires reselecting the input and output objects individually for each job. In order to ensure that you can achieve concurrent processing of a series of such jobs, you only need to take care to direct the output object for each job to a different Project File using the familiar file and object naming procedures that you use with any TNTmips process.
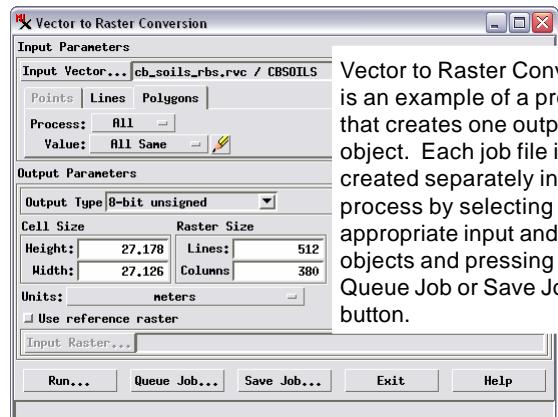

Vector to Raster Conversion is an example of a process that creates one output object. Each job file is created separately in the process by selecting the appropriate input and output objects and pressing the Queue Job or Save Job button.

## Processes Making Multiple Output Objects (Multiple Job Files)

A number of TNTmips processes that support Job Processing can produce multiple output objects from one or more input objects. For example, Raster Extract (Image / Extract), Raster Resampling (Image / Resample and Reproject / Automatic), and Geometric Warping (Geometric / Reproject) allow you to select a number of input objects and create one output object for

(over)
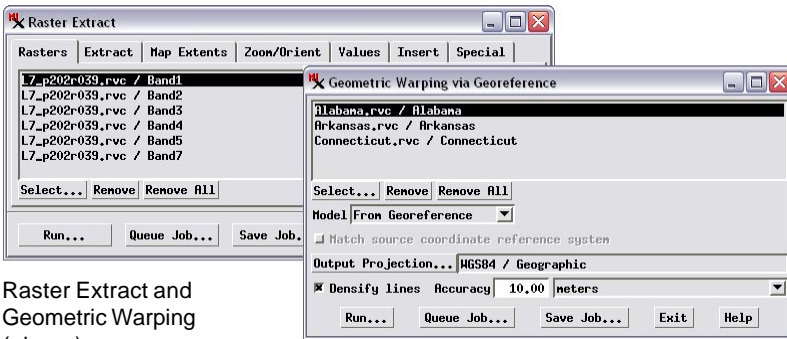
Raster Extract and Geometric Warping (above) are processes that allow selection of multiple input objects and produce one output object for each. The Topographic Properties process (right) is an example of a process that can create more than one output object from a single input object (a digital elevation model raster object). The Queue Job and Save Job buttons in these processes automatically create a separate job file for each output object. These jobs can then run concurrently if each of these output objects is directed to a different Project File.
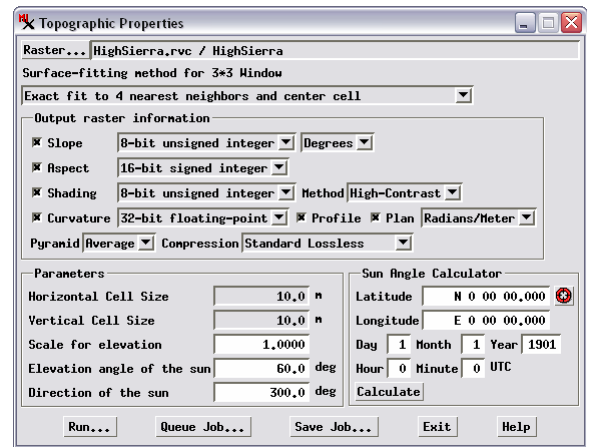
each input object. The Topographic Properties process (Terrain / Topographic Properties) lets you create various output products from a single input elevation raster. Pressing the Queue Job or Save Job button in any of these processes automatically creates a *separate* job file for each output object. These jobs can run concurrently if each of these output objects is directed to a different output Project File.

You can automatically create a separate Project File for each output object in such processes by using the Auto-Name button on the Select Objects dialog. You may have used this button to automatically name all of the new objects *after* you selected an output project file. However, this button is also available *before* you have selected and navigated into a particular Project
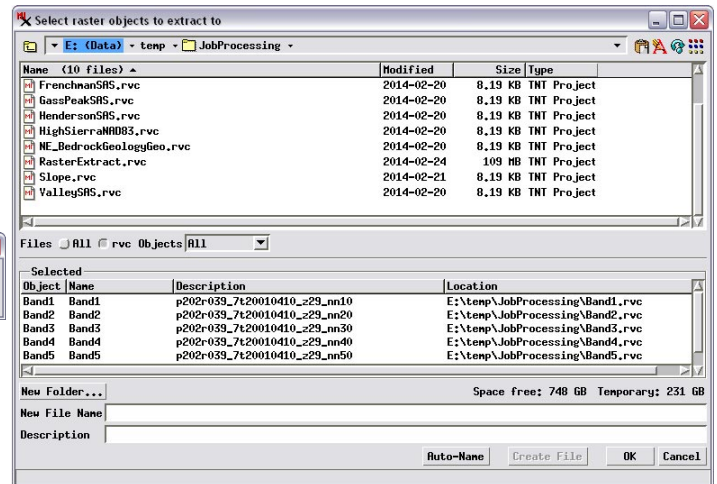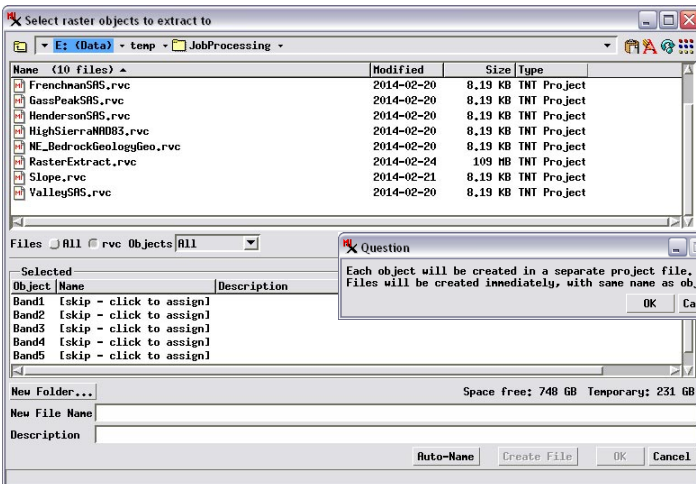
Select Objects dialog for designating the output rasters for the Raster Extract process window shown at the top left of this page, opened using the Queue Job or Save Job button. Pressing the Auto-Name button while at the file level of navigation (above left) immediately creates a separate new Project File for each output object. The name of each input object is used to name its output Project File and object (result above right).

File. Used at the file level of navigation, it creates a separate new Project File for each output object. The input object name is used to name the corresponding output Project File and output object. (If any of these file names is already in use within the selected directory, you are prompted to choose a different output directory).
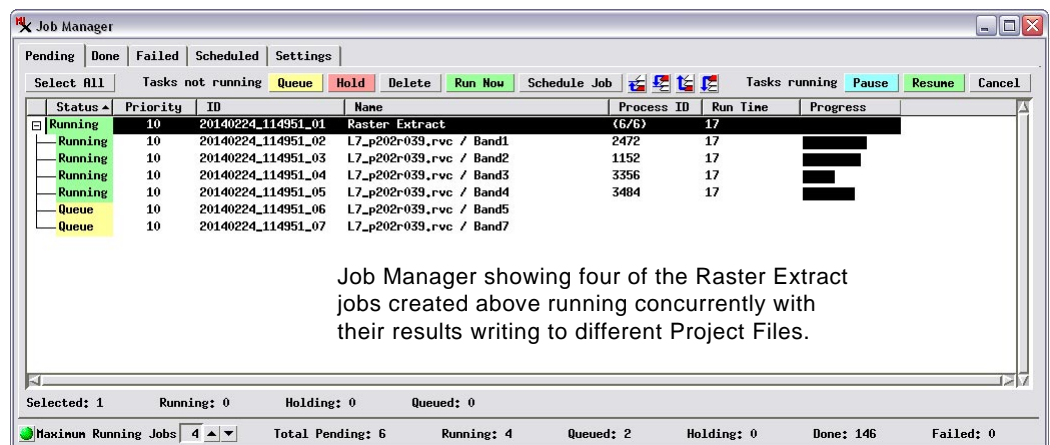
Job Manager showing four of the Raster Extract jobs created above running concurrently with their results writing to different Project Files.