

# 地理空間スクリプトによるカスタムジョブ処理

地理空間スクリプトと TNTmips のジョブ処理を組み合わせ、標準の TNT 処理ではできないユーザ独自のバッチ処理を効率的に行うことができます。それを行うには、入出力に関わる I0 スクリプトとプロセスに関わる P スクリプトの 1 組のカスタムスクリプトをセットアップします。対話的な I0 スクリプトを一度実行し、地理空間入力情報（すなわちレイヤー）を選び、ジョブに固有なプロセス変数を設定し、各入力に対応してバッチ実行を行うための個別のジョブファイルを作成します。各ジョブファイルで定義される P スクリプトはジョブファイルから処理パラメータを入手し、そのジョブがジョブマネージャにより実行されて実際の処理が行われます。

TNT の地理空間スクリプト言語 (SML) には、自動的に適切なフォーマットのジョブファイルを作成し、変数定義と値をそれに加える簡単な方法を備えたクラス (MIJOB) が含まれています。ジョブファイルはユーザがインストール時に設定した TNT ジョブディレクトリに自動的に書き出されます。

このカスタムジョブ処理の例として、多数の 4 バンドの TIFF ファイルの一群から同一バンドの JP2 ファイルへのバッチ変換を実行する I0 スクリプトと P スクリプトのペア（次ページに抜粋あり）を説明しています。I0 スクリプトは繰り返しの変換をセットアップします。各入力 TIFF ファイルに対して、指定された JPEG2000 圧縮オプションを持った個別のジョブファイルを生成します。

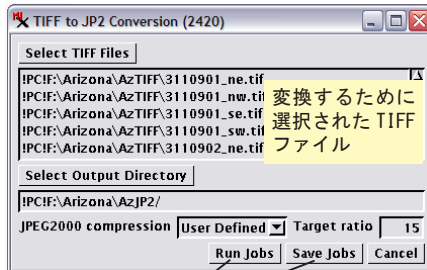
## TIFF から JP2 に変換するジョブファイルに書かれるスクリプト変数

inputPath\$: 現在の入力 TIFF ファイルのファイルパス  
 outputDir\$: 出力ディレクトリのファイルパス  
 compType\$: セットする JPEG2000 圧縮のタイプ  
 compRatio: ターゲットの非可逆（損失あり）圧縮率

```
<?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>!PCIF:\Arizona\AzTIFF\m_3110901_ne_12_1_20070625.tif</value>
    </variable>
    <variable name="outputDir">
      <value>!PCIF:\Arizona\AzJP2</value>
    </variable>
    <variable name="compType">
      <value>user</value>
    </variable>
    <variable name="compRatio">
      <value>15.00000</value>
    </variable>
  </runparms>
</job>
```

入力 TIFF ファイルの一つに対して書かれたジョブファイル  
 ジョブファイルはユーザがインストール時に設定した TNT ジョブディレクトリに自動的に書き出されます。

この例の I0 スクリプト "TiffToJP2getJobParms.sml" では、ユーザが JP2 にコンバートする TIFF ファイルと出力ディレクトリを指定し、JPEG2000 圧縮率を設定するダイアログが開かれます。[ジョブの実行 (Run Jobs)] または [ジョブの保存 (Save Jobs)] ボタンが押されると、



スクリプトはジョブ作成 (MakeJobs) プロシージャ（下のピンク枠内）を呼び、各入力 TIFF ファイルに対する TNT ジョブファイル（左下、緑枠内に XML のサンプルがあります）を生成します。

TiffToJP2getJobParms.sml スクリプトの中のジョブファイルを作るプロシージャ（抜粋）  
 （次ページに続く）

```
proc MakeJobs(numeric hold)
{
  ダイアログから圧縮オプションを取り込みます
  compType$ = dlgwin.GetCtrlValueStr("jp2compOptions");
  if (compType$ == "user") then
    ユーザ定義のターゲット圧縮率の設定を取り込みます
    compRatio = dlgwin.GetCtrlValueNum("compRatioNum");
  else
    compRatio = 0;
  ジョブファイルによってコールされる SML スクリプトのパス
  ScriptPath$ = _context.ScriptDir + "/TiffToJP2fromJob.sml";
  local class FILEPATH scriptPath(ScriptPath$);

  for i = 1 to numfiles
  {
    それぞれのジョブファイルを作るため入力 TIFF ファイル分繰り返します
    local class MIJOB job;
    inputPath$ = filepathlist[i-1];
    inFilePath = inputPath$;
    ジョブファイルのセットアップおよび書き出しのためのクラス
    ジョブマネージャで表示されるジョブディスクリプション（文字列）
    description$ = sprintf("Convert %s to GeoJP2", inFilePath.GetName());
    ジョブディスクリプションとコールされる SML スクリプトのパスを持つジョブファイル構造体を作ります
    job.CreateJob(description$, scriptPath, 2);
    変数名と値をジョブファイル構造体に加えます
    job.AddValue("inputPath$", inputPath$);
    job.AddValue("outputDir$", outputDir$);
    job.AddValue("compType$", compType$);
    job.AddValue("compRatio", compRatio);
    完成したジョブファイルを指定された TNT ジョブディレクトリに書き出します
    job.Write(hold);
  }
  OnCancel();
  ダイアログを閉じスクリプトを終わるプロシージャを呼びます
}
```

Status	Priority	ID	Name	Process ID	Run Time	Progress
Pending	2	20090218_141659_01	Convert 3110901_nw.tif to GeoJP2	6892	00:01:17	
Running	2	20090218_141659_02	Convert 3110901_se.tif to GeoJP2	4672	00:01:14	
Queue	2	20090218_141659_03	Convert 3110901_sw.tif to GeoJP2			
Queue	2	20090218_141659_04	Convert 3110902_ne.tif to GeoJP2			
Queue	2	20090218_141659_05	Convert 3110902_nw.tif to GeoJP2			

各ジョブファイルはスクリプト "TiffToJP2fromJob.sml" をコールします。この P スクリプトにはジョブファイルの中で定義された処理変数の宣言が含まれます。

ジョブファイルで定義される処理変数の宣言：  
 string inputPath\$; string outputDir\$;  
 string compType\$; numeric compRatio  
 これらの変数の値はジョブファイルから自動的に読み込まれて割り当てられます。

www.microimages.com/downloads/scripts.htm にはダウンロード可能な多くのサンプルスクリプトがあり、スクリプトやクエリーで TNT 製品のスクリプト言語の特徴をどのように利用したらよいかを解説しています。

### "TiffToJP2getJobParms.sml" からの抜粋 (I/O スクリプト)

TIFF から JP2 ファイルにファイル変換するインタフェースを提供します。  
TiffToJP2fromJob.sml を呼ぶジョブファイルを作ります。

```
class STRINGLIST filepathlist;
```

選択された TIFF ファイルのファイルパスリスト

プロシージャは、ダイアログの [TIFF ファイル選択 (Select TIFF Files)] ボタンで呼び出されます。選択されたファイルのファイルパスの文字列リストを取り込み、ダイアログボックス上のファイル一覧 (リストボックス) にファイルパス (文字列) を書き出し、[ディレクトリ指定 (Get Directory)] ボタンを使用可能にします。

```
proc GetInputFiles()
```

```
{
  local class GUI_CTRL_LISTBOX filelistbox;
  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);
```

ダイアログのファイルリストボックスへのハンドル

```
for i = 1 to numfiles
```

```
{
  filelistbox.AddItem( filepathlist[i-1] );
}
dlgwin.GetCtrlByID("dirBtn").SetEnabled(1);
}
```

リストボックスにファイルパス文字列を加えます

出力ディレクトリ選択ボタンを使用可能にします

ダイアログの [出力ディレクトリ選択 (Select Output Directory)] ボタンによりコールされるプロシージャ。出力ディレクトリのパスを取り入れ、[ジョブの実行][ジョブの保存] ボタンを使用可能にします。

```
proc GetOutputDirectory()
```

```
{
  outputDir$ = GetDirectory("", "Choose directory for JP2 files:");
  outputDir$ = outputDir$ + "/";
  dlgwin.SetCtrlValueStr("outDirText", outputDir$);
```

ディレクトリ名をダイアログに書き出します

```
dlgwin.GetCtrlByID("runJobsBtn").SetEnabled(1);
dlgwin.GetCtrlByID("saveJobsBtn").SetEnabled(1);
}
```

[ジョブの実行][ジョブの保存] ボタンを使用可能にします

JPEG2000 圧縮オプションが選択されるとプロシージャが呼ばれます。ダイアログ上で関連した状態表示が変わります。

```
proc OnCompSelected()
```

```
{
  if (dlgwin.GetCtrlByID("jp2compOptions").GetValueStr() == "user")
  {
    if (ratioEnabled <> 1)
    {
      dlgwin.GetCtrlByID("ratioLabel").SetEnabled(1);
      dlgwin.GetCtrlByID("compRatioNum").SetEnabled(1);
      ratioEnabled = 1;
    }
  }
  else
  {
    if (ratioEnabled == 1)
    {
      dlgwin.GetCtrlByID("ratioLabel").SetEnabled(0);
      dlgwin.GetCtrlByID("compRatioNum").SetEnabled(0);
      ratioEnabled = 0;
    }
  }
}
```

圧縮率のラベルやフィールドが使用できなければ使用可能にします

圧縮率オプション=ロス無し、または最高画質

圧縮率のラベルとフィールドが使用可能であれば、使用不可能にします

[ジョブファイルを作るプロシージャはこの前のページにあります。]  
[ダイアログウィンドウを作り開くコードは省略されています。]

### "TiffToJP2fromJob.sml" からの抜粋 (P スクリプト)

ジョブファイルから呼ばれる TIFF を JP2 ファイルにファイル変換するスクリプト。

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

ジョブファイルより読み込まれる変数

```
class STRING logfileName$;
```

処理情報を記録するログファイル用の変数

```
class FILE logfile;
```

```
class DATETIME currentDT;
```

現在のローカルの日付 / 時刻

このスクリプトと同じディレクトリにあるログファイルを開き、ステータス情報を追加します。ログファイルが存在しない場合は新規に作られます

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

現在の日付と時刻を得てログに書き出します

```
currentDT.SetCurrent();
```

```
fprintf(logfile, "\nJob initiated %s Central Standard Time\n", currentDT);
```

ジョブファイルのパスの文字列を使って入力 TIFF ファイル用のファイルパスをセットアップします

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

TIFF ファイル名から拡張子をとって、出力 JP2 ファイルの名前付けに使用します

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

ジョブファイルに指示されたディレクトリに JP2 ファイルを出力するためのパスをセットアップします

```
class FILEPATH outFilepath(outputDir$);
outFilepath.Append(filename$);
outFilepath.SetExtension("jp2");
fprintf(logfile, "output filepath = %s\n", outFilepath);
```

他の処理パラメータをジョブファイルから読み込み、ログファイルにプリントします

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

ソース (TIFF file) のパイプライン

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

ターゲット (JPEG2000 圧縮の J2K ファイル) のパイプライン

```
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"); }
```

パイプラインの処理

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

```
fclose(logfile);
```