Scheduling Individual and Repeating Jobs

The Job Processing System in TNTmips Pro allows you to run multiple simultaneous processes off-line, in the background. This distributes the processing load across your computer's multiple cores (see the collection of Technical Guides on job processing with TNTmips Pro at http://www.microimages.com/documentation/JobProcessing.htm). You can also schedule individual jobs to run at particular times and create schedules for jobs to automatically repeat at specific times and intervals.

Scheduling repeat jobs is useful when identical processing needs to be applied to input data that is continuously or periodically updated, but otherwise retains the same name and location. Examples might include exporting updated geospatial objects to external file formats for distribution, running a TNT geospatial script to download and process updated data from the Internet (see the TechGuide entitled *Sample Script: Building Dynamic Web Geomashups*), or updating your TNT geospatial data catalogs.

> The Job Manager in TNTmips Pro also allows you to set up scheduling for your TNT Job Processing System as a whole. For example, you can restrict the running of all off-line jobs to overnight hours or weekends. System-wide scheduling overrides any conflicting schedules for individual jobs. You set up system-wide scheduling using controls on the Job Manager's Settings tabbed panel (see the Technical Guide entitled *System: Managing Job Processing*).

Job Scheduling

Job scheduling for automatically updating your geocatalogs can be set up directly through the Geospatial Catalog Manager (see the TechGuide entitled *System: Automatic Update of Geocatalogs*). For other TNTmips processes and SML scripts, use the Job Manager to schedule one-time jobs and automatically repeating jobs. To start this procedure, make the input and output data selections and set the processing parameters in the desired process, then use the Save Job button to create a job that is held for later execution rather than being immediately added to the queue for processing (see the reverse for a discussion of creating a held job for an SML script). This job appears on the Job Manager's Pending tabbed panel with its status set to *Holding*. Select the job in the Pending list and press the Schedule Job button to open the Schedule window.


To create a repeating schedule for an automated job for most TNT processes or an SML script, save the job, select it in the list on the Job Manager's Pending tabbed panel, and press the Schedule Job button.

The Schedule window includes numeric fields in which you can specify the starting time and starting date for your job and to set it to be repeated if desired. The choices in the rest of the window change depending upon your selection on the *Perform this job* menu, which provides Daily, Weekly, and Monthly repeat options (see illustrations below).

## Daily and Hourly/Subhourly Jobs

The Daily option lets you schedule a job to repeat every day, on weekdays or weekend days, or at an interval of more than one day. Turn on the *Repeat every* toggle button to set the job to repeat at hourly or shorter intervals. Be sure to also set the time span over which those repeats occur (from 15 minutes to 1 day, or indefinitely).
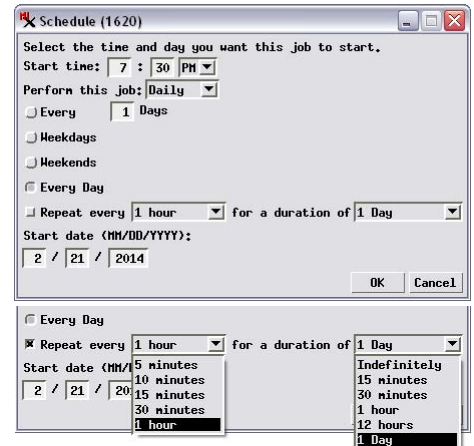
Schedule window with choices for daily and more frequent repeat intervals.

Weekly and Monthly Jobs

The Weekly option allows you to set a repeat interval of one or more weeks and to specify the day of the week on which to run the job. When you choose the Monthly option, you can set the job to run on a particular day in the month (by date or by week number and day of week) and specify the job to only repeat in specific months.
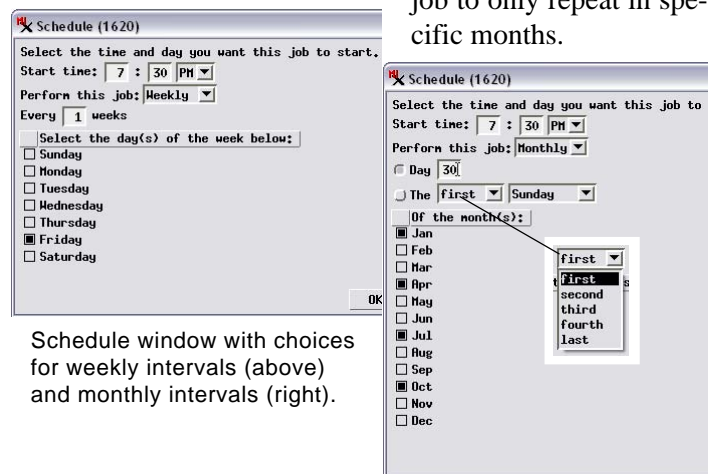
Schedule window with choices for weekly intervals (above) and monthly intervals (right).

Job File for a Scheduled Job

Once you have set the job schedule and pressed OK on the Schedule window, the file extension for this job file is set to *.tntjobschedule. The file is also moved

(over)

from the PendingJobs subdirectory to the ScheduleJobs subdirectory in your TNT Job folder. The Job Manager uses the scheduling parameters stored in this special job file to run the job with the specified repeat interval and duration.

Scheduled jobs are listed on the Scheduled tabbed panel of the Job Manager. Buttons are provided to edit the job schedule, disable and enable the job, and to delete a scheduled job.

Jobs you have scheduled are listed on the Job Manager's Scheduled tabbed panel. You can select a particular job in this list and use the provided buttons to temporarily disable and then renable it, to edit the schedule, or permanently delete the scheduled job.

Scheduled Jobs with Geospatial Scripts

Running a custom TNT geospatial (SML) script under job processing in general requires a pair of scripts. A process or P script does the actual processing. An Input/Output or IO script provides interactive selection of input and output data and makes a job file that identifies the process script to be run (as described in the TechGuide entitled *System: Custom Job Processing with Geospatial Scripts*). For a scheduled script job, all inputs and outputs must be specified explicitly in the process script itself. The role of the IO script then is simply to make a held job file that identifies the correct process script. The Job Manager procedures described on the front side of this page are then used to set up the scheduling for this job. A sample script used to make such a job file is shown to the right. The associated process script and its use are described in the TechGuides entitled *Sample Script: Building Dynamic Web Geomashups* and *Geomedia Publishing: Today's Earthquakes in California and Nevada*.

quakesmakejob.sml

Utility script to make a held job file to run canvquakes.sml. The TNT Job Manager can then be used to set a schedule for running the earthquake script.

```
class MIJOB job;
class STRING description$ =
        "Daily earthquake epicenters for California & Nevada";
class STRING scriptpath$ = _context.ScriptDir + "/canvquakes.sml";
class FILEPATH scriptpath(scriptpath$);

job.CreateJob(description$, scriptpath, 2);
job.Write(1);
```

parameter value = 1 sets job status to Holding