

Custom Job Processing with Geospatial Scripts

You can combine geospatial scripting and TNTmips Job Processing to efficiently perform specialized batch processing that cannot be accomplished by a standard TNT process. To do so you set up a pair of custom scripts: an Input/Output or IO script and a Process or P script. The interactive IO script can be run once to choose the geospatial inputs (i.e. layers), set job-specific processing variables, and automatically make an individual job file for each input in the batch run. The P script, which is specified in each job file, obtains its processing parameters from the job file and does the actual processing when the job is executed by the Job Manager.

The TNT geospatial scripting language (SML) includes a class (MIJOB) with simple methods for automatically creating a properly formatted job file and adding variable definitions and values to it. Job files are automatically written to the TNT job directory you have set for your installation.

This custom job processing example illustrates an IO and P script pair (excerpted on the reverse) that perform batch conversion of a large collection of four-band TIFF files to JP2 files with the same bands. The IO script sets up the repeated conversions by creating for each input TIFF file a individual job file with the selected JPEG2000 compression options.

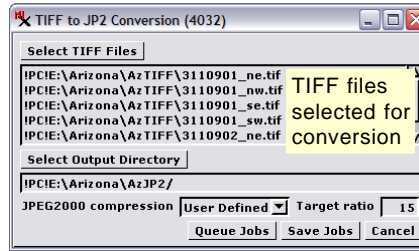
Script Variables Written to the Job File for Each Tiff to JP2 conversion Job

inputPath\$: string with the filepath of the current input TIFF file
 outputDir\$: string with the filepath of the directory for output
 compType\$: type of JPEG2000 compression to set
 compRatio: target compression ratio for lossy compression

```
<?xml version="1.0"?>
<job id="20090115_113435_00">
  <desc>Convert m_3110901_ne_12_1_20070625.tif to GeoJP2</desc>
  <process>tntdisp smljob</process>
  <version>75</version>
  <priority>2</priority>
  <runparms>
    <script>F:\SML\TIFFtoJP2\TiffToJP2fromJob.sml</script>
    <variable name="inputPath">
      <value>!PC!F:\Arizona\AzTIFF\m_3110901_ne_12_1_20070625.tif</value>
    </variable>
    <variable name="outputDir">
      <value>!PC!F:\Arizona\AzJP2</value>
    </variable>
    <variable name="compType">
      <value>user</value>
    </variable>
    <variable name="compRatio">
      <value>15.000000</value>
    </variable>
  </runparms>
</job>
```

Job File Written for One of the Input TIFF Files
 Job files are automatically written to the TNT jobs directory you have set for your TNT installation.

The IO script **TiffToJP2getJobParams.sml** in this example creates and opens a dialog for the user to select the TIFF files to convert to JP2, select the output directory, and set JPEG2000 compression options.



When the Queue Jobs or Save Jobs button is pressed, the script calls its MakeJobs procedure (shown in pink box below) to create a TNT job file for each input TIFF file (sample XML job file shown in green box below left).

Procedure to make job files in script TiffToJP2getJobParams.sml (further excerpts on reverse of this page)

```
proc MakeJobs(numeric hold)
{
  get compression options from the dialog
  compType$ = dlgwin.GetCtrlValueStr("jp2compOptions");
  if (compType$ == "user") then
    get target ratio set for user-defined compression
    compRatio = dlgwin.GetCtrlValueNum("compRatioNum");
  else
    compRatio = 0;

  path for the SML script to be called by the job files
  ScriptPath$ = _context.ScriptDir + "/TiffToJP2fromJob.sml";
  local class FILEPATH scriptPath(ScriptPath$);

  for i = 1 to numfiles
  {
    loop through input TIFF files to make job file for each
    local class MIJOB job;
    inputPath$ = filepathlist[i-1];
    inFilePath = inputPath$;
    class for setting up and writing a job file

    string with the job description to be shown in the Job Manager
    description$ = sprintf("Convert %s to GeoJP2", inFilePath.GetName());

    create job file structure with job description and path to the SML script to be called
    job.CreateJob(description$, scriptPath, 2);

    add variable names and values to the job file structure
    job.AddValue("inputPath$", inputPath$);
    job.AddValue("outputDir$", outputDir$);
    job.AddValue("compType$", compType$);
    job.AddValue("compRatio", compRatio);

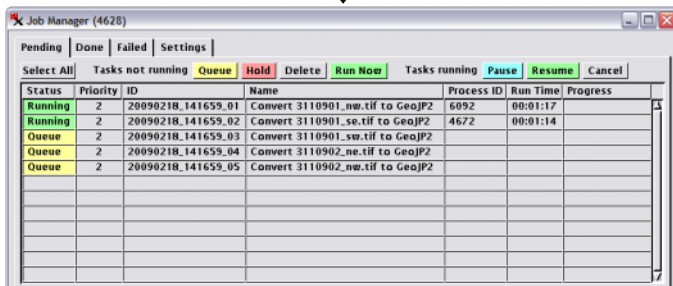
    write completed job file to the established TNT Jobs directory
    job.Write(hold);

    OnCancel();
  }
  call procedure to close dialog and exit the script
}
```

Each job file calls the script **TiffToJP2fromJob.sml**. This P script includes declarations for the processing variables defined in the job file:

```
string inputPath$;      string outputDir$;
string compType$;      numeric compRatio
```

Values for these variables are automatically read and assigned from the job file.



Many sample scripts have been prepared to illustrate how you might use the features of the TNT products' scripting language for scripts and queries. These scripts can be downloaded from www.microimages.com/downloads/scripts.htm.

Excerpts from TiffToJP2getJobParms.sml

Provides user interface for TIFF to JP2 file conversion.
Makes job files that call TiffToJP2fromJob.sml.

class STRINGLIST filepathlist; list of filepaths of the TIFF files selected

Procedure called by the Select TIFF Files button on the dialog. Gets a stringlist with the filepath strings for the selected files, writes the filepath strings to the file listbox on the dialog, and enables the Get Directory button

```
proc GetInputFiles()
{
  local class GUI_CTRL_LISTBOX filelistbox; handle for file listbox on the dialog
  local numeric i;
  filelistbox = dlgwin.GetCtrlByID("filelistbox");
  numfiles = GetInputFileNames(_context.ScriptDir,
    "Select TIFF file(s) for conversion to JP2:", "TIF", filepathlist);
  printf("numfiles = %d\n", numfiles);

  for i = 1 to numfiles add filepath string to the listbox
  {
    filelistbox.AddItem( filepathlist[i-1] );
  }
  dlgwin.GetCtrlByID("dirBtn").SetEnabled(1); enable the button to select output directory
}
```

Procedure called by the Select Output Directory button on the dialog. Gets the output directory path and enables the Run Jobs and Save Jobs buttons.

```
proc GetOutputDirectory()
{
  outputDir$ = GetDirectory("", "Choose directory for JP2 files:");
  outputDir$ = outputDir$ + "/";
  dlgwin.SetCtrlValueStr("outDirText", outputDir$); write directory name to the dialog

  dlgwin.GetCtrlByID("runJobsBtn").SetEnabled(1);
  dlgwin.GetCtrlByID("saveJobsBtn").SetEnabled(1); enable the Queue Jobs and Save Jobs buttons
}
```

Procedure called when JPEG2000 compression option is selected. Changes the state of associated controls on the dialog.

```
proc OnCompSelected()
{
  if (dlgwin.GetCtrlByID("jp2compOptions").GetValueStr() == "user")
  {
    if (ratioEnabled <> 1) if ratio label and field are currently disabled, enable them
    {
      dlgwin.GetCtrlByID("ratioLabel").SetEnabled(1);
      dlgwin.GetCtrlByID("compRatioNum").SetEnabled(1);
      ratioEnabled = 1;
    }
  }
  else compression option = lossless or best quality
  {
    if (ratioEnabled == 1) if ratio label and field are enabled, disable them
    {
      dlgwin.GetCtrlByID("ratioLabel").SetEnabled(0);
      dlgwin.GetCtrlByID("compRatioNum").SetEnabled(0);
      ratioEnabled = 0;
    }
  }
}
```

[Procedure to make job files is shown on the reverse of this page.]

[Code to create and open the dialog window is omitted.]

Excerpts from TiffToJP2fromJob.sml

TIFF to JP2 conversion script called by job file.

string inputPath\$; string outputDir\$; variables read from the job file
string compType\$; numeric compRatio;

class STRING logfileName\$; variables for a log file to record processing information
class FILE logfile;
class DATETIME currentDT; the current local date/time

open log file in the same directory as this script to append status information; if no log file is present, a new one will be created

```
logfileName$ = _context.ScriptDir + "TiffToJP2.log";
logfile = fopen(logfileName$, "a");
```

```
currentDT.SetCurrent(); get current date and time and write to log
fprintf(logfile, "\nJob initiated %s Central Standard Time\n", currentDT);
```

set up filepath for the input TIFF file using the path string from the job file

```
class FILEPATH inFilepath(inputPath$);
fprintf(logfile, "Input filename = %s\n", inFilepath);
```

get the TIFF filename (minus extension) to use for naming the output JP2 file

```
class STRING filename$ = inFilepath.GetNameOnly();
```

```
class FILEPATH outFilepath(outputDir$); set up filepath for the output JP2 file in the directory designated in the job file
outFilepath.Append(filename$);
outFilepath.SetExtension("jp2");
fprintf(logfile, "output filepath = %s\n", outFilepath);
```

print other processing parameters read from the job file

```
fprintf(logfile, "compType$ = %s\n", compType$);
fprintf(logfile, "compRatio = %.1f\n", compRatio);
```

PIPELINE SOURCE: TIFF file

```
class IMAGE_PIPELINE_SOURCE_TIFF sourceTIFF(inFilepath);
err = sourceTIFF.Initialize();
if (err < 0 ) ReportError(_context.CurrentLineNum, err);
else { fprintf(logfile, "Source TIFF file initialized"); }
```

PIPELINE TARGET: J2K File with JPEG2000 Compression

```
class IMAGE_PIPELINE_TARGET_J2K_SETTINGS settings;
if (compType$ == "lossless") then
  settings.SetReversible(1);
else
  settings.SetTargetRatio(compRatio);
```

```
class IMAGE_PIPELINE_TARGET_J2K targetJ2K(sourceTIFF,
  outFilepath, settings);
err = targetJ2K.Initialize();
if (err < 0 ) ReportError(_context.CurrentLineNum, err);
else { fprintf(logfile, "Target J2K file initialized"); }
```

```
err = targetJ2K.Process(); PROCESS THE PIPELINE
if (err < 0 ) ReportError(_context.CurrentLineNum, err);
else { fprintf(logfile, "Target processed"); }
```

```
fclose(logfile);
```