

## Sample SML Tool Script

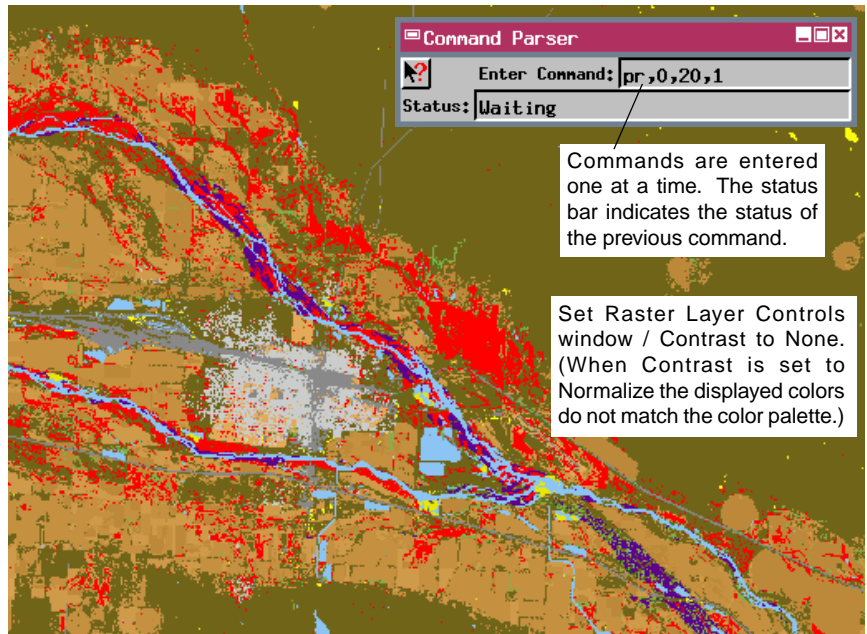
# Printing Fixed Colors

The `compar.sml` script provides an example that can be used and modified to assign colors to cell values in an 8-bit raster in a quantitative fashion. If your printer is color stable, you can determine which numerically input RGB color value produces the exact color required. This script will then let you specify a cell value in your input raster and assign it a precise RGB color in the associated color palette. The script was written for someone who wanted a fast way to assign a specific color to each class in a classified image. Assuming your color printer does not drift in color, this approach will ensure that each class is printed in a consistent color from raster to raster. This script also illustrates how user input is parsed.

When you select this tool, a Command Parser window opens for you to enter commands.

The paint commands require you to specify a color with a color number. The script looks up the color number in a specified text file, which contains sequential color numbers (called index numbers) from 0 to 255. Each index number has red, green, and blue color component percentages as well as a transparency value. You can create this file from any raster's color palette by using the `t` command. Edit this file or even create it from scratch using a text editor or spreadsheet program. Depending on the command you enter, the script can copy these colors all at once to a raster's color palette (`lc` command) or load them (`b` command) if you want to access colors individually in order to paint specific raster cells (`p` and `pr` commands).

The best way to find the index number of a color (or a class represented by that color) is to make a standard color chart you can refer to. A color chart makes it easy to use any number of color shades to represent unique classes. All you have to do is refer to the chart, find the index number of the color you need, and use the `compar.sml` tool to paint all corresponding raster cells. The color chart is even more important when you have a large number of features to classify and you have no



choice but to use similar colors for different feature classes. Even with a few classes, subtle but noticeable changes in brightness and shade are an attractive and effective way to convey information about one class relative to another. There are many ways to create a color chart; the simplest is to list all of your color numbers along with the feature classes they correspond to. The chart to the left was created in TNTmips.

Before painting a raster, enter the `b` command to load the text file that contains the index numbers and color component percentages for all your colors. Use the `p` and `pr` commands to paint the raster. All you need to know to use these commands is the color number corresponding to the class, which you can get from your color chart, and the corresponding cell values that you want to paint.

### Commands with sample variables:

- `t` (outputs color palette to text file)
- `lc` (make a new SMLcolor palette from text file)
- `b` (load text file colors to use to paint)
- `r,0,255` (cell value 0 to 255 becomes transparent)
- `pr,0,20,1` (paints range of cell values 0 to 20 with color # 1)
- `p,21,2` (paint cell value 21 with color # 2)

Group 1 - Group View 1			
View Tool LegendView GPS Options			
Woody Herbaceous Wetlands	0	Emergent Residential Wetlands	1
Herbaceous planted/cultivated urban/recreational Grasses	5	Herbaceous Fallow	6
Shrubland	10	Orchards/ Vineyard	11
Deciduous Forest	15	Evergreen Forest	16
Barren Bare Rock	20	Barren Quarries	21
		Barren Transitional	22
		Low Intensity Residential	2
		High Int Reside	3
		Herbaceous Small Grains	7
		Herbaceous Pasture Hay	8
		Grasslands	12
		Water	13

CLASS: Emergent  
Herbaceous Wetlands  
Color # (index): 1  
Red: 100%  
Green: 0%  
Blue: 0%  
Transparent: 0%

Sample scripts have been prepared to illustrate how you might use the features of TNTmips' Spatial Manipulation Language (SML). Key sections of the script are printed below for your quick perusal. The entire script can be downloaded from the SML script exchange at [www.microimages.com/sml/ftpsmlink/TNT\\_Products\\_V6.5\\_CD](http://www.microimages.com/sml/ftpsmlink/TNT_Products_V6.5_CD).

## Excerpts from Command Parser Tool Script (compar.sml)

```

proc OutputColorMap() {
  if (Group.ActiveLayer.Type == "Raster") {
    DispGetRasterFromLayer(rast,Group.ActiveLayer);
    cmap = ColorMapFromRastVar(rast);
    if (cmap.name == "") {
      PopupMessage("No ColorMap was found in raster object Paint function
      aborted\n Use lc command to save a csv file as a colormap in the raster
      object first\n or create one yourself");
      Command.value = "";
      Status.value = "OutputColorMap aborted";
      return;
    }
    myfile = GetOutputTextFile("c:/color.csv","Select file for output:","csv");
    fprintf(myfile,"%s,%s,%s,%s,%s\n","Index","Red","Green","Blue",
      "Transparency");
    Status.value = "Outputting Colormap";
    for i = 0 to 255 {
      mycolor = ColorMapGetColor(cmap,i);
      fprintf(myfile,"%d,%d,%d,%d,%d\n",i,round(mycolor.red),
        round(mycolor.green),round(mycolor.blue),
        round(mycolor.transp));
    }
    fclose(myfile);
    Status.value = "Colormap outputted";
    Command.value = "";
  }
  else {
    Status.value = "Active Layer must be a raster w/colormap for this function";
    Command.value = "";
  }
}

proc Paint() {
  if (NumberTokens(Command.value,delim) != 3) {
    Status.value = "Not enough parameters for Paint function";
    return;
  }
  if (!hascolorarray) {
    PopupMessage("Use code b (SetColorArray function) to load a colormap
    to use first");
    Status.value = "Paint function aborted";
    Command.value = "";
    return;
  }
  local numeric cellvalue;
  local numeric colomumber;
  cellvalue = StrToNum(GetToken(Command.value,delim,2));
  colomumber = StrToNum(GetToken(Command.value,delim,3));
  if (Group.ActiveLayer.Type != "Raster") {
    PopupMessage("Active Layer must be a raster object for this function");
    Command.value = "";
    Status.value = "Active layer must be a raster";
    return;
  }
  DispGetRasterFromLayer(rast,Group.ActiveLayer);
  cmap = ColorMapFromRastVar(rast);
  if (cmap.name == "") {
    PopupMessage("No ColorMap was found in raster object Paint function
    aborted\n Use lc command to save a csv file as a colormap in the raster
    object first\n or create one yourself");
    Command.value = "";
    Status.value = "Paint function aborted aborted";
    return;
  }
  mycolor.red = ared[colomumber+1];
  mycolor.green = agreen[colomumber+1];
  mycolor.blue = ablue[colomumber+1];
  mycolor.transp = atransp[colomumber+1];
  ColorMapSetColor(cmap,cellvalue,mycolor);
}

```

OutputColorMap called when t command entered; copies color map in Raster to CSV file  
Command Syntax: t

Opens Select File window for you choose the output file name

Returns the color at index 'i' in color palette 'cmap'

fprintf output the CSV file, example:

Index	Red	Green	Blue	Transparency
0	0	0	0	50
1	0	100	0	50
2	0	100	100	50
.	.	.	.	.
.	.	.	.	.
255	100	100	100	100

Paint procedure invoked when p command entered; sets 'cellvalue' in color palette to 'index' value in array  
Command Syntax: p,cellvalue,index

'hascolorarray' is initially set to false; when b command (SetColorArray function) is executed it is set to true

Gets Raster 'rast' from active layer

Returns Raster's color palette

array variables (ared, agreen, ablue, & atransp) hold paint values; 'colomumber' can be 0-255; you must add one since the array holds colors 1-256 (SML arrays start at 1)

Sets a color palette color  
ColorMapSetColor(colormap,index,color)

Writes a color palette under a raster  
ColorMapWriteToRastVar(Raster,colormap.Name\$,description\$)

ColorMapWriteToRastVar(rast,cmap,cmap.Name,cmap.Desc);

```

View.DisableRedraw = 1;
LayerDestroy(Group.ActiveLayer);
GroupQuickAddRasterVar(Group,rast);
Group.ActiveLayer.AllowDeleteLayer = 1;
View.DisableRedraw = 0;
ViewRedraw(View);

```

\*These six lines of code force raster to reload with edited color palette

Status.value = "Cell Painted";  
Command.value = "";

LoadColorMap procedure invoked when lc command is entered; creates a new color map for the Raster  
Command Syntax: lc

```

proc LoadColorMap() {
  local string line;
  myfile = GetInputTextFile("c:/colormap.csv","Select Colormap file","csv");
  Checks to see if the input file is valid:
  line = fgetline(myfile);
  if ((NumberTokens(line,",") != 5) || ("Index" != GetToken(line,",",1))) {
    PopupMessage("This does not appear to be a valid colormap csv file\n
    The proper format is one row of labels then 256 numeric lines of the
    form red,green,blue,transp\n where red,green,blue,transp are in the range
    0-100\n Display a raster with a colormap already and use the t command to
    export the colormap to a file to see how the file should look");
    Status.value = "Fatal Error Colormap load halted";
    Command.value = "";
    return;
  }
  Status.value = "Loading Colormap";
  for i = 0 to 255 {
    line = fgetline(myfile);
    Additional checks to see if the input file is valid:
    if ((i > 0) && (StrToNum(GetToken(line,",",1)) = 0)) {
      PopupMessage("Bad Index Value Encountered while loading Colormap file
      Colormap load aborted");
      Status.value = "Fatal Error Colormap load halted";
      Command.value = "";
      return;
    }
    mycolor.red = StrToNum(GetToken(line,",",2));
    mycolor.green = StrToNum(GetToken(line,",",3));
    mycolor.blue = StrToNum(GetToken(line,",",4));
    mycolor.transp = StrToNum(GetToken(line,",",5));
    ColorMapSetColor(cmap,i,mycolor);
  }
  if (Group.ActiveLayer.Type != "Raster") {
    PopupMessage("Active Layer must be a raster object for this function");
    Command.value = "";
    Status.value = "Active layer must be a raster";
    return;
  }
  DispGetRasterFromLayer(rast,Group.ActiveLayer);
  ColorMapWriteToRastVar(rast,cmap,"SMLcolor","
  ColorMap created by SML script");
  Status.value = "ColorMap saved as SMLcolor in raster object";
  PopupMessage("Colormap was saved as a SMLcolor colormap under raster
  object\n you must select this colormap to see the changes\n if you were displaying
  the raster with a different colormap\n this is the only function that behaves this
  way\n All other function modify the most recently used colormap");
  Command.value = "";
}

```

Opens Select File window for you choose the input file name

Get next line of text from the csv file

GetToken(string\$,delimiters\$,tokenNumber) gets a token, which is a delimited portion of a string

writes new color palette called SMLcolor for Raster;  
ColorMapWriteToRastVar(Raster,colormap,name\$,description\$)

code to force raster to reload with new color palette is not shown (\*see identical lines of code in Paint procedure)

global variables declared outside of these procedures  
class RASTER rast; class FILE myfile; array ablue[256];  
class ColorMap cmap; array ared[256]; array atransp[256];  
class Color mycolor; array agreen[256]; numeric hascolorarray;