

**NAME: Topographic Position Index (TPI) v. 1.2**

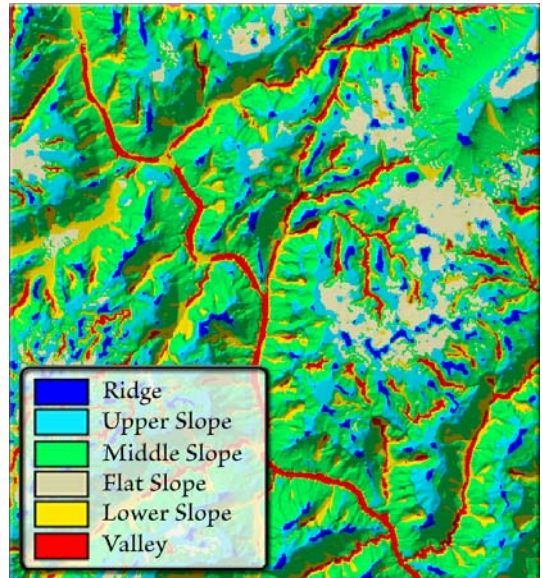
**Aka:** tpi\_jen.avx

**Last modified:** March 22, 2006

**TOPICS:** Topography Position Index TPI Landform Slope Classification Standardization Neighborhood Statistics

**AUTHOR:** Jeff Jenness  
Wildlife Biologist, GIS Analyst  
Jenness Enterprises  
3020 N. Schevene Blvd.  
Phone: 1-928-607-4638  
Flagstaff, AZ 86004 USA

Email: [jeffj@jennessent.com](mailto:jeffj@jennessent.com)  
Web Site: <http://www.jennessent.com>



**DESCRIPTION:** This extension calculates Topographic Position Index (TPI) grids from elevation grids, and provides a simple and repeatable method to classify the landscape into slope position and landform category using the TPI values.

This extension provides additional functions to standardize and classify grids, and to generate grid and neighborhood statistics.

**REQUIRES:** Spatial Analyst for ArcView 3.x.

Also requires that the file "avdlog.dll" be present in the ArcView/BIN32 directory (or \$AVBIN/avdlog.dll) and that the Dialog Designer extension be located in your ArcView/ext32 directory, which they usually are if you're running AV 3.1 or better. The Dialog Designer doesn't have to be loaded; it just has to be available. If you are running AV 3.0a, you can download the appropriate files for free from ESRI at:

<http://support.esri.com/index.cfm?fa=downloads.patchesServicePacks.viewPatch&PID=25&MetalD=483>

**REVISIONS:** Version 1.01 (July 12, 2005) corrects a bug producing an error message stating that "The script 'grid\_tools\_jen.RescaleRainbowIncrements'" could not be found.

Version 1.01a (July 17, 2005) makes minor changes to the extension and edits to the manual.

Version 1.2 (March 22, 2006) fixes an intermittent bug related to creating, saving and reusing custom landform classification sets, and which displayed an error stating that "Variable theTPI1Units2GridString has not been initialized."

**Recommended Citation Format:** For those who wish to cite this extension, the author recommends something similar to:

Jenness, J. 2006. Topographic Position Index (tpi\_jen.avx) extension for ArcView 3.x, v. 1.2. Jenness Enterprises. Available at: <http://www.jennessent.com/arcview/tpi.htm>.

Please let me know if you cite this extension in a publication ([jeffj@jennessent.com](mailto:jeffj@jennessent.com)). I will update the citation list to include any publications that I am told about.

**Acknowledgments:** The author gratefully acknowledges the work of Andrew Weiss who originally developed the TPI methodology and concepts. This extension automates Weiss' methods and allows users to save and share classification criteria sets.

The author also acknowledges and thanks Joyce Miller of the National Oceanic and Atmospheric Administration for suggesting this tool and for funding an early version of it.

## TABLE OF CONTENTS

<b>GENERAL INSTRUCTIONS:</b> .....	<b>4</b>
<b>TOPOGRAPHIC POSITION INDEX:</b> .....	<b>4</b>
DISCUSSION:.....	4
Scales and Neighborhoods: .....	4
Classifying by Slope Position: .....	6
Classifying by Landform Category: .....	8
<b>USING THIS EXTENSION:</b> .....	<b>11</b>
CALCULATING THE TOPOGRAPHIC POSITION INDEX: .....	11
CLASSIFYING LANDSCAPE INTO SLOPE POSITION: .....	12
Selecting and Defining a Classification Regime:.....	13
Criteria Label and Value:.....	14
TPI Criteria: .....	14
Slope Criteria:.....	15
Additional Grids:.....	15
Classification Criteria List: .....	16
Loading Criteria Sets:.....	16
Saving Criteria Sets:.....	17
Options:.....	17
CLASSIFYING LANDSCAPE INTO LANDFORM CATEGORY: .....	19
Selecting and Defining a Classification Regime:.....	20
Criteria Label and Value:.....	21
Small- and Large-Neighborhood TPI Criteria: .....	22
Additional Grids:.....	22
Classification Criteria List: .....	23
Loading Criteria Sets:.....	23
Saving Criteria Sets:.....	24
Options:.....	25
<b>ADDITIONAL FUNCTIONS:</b> .....	<b>28</b>
CALCULATING NEIGHBORHOOD STATISTICS .....	28
Statistic Options: .....	28
Neighborhood Options: .....	29
GENERAL GRID CLASSIFICATION .....	31
Classifying by Multiple Grids: .....	31
Classifying a Single Grid: .....	31
Selecting and Defining a Classification Regime:.....	32
Criteria Label and Value:.....	33
Classification Criteria List: .....	33
Loading Criteria Sets:.....	33
Saving Criteria Sets:.....	33
Options:.....	34
GENERAL GRID STATISTICS .....	35
STANDARDIZING GRIDS.....	36
<b>SAVED CRITERIA SETS .....</b>	<b>37</b>

DISCUSSION.....	37
SETTING DEFAULT TABLE LOCATIONS .....	37
SHARING CRITERIA SETS.....	38
TECHNICAL DATA: FIELDS AND VALUES .....	38
<b>TROUBLESHOOTING: .....</b>	<b>39</b>
<b>TECHNICAL NOTES: .....</b>	<b>41</b>
<b>REFERENCES: .....</b>	<b>41</b>

## GENERAL INSTRUCTIONS:

- 1) Begin by placing the "tpi\_jen.avx" file into the ArcView extensions directory (../Av\_gis30/Arcview/ext32/).
- 2) After starting ArcView, load the extension by clicking on **File --> Extensions...** , scrolling down through the list of available extensions, and then clicking on the checkbox next to the extension called "**Topographic Position Index.**"
- 3) This extension adds a new menu to your View menu bar called "TPI". Within this menu are options to create a TPI grid, a Slope Position grid and a Landform Classification grid, as well as general options for grid classification, grid standardization and calculating grid and neighborhood statistics.

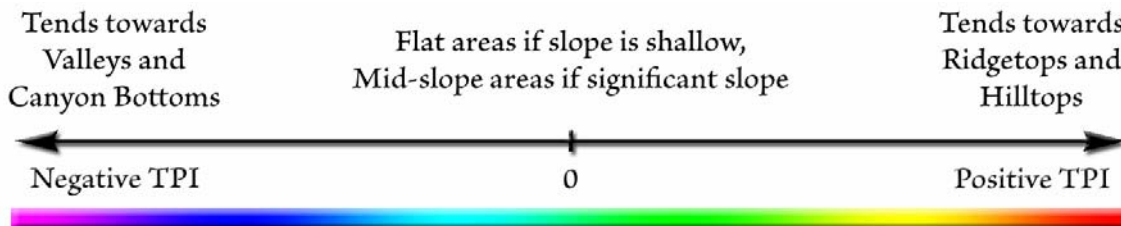
## TOPOGRAPHIC POSITION INDEX:

### DISCUSSION:

Andrew Weiss presented a very interesting and useful poster at the 2001 ESRI International User Conference describing the concept of Topographic Position Index (TPI) and how it could be calculated (Weiss 2001; see also Guisan et al. 1999 and Jones et al. 2000). Using this TPI at different scales, plus slope, users can classify the landscape into both slope position (i.e. ridge top, valley bottom, mid-slope, etc.) and landform category (i.e. steep narrow canyons, gentle valleys, plains, open slopes, mesas, etc.).

The algorithms are clever and fairly simple. The TPI is the basis of the classification system and is simply the difference between a cell elevation value and the average elevation of the neighborhood around that cell. Positive values mean the cell is higher than its surroundings while negative values mean it is lower.

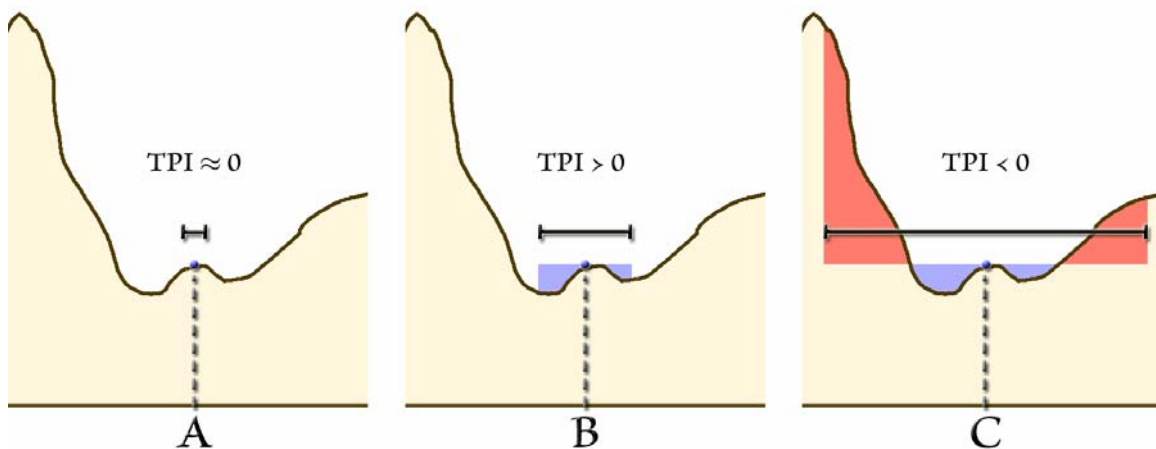
The degree to which it is higher or lower, plus the slope of the cell, can be used to classify the cell into slope position. If it is significantly higher than the surrounding neighborhood, then it is likely to be at or near the top of a hill or ridge. Significantly low values suggest the cell is at or near the bottom of a valley. TPI values near zero could mean either a flat area or a mid-slope area, so the cell slope can be used to distinguish the two.



**Scales and Neighborhoods:** TPI is naturally very scale-dependent. The same point at the crest of a mountain range might be considered a ridgetop to a highway construction crew or a flat plain to a mouse. The classifications produced by this extension depend entirely on the scale you use to analyze the landscape.

For example, in the illustration below, TPI is calculated for the same point on the landscape using 3 different scales. In each case, the point is located on top of a small hill set inside a larger valley. In Case A, the scale is small enough that the point is at about the same elevation as the entire analysis region so the TPI value would be approximately 0. In Case B, the analysis region is big enough to encompass the entire small hill, and the point is consequently much higher than its neighbors and has a correspondingly high TPI value. In Case C, the neighborhood includes the hills on either side of the valley, and therefore the point is lower than its neighbors and has a negative TPI value.

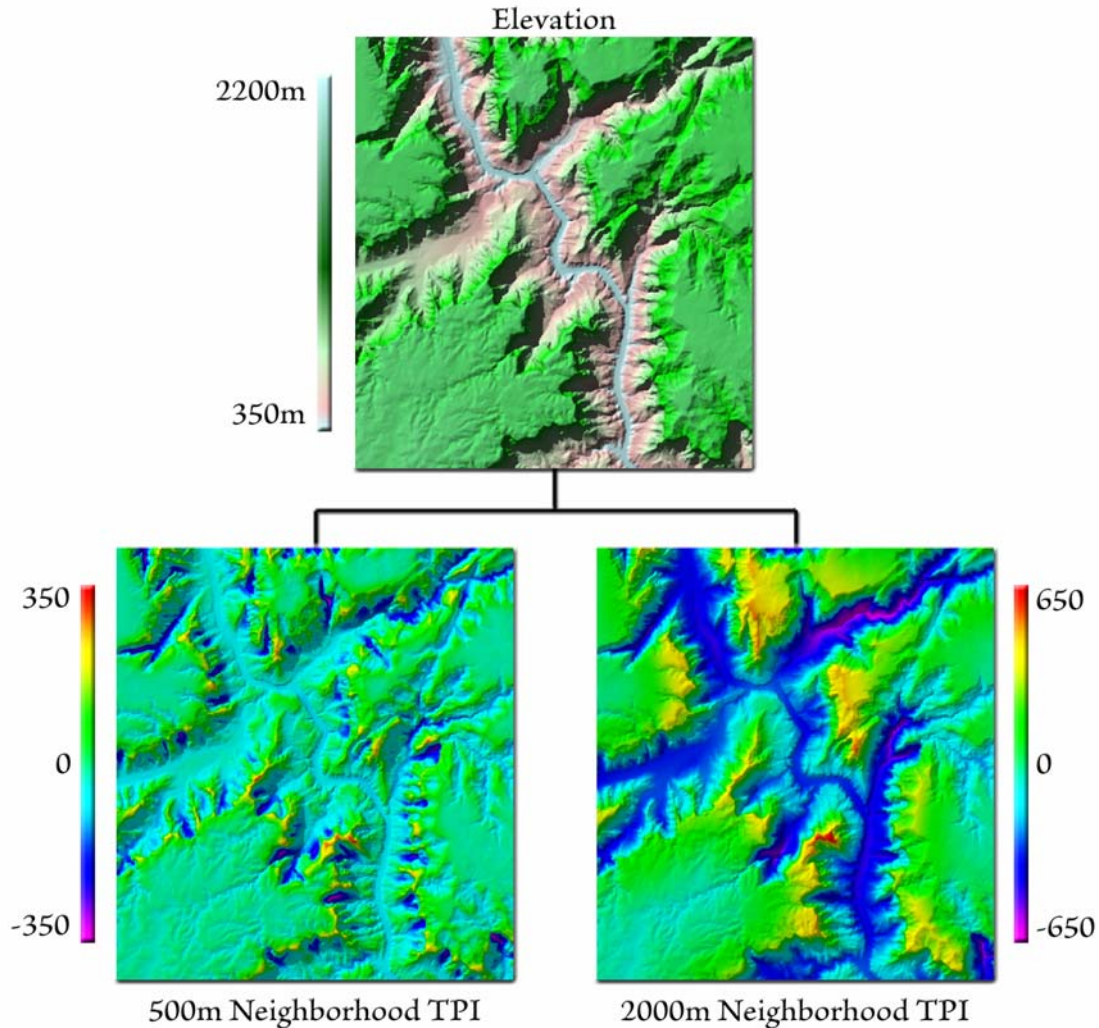
## TPI Values at 3 Different Scales



Users should consider carefully what scale is most relevant for the phenomenon being analyzed. If you are interested in topographic habitat characteristics of large, wide-ranging animals, you would likely define your landscape classifications in terms of large, distinctive topographic features. Cougars, for example, are likely to be much more influenced by a nearby ridgeline hundreds of feet high than in minor ripples and bumps on the landscape immediately surrounding them. Furthermore, a point on top of a small hill at the bottom of a canyon may be classified as a canyon bottom at one scale, or a hilltop at a different scale. Both are accurate and valid classifications, and the user must be responsible for knowing what scale is reasonable for their analysis.

Scale is determined by the neighborhood used in the analysis. The TPI values reflect the difference between the elevation in a particular cell and the average elevation of the cells around that cell. The Neighborhood defines what cells are considered to be “around” that cell.

In the illustration below, TPI values were calculated using 2 different neighborhoods. The left example used a circular neighborhood with a 500m radius, meaning that the TPI value for each cell reflected the difference between the elevation of that cell and the average elevation of all cells within 500m of that cell. This neighborhood did a good job of identifying extreme values in the side drainages of the canyon. The example on the right used a circular neighborhood with a 2000m radius and did a much better job of highlighting the overall canyon system.



These examples used circular neighborhoods, but other options are available. Weiss' examples used annular (ring- or doughnut-shaped) neighborhoods where only cells within a specified distance range are considered. Some researchers use rectangular neighborhoods, although in most cases circular or annular neighborhoods are more reasonable.

Wedge-shaped neighborhoods are useful for restricting your analysis to a particular direction. Weiss' poster discusses some ideas for future research in which he plans to compare directional TPI values in order to distinguish saddles from flat areas, ridges from hilltops and valleys from local depressions, as well as identify the general aspect of landforms.

For sophisticated neighborhood delineation, you can also define exactly which local cells should be considered as a neighborhood. These are referred to as "irregular" neighborhoods in ESRI software, and this extension provides a means for designing these specific custom neighborhoods (see p. 29).

**Classifying by Slope Position:** TPI values can easily be classified into slope position classes based on how extreme they are and by the slope at each point. There are a couple of strategies you can take to do this.

The easiest way is simply to set threshold values for the TPI grids themselves (or for standardized TPI grids; see p. 36). TPI values above a certain threshold might be classified as ridgetops or hilltops, while TPI values below a threshold might be classified as valley bottoms or

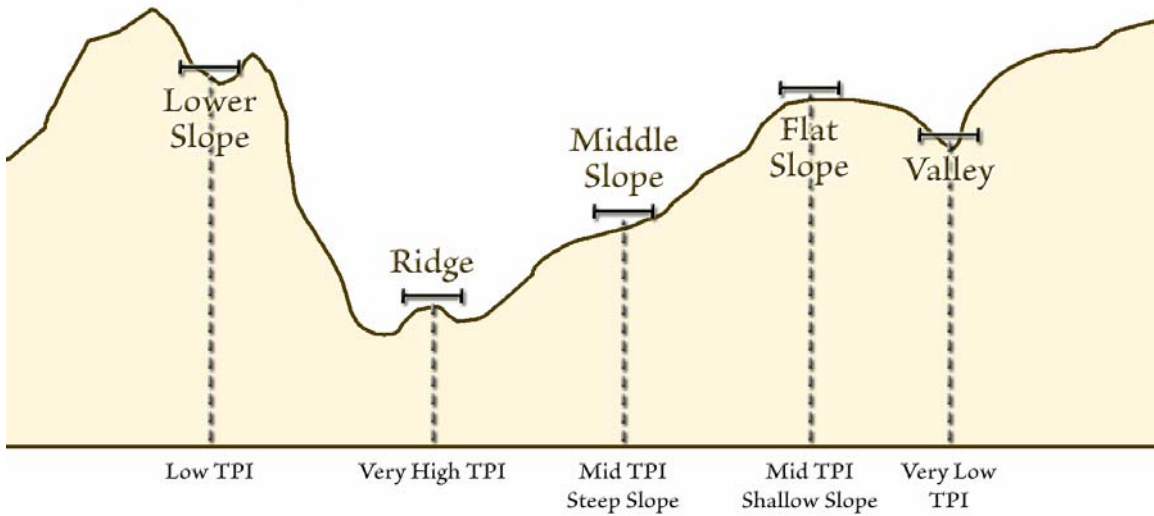


depressions. TPI values near 0 could be classified as flat plains (if the slope is near 0) or as mid-slope areas (if the slope is above a certain threshold). Dickson and Beier (in review) use this method in a study of the influences of topography on cougar movement.

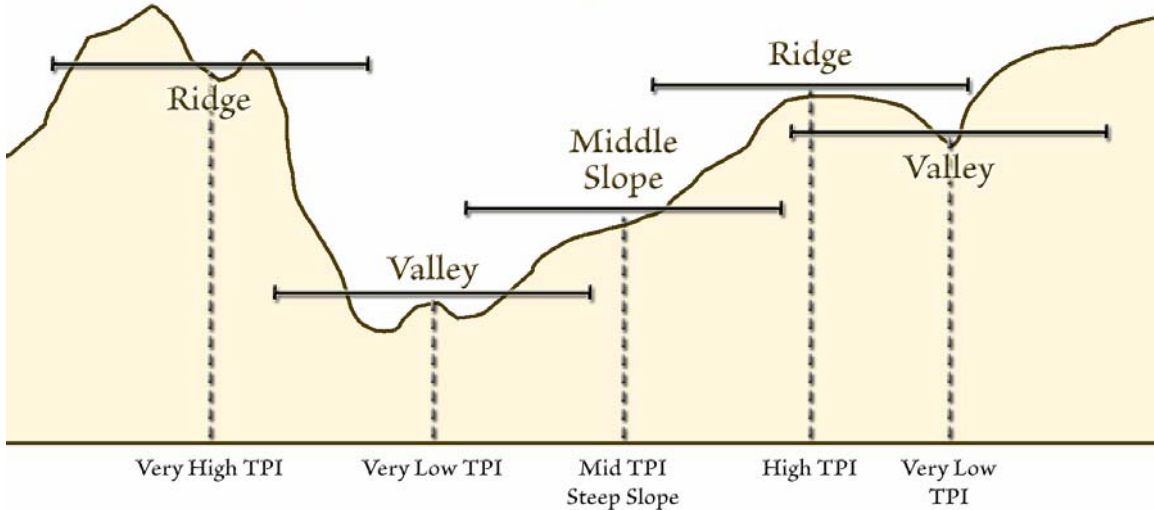
A somewhat more sophisticated method, illustrated by Weiss in his poster, is to define threshold TPI values in terms of standard deviations from the elevation, which therefore take into account the variability of elevation values within that neighborhood. This means that grid cells with identical TPI value may be classified differently in different areas, depending on the variability in their respective neighborhoods. This method may or may not be useful in your analysis. You would use this method if you felt that cells with high neighborhood elevation variability should have to meet a higher TPI threshold in order to be classified into some category.

As with TPI values in general, neighborhood size is also a critical component of the Slope Position classification process. Small neighborhoods capture small and local hills and valleys while large neighborhoods capture larger-scale features.

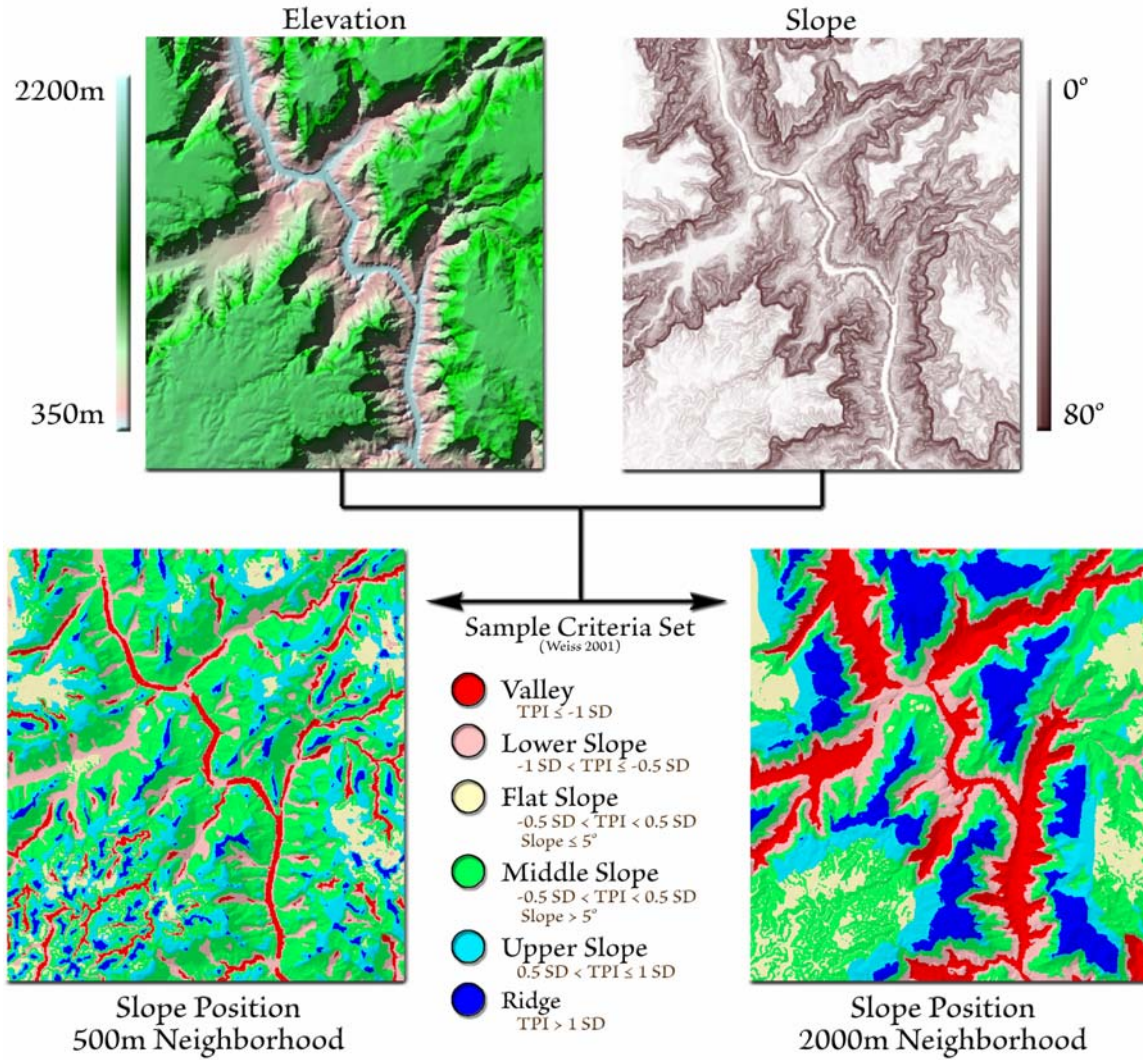
### Small-Neighborhood Slope Position Classification



### Large-Neighborhood Slope Position Classification



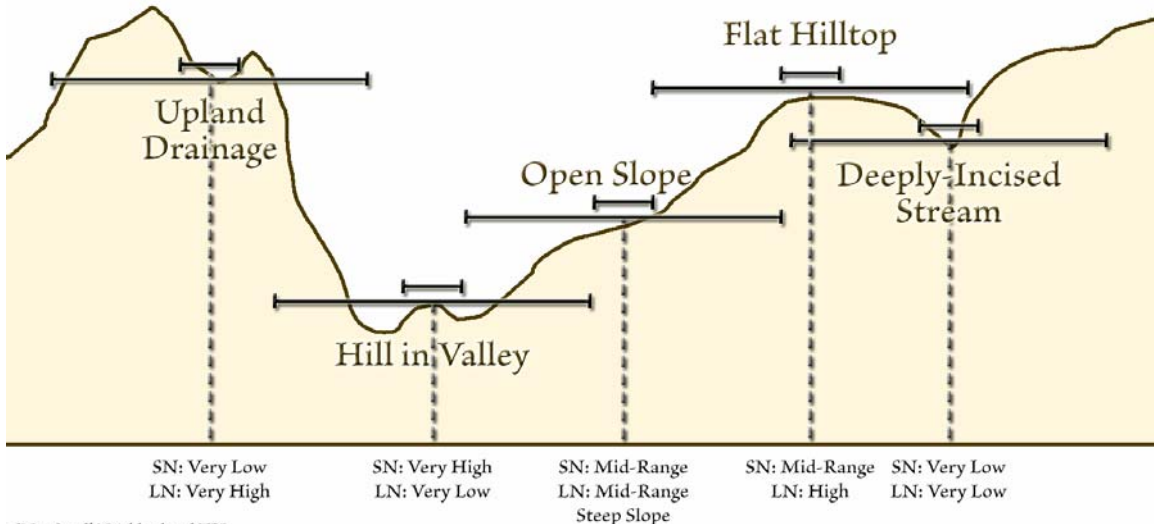
TPI values near 0 mean only that the elevation is close to the mean elevation of the neighborhood cells, and this could happen if that cell is in a flat area or if it is mid-slope somewhere. An easy way to distinguish between these 2 possibilities is to check the slope at that point. If the slope is near 0, then the point is probably on a flat area. A high slope value implies that the point is mid-slope somewhere. In his poster, Weiss demonstrates one possible classification process using both TPI and slope to generate a 6-category Slope Position grid.



**Classifying by Landform Category:** Landform category can be determined by classifying the landscape using 2 TPI grids at different scales. The combination of TPI values from different scales suggest various landform types.

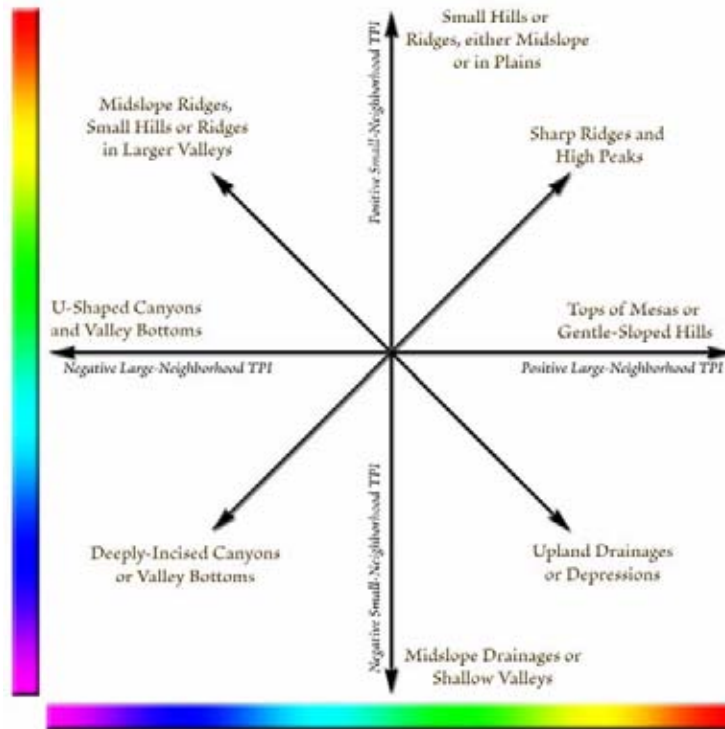


## Landform Classification, using Large and Small Neighborhood TPI

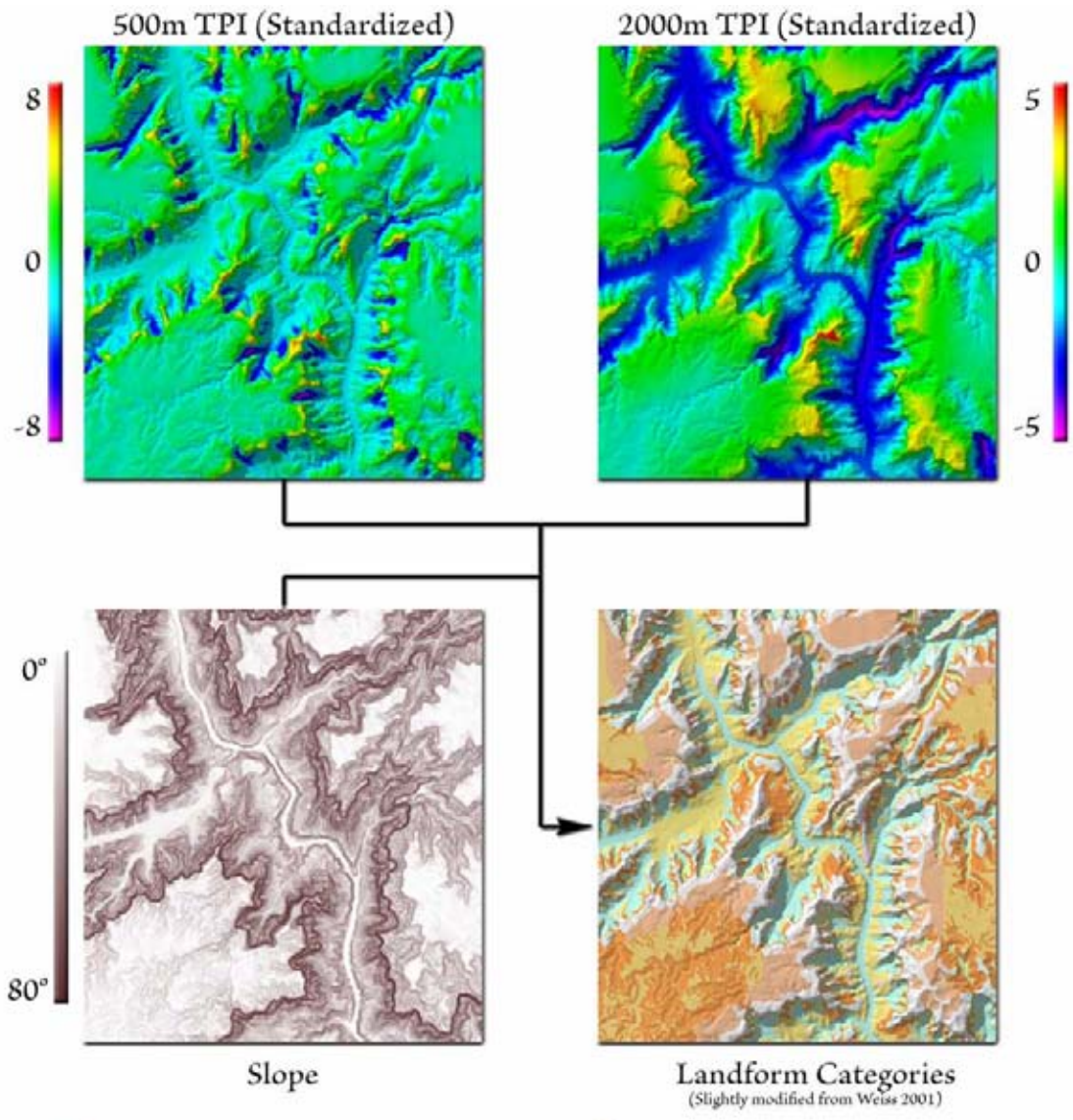


SN = Small Neighborhood TPI  
LN = Large Neighborhood TPI

For example, a high TPI value in a small neighborhood, combined with a low TPI value in a large neighborhood, would be classified as a local ridge or hill in a larger valley, while a low small-neighborhood TPI plus a high large-neighborhood TPI would be classified as an upland drainage or depression.



Weiss' poster provides examples demonstrating how 2 TPI grids and a slope grid can be used to identify canyons, mid-slope drainages, U-shaped valleys, plains, open slopes, upper slopes, mesas, mid-slope ridges and mountain tops.

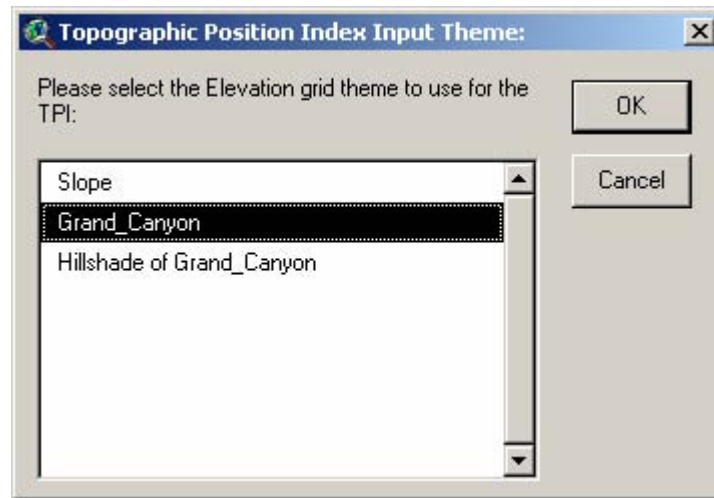


- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>● Canyons, Deeply Incised Streams<br/>SN: <math>TPI \leq -1</math><br/>LN: <math>TPI \leq -1</math></li> <li>● Midslope Drainages, Shallow Valleys<br/>SN: <math>TPI \leq -1</math><br/>LN: <math>-1 &lt; TPI &lt; 1</math></li> <li>● Upland Drainages, Headwaters<br/>SN: <math>TPI \leq -1</math><br/>LN: <math>TPI \geq 1</math></li> <li>● U-shaped Valleys<br/>SN: <math>-1 &lt; TPI &lt; 1</math><br/>LN: <math>TPI \leq -1</math></li> <li>● Plains<br/>SN: <math>-1 &lt; TPI &lt; 1</math><br/>LN: <math>-1 &lt; TPI &lt; 1</math><br/>Slope <math>\leq 5^\circ</math></li> </ul> | <ul style="list-style-type: none"> <li>● Open Slopes<br/>SN: <math>-1 &lt; TPI &lt; 1</math><br/>LN: <math>-1 &lt; TPI &lt; 1</math><br/>Slope <math>&gt; 5^\circ</math></li> <li>● Upper Slopes, Mesas<br/>SN: <math>-1 &lt; TPI &lt; 1</math><br/>LN: <math>TPI \geq 1</math></li> <li>● Local Ridges/Hills in Valleys<br/>SN: <math>TPI \geq 1</math><br/>LN: <math>TPI \leq -1</math></li> <li>● Midslope Ridges, Small Hills in Plains<br/>SN: <math>TPI \geq 1</math><br/>LN: <math>-1 &lt; TPI &lt; 1</math></li> <li>● Mountain Tops, High Ridges<br/>SN: <math>TPI \geq 1</math><br/>LN: <math>TPI \geq 1</math></li> </ul> |
|---|--|

## USING THIS EXTENSION:

### CALCULATING THE TOPOGRAPHIC POSITION INDEX:

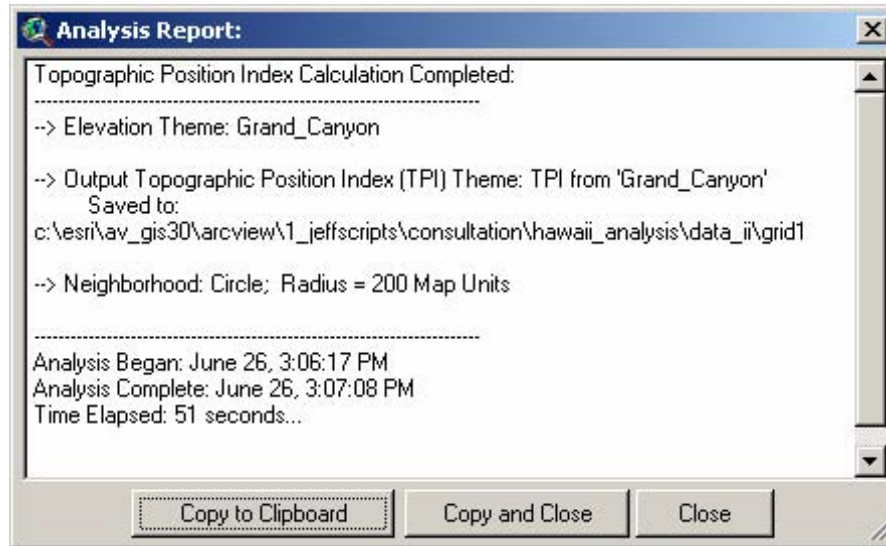
1. The Topographic Position Index grid is calculated from an elevation or surface model grid, so begin by adding this grid to your view.
2. In the “TPI” menu, click the option “Generate Topographic Position Index...”.



3. Next, specify the neighborhood to use to generate the TPI (see “Neighborhood Options” on p. 29 for a discussion of neighborhood types).

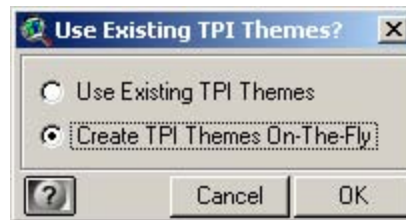


4. Click “OK” and the new TPI grid will be generated and added to your view. Upon completion, you will see a short report describing what was done:



#### CLASSIFYING LANDSCAPE INTO SLOPE POSITION:

1. The Slope Position grid is calculated from either an elevation or TPI grid, plus a slope grid, plus any additional classification criteria grids you may wish to include, so begin adding all relevant grids to your view.
2. In the “TPI” menu, click the option “Generate Slope Position Classification...”.
3. You will first be asked whether you have whether you are using an existing TPI grid or if you will generate it on-the-fly:



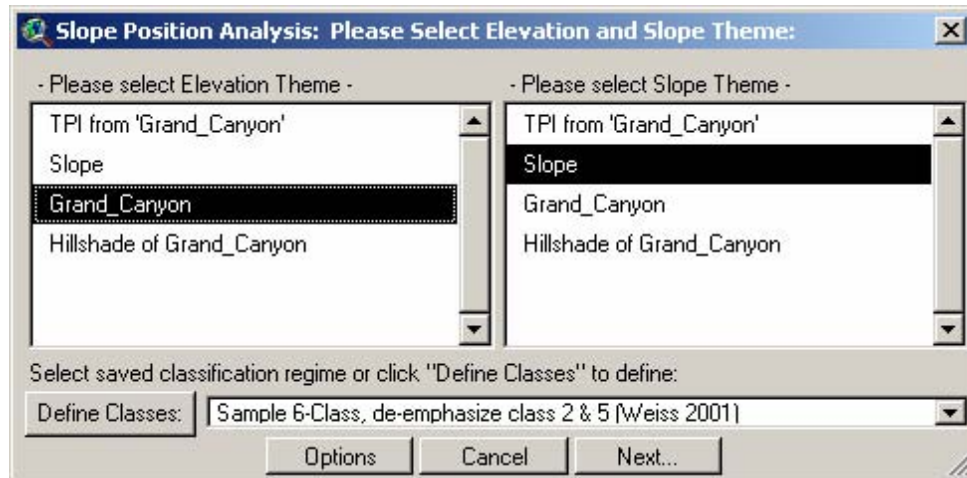
Using existing TPI grids makes the Slope Position classifications run faster (especially when using large grids or large neighborhoods), but they limit the options available when performing the classifications. When using existing TPI grids, you may only define classes in terms of actual TPI values, or standardized TPI values based on the mean and standard deviation of the entire TPI grid. When generating TPI grids on-the-fly, you may also define classes in terms of the standard deviation or variance of elevation values within the TPI neighborhood.

Recall that Slope Position classes are defined according to how extreme the TPI values are. If you use pre-generated TPI grids, then you are limited to setting threshold values on the TPI grids themselves (using either actual TPI units or standardized TPI units). There is no way to define classifications based on the standard deviation or variance of elevation values within the TPI neighborhood because the original elevation grid is not used in the analysis.

If you generate TPI grids on-the-fly, you have the option to save these TPI grids to your view by clicking the “Options” button in the next dialog.

4. Next, select the themes and the classification criteria in the “Slope Position Analysis” dialog to use in the analysis:





Select your elevation theme or your existing TPI theme from the first listbox, and the Slope theme from the second listbox, and then either select an existing classification regime or define a new one.

**Selecting and Defining a Classification Regime:** Your classification regime is the set of rules determining how your TPI and Slope values will be classified. You may use an existing criteria set or you may generate a new one. You may also load an existing one and modify it.

This extension comes with 2 criteria sets pre-loaded. The first set, from Weiss' poster, produces 6 categories and sets TPI thresholds using standard deviations (please refer to the discussion regarding "Classifying by Slope Position" [p. 6] to review this concept). Because it uses standard deviations, this criteria set can only be used if you opt to generate TPI grids on-the-fly.

**Sample 6-Class, de-emphasize class 2 & 5 (Weiss 2001)**

- Valley:**  $TPI \leq -1 \text{ SD}$
- Lower Slope:**  $-1 \text{ SD} < TPI \leq -0.5 \text{ SD}$
- Flat Slope:**  $-0.5 \text{ SD} < TPI < 0.5 \text{ SD}, \text{ Slope} \leq 5^\circ$
- Middle Slope:**  $-0.5 \text{ SD} < TPI < 0.5 \text{ SD}, \text{ Slope} > 5^\circ$
- Upper Slope:**  $0.5 \text{ SD} < TPI \leq 1 \text{ SD}$
- Ridge:**  $TPI > 1 \text{ SD}$

The second pre-loaded criteria set is derived from Dickson and Beier's Cougar study (Dickson and Beier in review), and produces 4 categories based on TPI values and slope. Because this criteria set uses TPI thresholds based on actual TPI values instead of standard deviations, you can use this set with existing TPI grids.

**Sample 4-Class (Dickson and Beier In Review)**

- Canyon Bottom:**  $TPI \leq -8$
- Gentle Slope:**  $-8 < TPI \leq 8, \text{ Slope} < 6^\circ$
- Steep Slope:**  $-8 < TPI \leq 8, \text{ Slope} \geq 6^\circ$
- Ridgeline:**  $TPI \geq 8$

If either of these criteria sets are appropriate for your analysis, simply select them from the list and click the "Next" button to continue. If you need to modify the criteria set or create a new one, click the "Define Classes" button to open the "Slope Position Parameters" dialog:



**Criteria Label and Value:** The “Criteria Description” at the top is the text label for any particular classification. The Classification Value is the numeric value associated with that classification. Both of these values will appear in the final attribute table for the classification grid.

**IMPORTANT:** The classification values also determine the rank or precedence level of a classification in cases where you may have overlapping criteria. If multiple classification criteria reference the same *Elevation / TPI / Slope* value or combination of values, then the classification with the lowest Classification Value will take precedence. For example, if 2 criteria both reference Grid A, and are defined as follows:

Criteria with Classification Value = 1; defined as  $[5 \leq \text{Grid A} < 10]$   
 Criteria with Classification Value = 2; defined as  $[8 \leq \text{Grid A} < 15]$

Then, all Grid A Cells with values 8 and 9 meet both criteria. Because the first criteria has the lowest Classification Value, then these cells would be classified as Criteria 1.

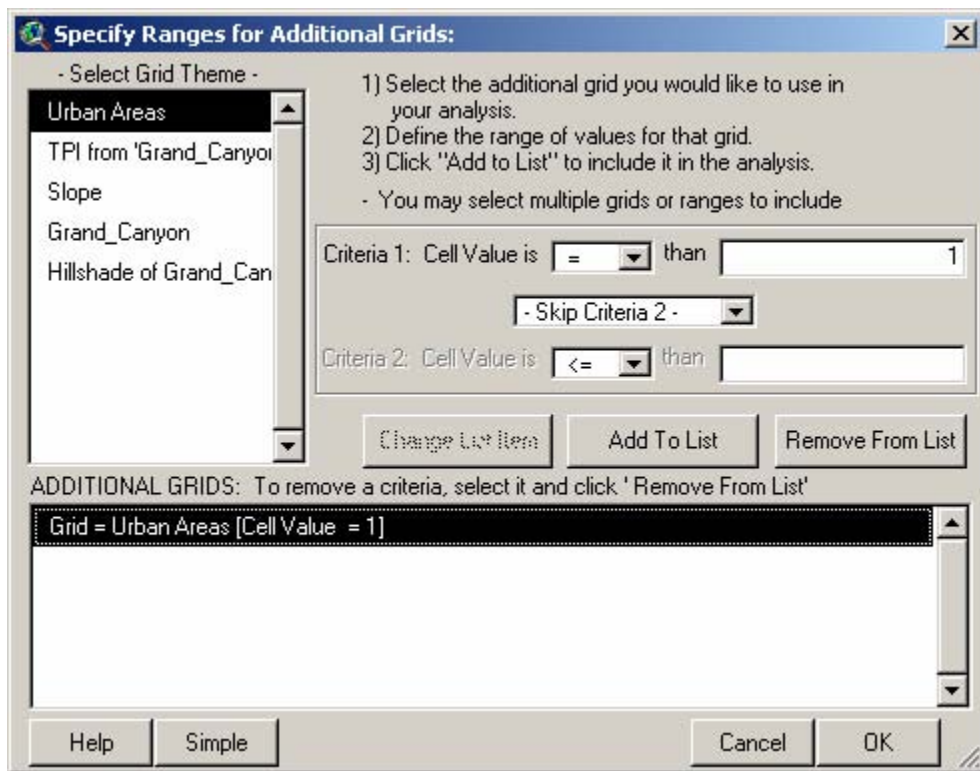
**TPI Criteria:** Below the Criteria Description section, define the TPI values that apply to this classification. Notice that you may opt to skip the second set of criteria if you only have one relationship that applies to this class (click the drop-down box containing the word “and” and select “Skip Criteria 2”). Note also that you must specify the units to use in your

classification. If you are generating your TPI grid on-the-fly as you calculate slope position, then you have the option to use any of the following units. If you are using an existing TPI grid, then you can only choose from Units and Standardized Units:

- 1) Units: Thresholds are based on actual TPI values.
- 2) Standardized Units: The TPI grid is standardized on-the-fly (see p. 36), and thresholds are based on these standardized units.
- 3) SD (Standard Deviations): The elevation grid is converted on-the-fly to a TPI grid based on some specified neighborhood, and a grid of neighborhood elevation standard deviation values is generated on-the-fly at the same time. TPI thresholds are then based on the number of specified standard deviation units from the mean. This option is only available if you are generating your TPI grid on-the-fly.
- 4) VAR (Variances): The elevation grid is converted on-the-fly to a TPI grid based on some specified neighborhood, and a grid of neighborhood elevation variance values is generated on-the-fly at the same time. TPI thresholds are then based on the number of variance units from the mean. One variance unit is equal to the square of the standard deviation unit. This option is only available if you are generating your TPI grid on-the-fly.

Slope Criteria: Below the TPI section, define the Slope values that are associated with this class (if any). If this class does not use slope, uncheck the "Include Slope Value" box.

Additional Grids: If you have any additional grids to include in the analysis, check the box "Include Additional Grid Data in Criteria Definition" and then click the "Additional Grid Parameters" button. For example, if you have a grid of urban vs. non-urban land, and you want to set one of your classes so it only applies to urban areas, you would click the "Additional Grid Parameters" button and fill in the "Specify Ranges for Additional Grids" dialog as follows:



Select the grid theme of interest and assign the range of values that apply to this classification definition. Notice that you may opt to skip the second set of criteria if you only

have one relationship that applies to this class (click the drop-down box containing the word "and" and select "Skip Criteria 2"). Click "Add to List" to add it to the definition for this class.

If you want to edit existing grid threshold values, select that grid in the "ADDITIONAL GRIDS" listbox. The current threshold values will appear above, so make any changes you wish and click "Change List Item" to save the changes.

You may remove any existing grid threshold values by selecting it in the "ADDITIONAL GRIDS" listbox and clicking the "Remove from List" button. When you have finished defining all the additional grid parameters associated with this class, click the "OK" button to return to the main dialog.

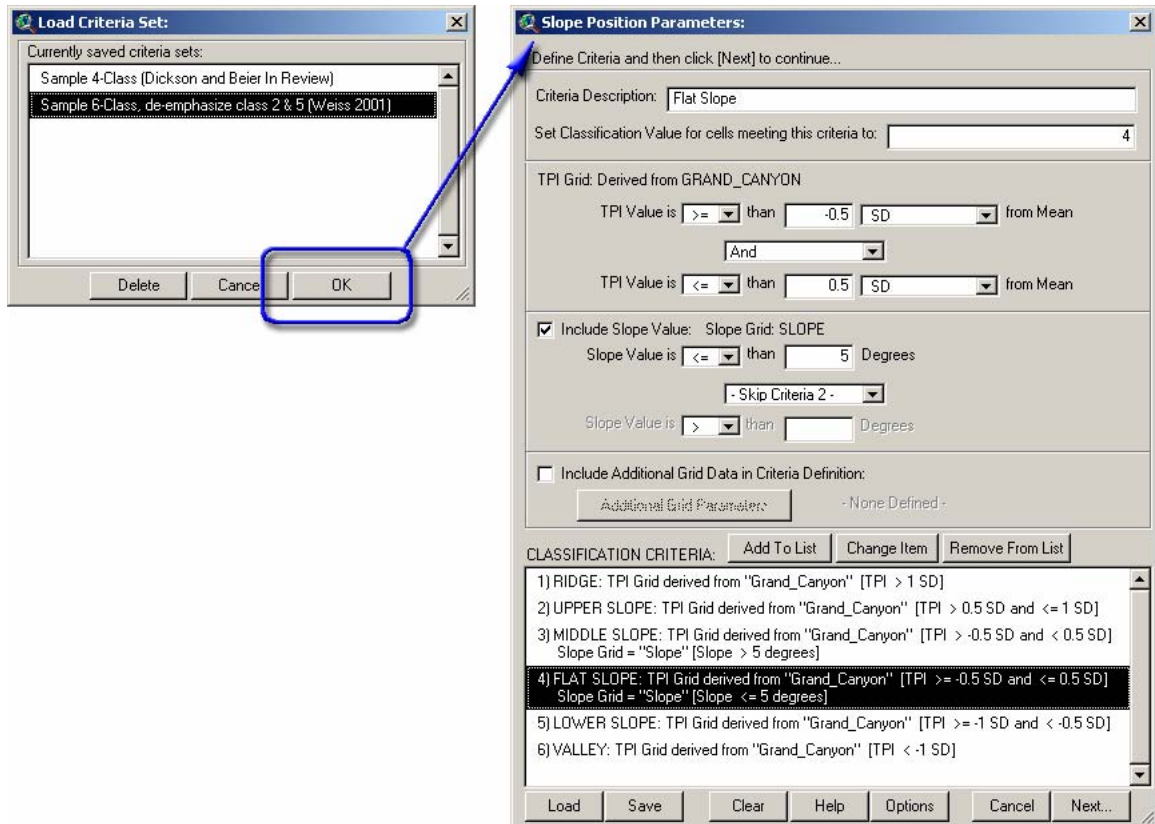
**IMPORTANT:** If you plan to save the grid criteria set that includes references to these additional grids, be aware that the saved criteria set will reference these grids specifically. If a user tries to use this saved criteria set and does not have grid themes with these names in their view, then the extension will halt operation and ask the user to identify that grid before continuing.

Classification Criteria List: The box at the bottom of the dialog labeled "CLASSIFICATION CRITERIA" lists all the classification definitions established for this criteria set. When you finish setting the label, value, TPI and Slope parameters for a particular class, click the "Add to List" button to add it to the complete list.

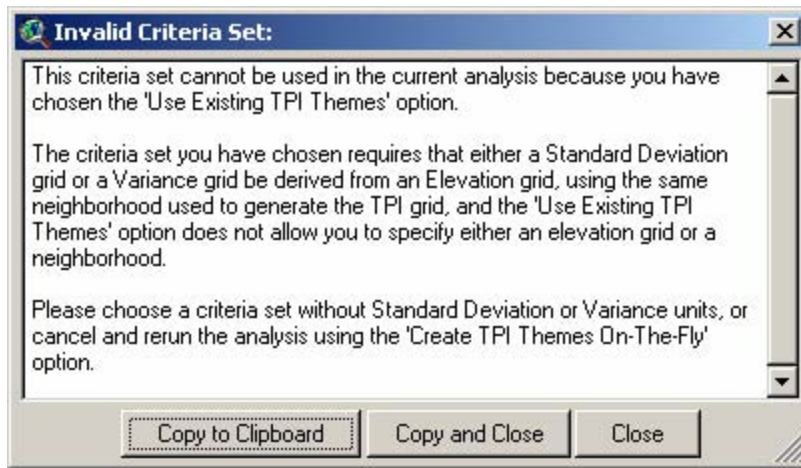
If you wish to edit any of the existing definitions, select it in the Criteria List box and all the parameter boxes will fill up with the respective values. After you have made any changes you want, click the "Change Item" button to save the changes.

If you wish to delete any of the existing definitions, select it in the Criteria List box and click the "Remove From List" button.

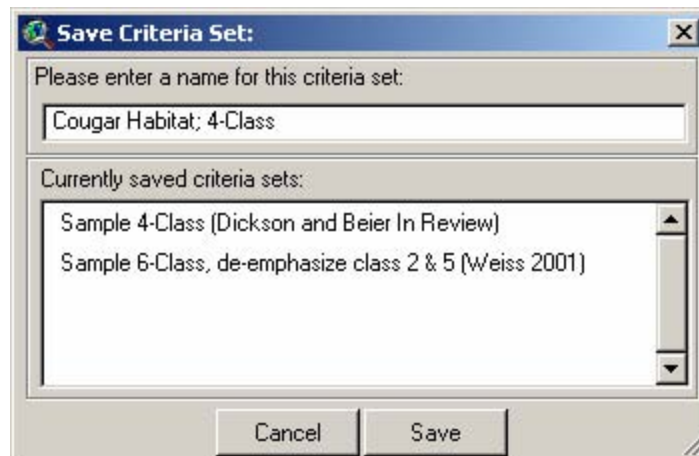
Loading Criteria Sets: You can load pre-existing or saved criteria sets by clicking the "Load" button. Simply select the criteria set to load and click "OK":



**IMPORTANT:** If you are using existing TPI themes in this analysis, then you cannot use criteria sets that use Standard Deviations or Variance units. If you try to load an existing criteria set that uses these units, you will see a message explaining that this is an invalid criteria set and the selected criteria set will not load:



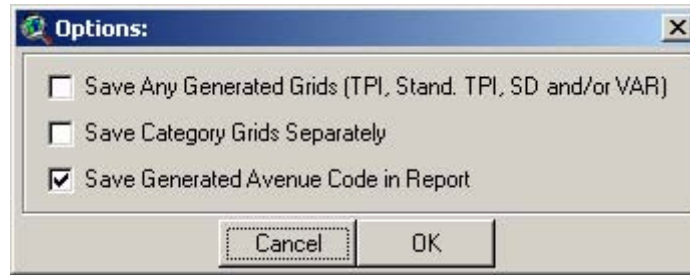
Saving Criteria Sets: If you expect to use your criteria set over again or wish to share it with others, it may make sense to save it so it can be distributed and easily reloaded. Click the “Save” button to open the “Save Criteria Set” dialog:



Give your new criteria set a name and click “Save”, and it will be saved to the “Slope Position Saved Criteria Set” table (../tpi\_slope\_criteria.dbf). See the discussion on Saved Criteria Tables (p. 37) for more information on this table and how to share saved criteria sets.

**Important:** The saved criteria set will reference any Elevation, TPI or Slope grid the user wants to use. The saved sets don’t reference specific grids, but rather any grid that the user designates as Elevation, TPI or Slope. HOWEVER, if any additional grids are added to the saved set, then those are saved by name. For example, the Weiss 10-class Landform criteria set references a slope grid as an additional grid, and therefore it requires a grid theme named “Slope”. If the extension doesn’t find a grid named “Slope” in your view, it will halt operation and ask you to identify that grid before continuing.

Options: The “Options” button allows you to save some of the intermediate data and some of the code used to generate the data.



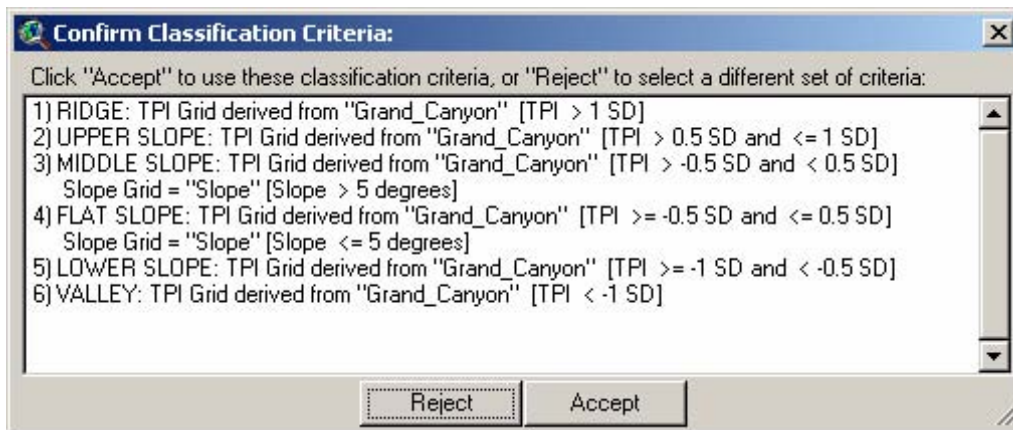
- a) *Save Any Generated Grids*: If you are generating your TPI grid on-the-fly, or if you are using standardized units, standard deviation units or variance units, then this option will save those grids and add them as new themes to your view. If you do not choose this option, these grids are erased after use.
- b) *Save Category Grids Separately*: This option saves each separate class in a separate grid and adds them to your view. Cells in these category grids will have either the class value or null values.
- c) *Save Generated Avenue Code in Report*: This extension generates the category grids by building an Avenue script containing the necessary parameters, then executing that script. This option lets you save the respective Avenue scripts to the final report so you can examine exactly what the extension is doing. A sample script would look something like the following:

```

-----<<<<<< Begin Avenue Script >>>>>>-----
theTPIGrid = self.Get(0)
theSlopeGrid = self.Get(1)
theSDGrid = self.Get(2)
theUnitsGrid = self.Get(3)
theVarGrid = self.Get(4)
theStdTPIGrid = self.Get(5)
theClassGrid = ((theStdTPIGrid >-5.00000000000000) And (theStdTPIGrid
<5.00000000000000)) and (theSlopeGrid >(5.00000000000000))
return {theClassGrid}
-----<<<<<< End Avenue Script >>>>>>-----

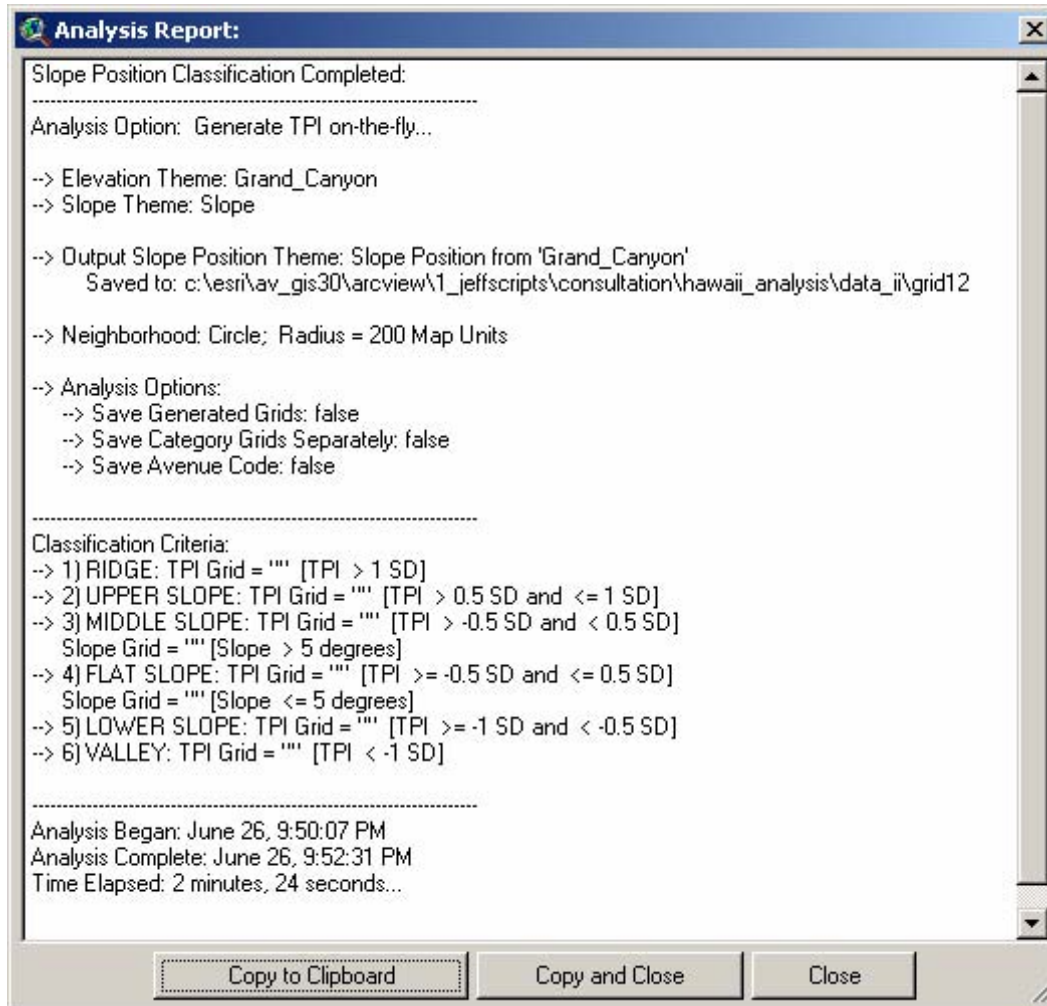
```

5. After you have selected or created your classification regime, click the "Next..." button on the "Slope Position Analysis" dialog and confirm your selected classification definitions:



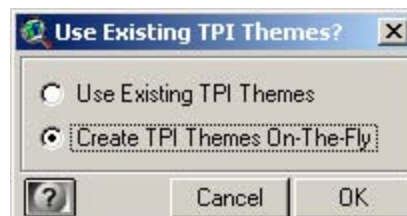
6. If you are generating TPI values on-the-fly, then you will next be asked to define your neighborhood (see "Neighborhood Options" on p. 29 for a discussion of neighborhood types). If you are using existing TPI grids, then the extension will proceed directly to the analysis.
7. Upon completion, the extension will add your new Slope Position grid to your view and generate a report detailing exactly what it did:





**CLASSIFYING LANDSCAPE INTO LANDFORM CATEGORY:**

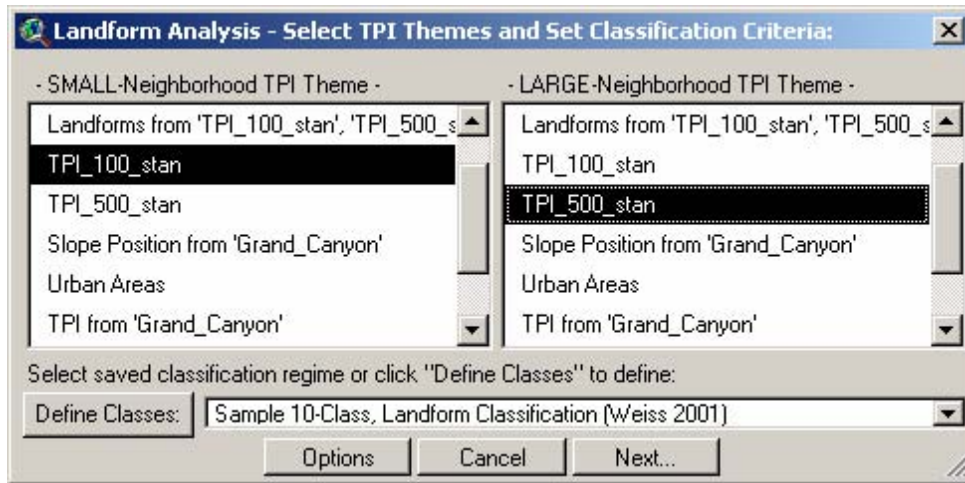
1. The Landform Classification grid is calculated from either a single elevation grid or 2 TPI grids, plus any additional classification criteria grids you may wish to include, so begin adding all relevant grids to your view.
2. In the “TPI” menu, click the option “Generate Landform Classification...”.
3. You will first be asked whether you have whether you are using existing TPI grids or if you will generate them on-the-fly:



Using existing TPI grids makes the Landform classifications run faster (especially when using large grids or large neighborhoods), but they limit the options available when performing the classifications. When using existing TPI grids, you may only define classes in terms of actual TPI values, or standardized TPI values based on the mean and standard deviation of the

entire TPI grid. When generating TPI grids on-the-fly, you may also define classes in terms of the standard deviation or variance of elevation values within the TPI neighborhood.

- Next, select the themes and the classification criteria in the “Landform Analysis” dialog to use in the analysis:



If you are generating TPI themes on-the-fly, then you will only need to select a single elevation theme in this dialog. The small- and large-neighborhood TPI themes will be generated automatically based on neighborhoods you will specify. If you are using existing TPI themes, then you will need to select your small- and large-neighborhood TPI themes in this dialog. After specifying the themes, either select an existing classification regime or define a new one.

**Selecting and Defining a Classification Regime:** Your classification regime is the set of rules determining how your Landform values will be classified based on your 2 TPI grids. You may use an existing criteria set or you may generate a new one. You may also load an existing one and modify it.

This extension comes with 1 landform criteria set pre-loaded. The set, from Weiss' poster, produces 10 landform categories and sets TPI thresholds using standardized TPI units (please refer to the discussions regarding “Classifying by Landform Category” [p. 8] and “Standardizing Grids” [p. 36] to review this concept). Because it uses standardized TPI units, this criteria set can be used whether you opt to generate TPI grids on-the-fly or to use existing TPI grids.

**Sample 10-Class, Landform Classification (Weiss 2001; slightly modified)**

**Canyons, Deeply Incised Streams:** Small Neighborhood TPI:  $TPI \leq -1$

Large Neighborhood TPI:  $TPI \leq -1$

**Midslope Drainages, Shallow Valleys:** Small Neighborhood TPI:  $TPI \leq -1$

Large Neighborhood TPI:  $-1 < TPI < 1$

**Upland Drainages, Headwaters:** Small Neighborhood TPI:  $TPI \leq -1$

Large Neighborhood TPI:  $TPI \geq 1$

**U-shaped Valleys:** Small Neighborhood TPI:  $-1 < TPI < 1$

Large Neighborhood TPI:  $TPI \leq -1$

**Plains Small:** Neighborhood TPI:  $-1 < TPI < 1$

Large Neighborhood TPI:  $-1 < TPI < 1$

Slope  $\leq 5^\circ$

**Open Slopes:** Small Neighborhood TPI:  $-1 < TPI < 1$

Large Neighborhood TPI:  $-1 < TPI < 1$

Slope  $> 5^\circ$

**Upper Slopes, Mesas:** Small Neighborhood TPI:  $-1 < TPI < 1$

Large Neighborhood TPI:  $TPI \geq 1$

- Local Ridges/Hills in Valleys:** Small Neighborhood TPI:  $TPI \geq 1$   
Large Neighborhood TPI:  $TPI \leq -1$
- Midslope Ridges, Small Hills in Plains:** Small Neighborhood TPI:  $TPI \geq 1$   
Large Neighborhood TPI:  $-1 < TPI < 1$
- Mountain Tops, High Ridges:** Small Neighborhood TPI:  $TPI \geq 1$   
Large Neighborhood TPI:  $TPI \geq 1$

If this criteria set, or any other saved criteria set appearing in the drop-down list, is appropriate for your analysis, simply select it from the list and click the "Next" button to continue. If you need to modify the criteria set or create a new one, click the "Define Classes" button to open the "Landform Classification Parameters" dialog:

Criteria Label and Value: The "Criteria Description" at the top is the text label for any particular classification. The Classification Value is the numeric value associated with that classification. Both of these values will appear in the final attribute table for the classification grid.

**IMPORTANT:** The classification values also determine the rank or precedence level of a classification in cases where you may have overlapping criteria. If multiple classification criteria reference the same small- and large-neighborhood TPI values or combination of

values, then the classification with the lowest Classification Value will take precedence. For example, if 2 criteria both reference Grid A, and are defined as follows:

Criteria with Classification Value = 1; defined as  $[5 \leq \text{Grid A} < 10]$

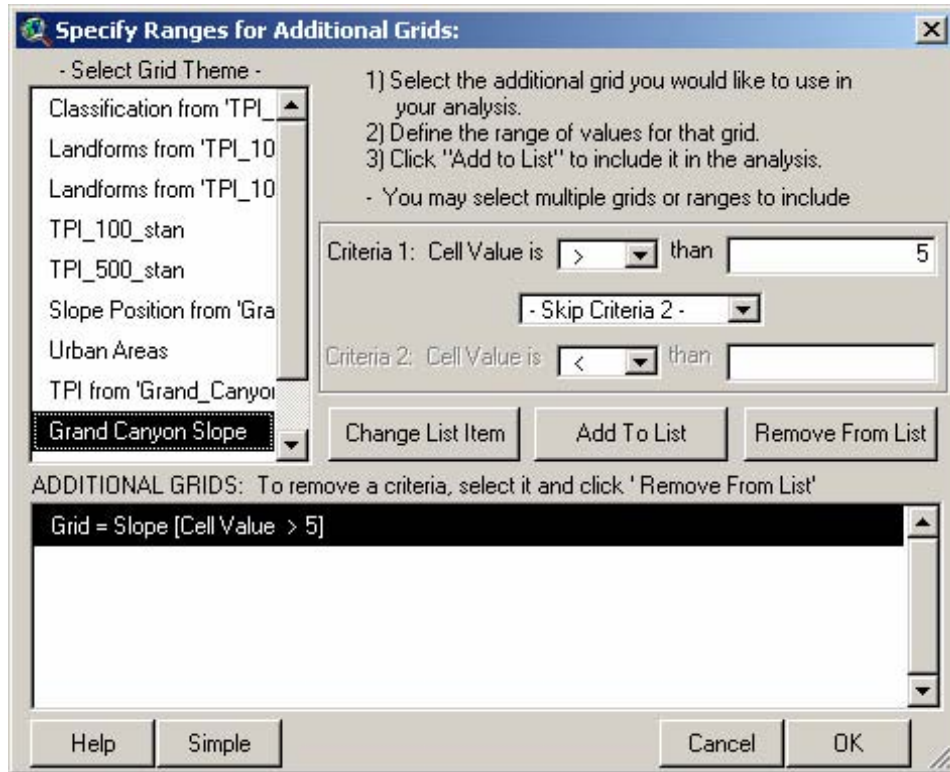
Criteria with Classification Value = 2; defined as  $[8 \leq \text{Grid A} < 15]$

Then, all Grid A Cells with values 8 and 9 meet both criteria. Because the first criteria has the lowest Classification Value, then these cells would be classified as Criteria 1.

Small- and Large-Neighborhood TPI Criteria: Below the Criteria Description section, define the small- and large-neighborhood TPI values that apply to this classification. Notice that you may opt to skip the second set of criteria if you only have one relationship that applies to this class (click the drop-down box containing the word “and” and select “Skip Criteria 2”). Note also that you must specify the units to use in your classification. If you are generating your TPI grids on-the-fly as you calculate landform category, then you have the option to use any of the following units. If you are using existing TPI grids, then you can only choose from Units and Standardized Units:

- 1) Units: Thresholds are based on actual TPI values.
- 2) Standardized Units: The TPI grid is standardized on-the-fly (see p. 36), and thresholds are based on these standardized units.
- 3) SD (Standard Deviations): The elevation grid is converted on-the-fly to a TPI grid based on some specified neighborhood, and a grid of neighborhood elevation standard deviation values is generated on-the-fly at the same time. TPI thresholds are then based on the number of specified standard deviation units from the mean. This option is only available if you are generating your TPI grid on-the-fly.
- 4) VAR (Variances): The elevation grid is converted on-the-fly to a TPI grid based on some specified neighborhood, and a grid of neighborhood elevation variance values is generated on-the-fly at the same time. TPI thresholds are then based on the number of variance units from the mean. One variance unit is equal to the square of the standard deviation unit. This option is only available if you are generating your TPI grid on-the-fly.

Additional Grids: If you have any additional grids to include in the analysis, check the box “Include Additional Grid Data in Criteria Definition” and then click the “Additional Grid Parameters” button. For example, if you have a grid of slope values, and you want to set some of your classes so that they only apply to certain slope ranges, you would click the “Additional Grid Parameters” button and fill in the “Specify Ranges for Additional Grids” dialog as follows:



Select the grid theme of interest and assign the range of values that apply to this classification definition. Notice that you may opt to skip the second set of criteria if you only have one relationship that applies to this class (click the drop-down box containing the word "and" and select "Skip Criteria 2"). Click "Add to List" to add it to the definition for this class.

If you want to edit existing grid threshold values, select that grid in the "ADDITIONAL GRIDS" listbox. The current threshold values will appear above, so make any changes you wish and click "Change List Item" to save the changes.

You may remove any existing grid threshold values by selecting it in the "ADDITIONAL GRIDS" listbox and clicking the "Remove from List" button. When you have finished defining all the additional grid parameters associated with this class, click the "OK" button to return to the main dialog.

**IMPORTANT:** If you plan to save the grid criteria set that includes references to these additional grids, be aware that the saved criteria set will reference these grids specifically. If a user tries to use this saved criteria set and does not have grid themes with these names in their view, then the extension will halt operation and ask the user to identify that grid before continuing.

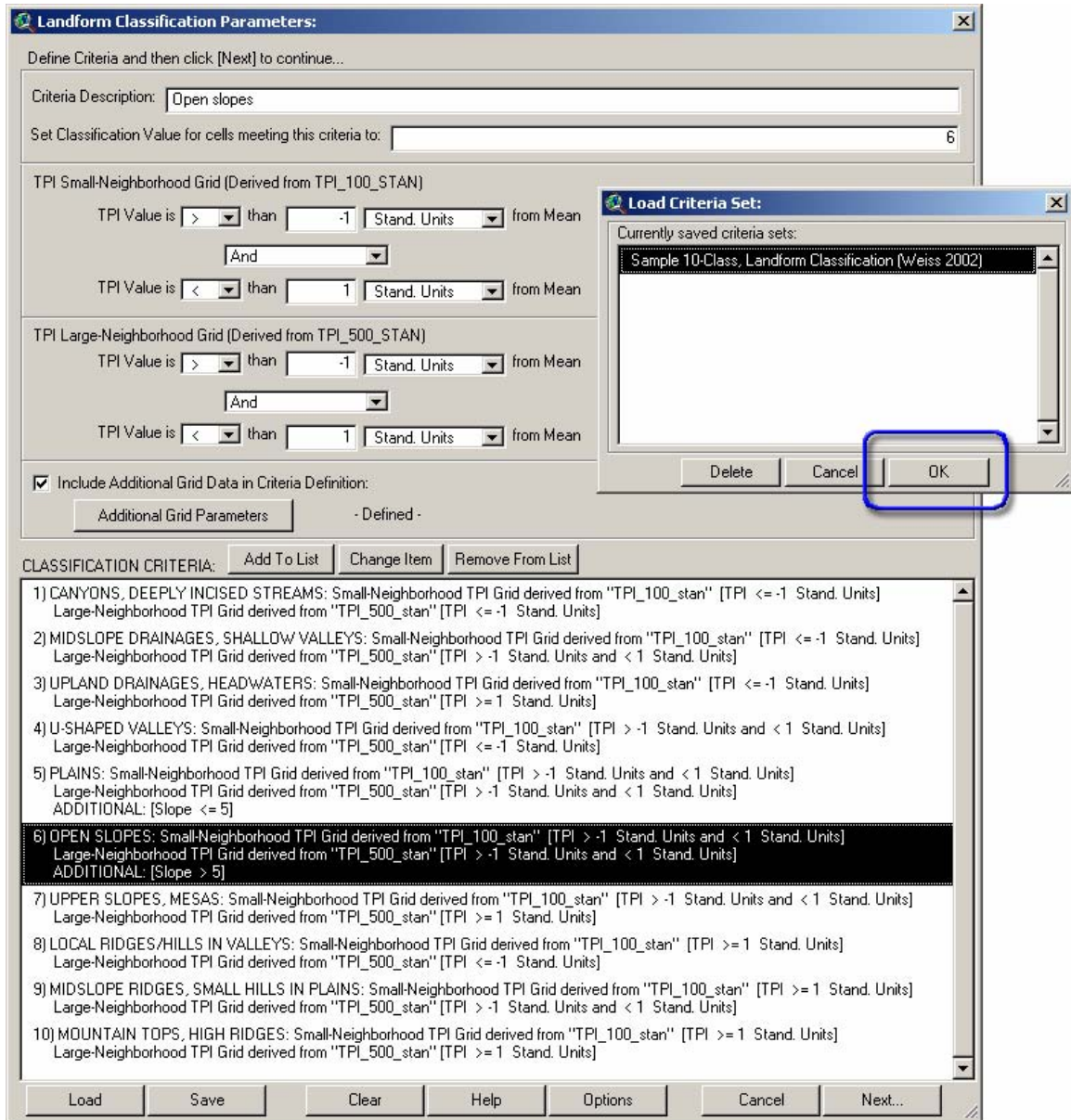
Classification Criteria List: The box at the bottom of the dialog labeled "CLASSIFICATION CRITERIA" lists all the classification definitions established for this criteria set. When you finish setting the label, value, small- and large-neighborhood parameters for a particular class, click the "Add to List" button to add it to the complete list.

If you wish to edit any of the existing definitions, select it in the Criteria List box and all the parameter boxes will fill up with the respective values. After you have made any changes you want, click the "Change Item" button to save the changes.

If you wish to delete any of the existing definitions, select it in the Criteria List box and click the "Remove From List" button.

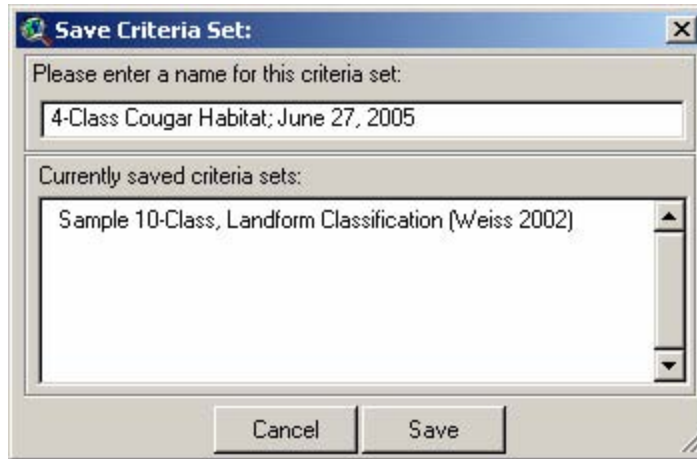
Loading Criteria Sets: You can load pre-existing or saved criteria sets by clicking the "Load" button. Simply select the criteria set to load and click "OK":





**IMPORTANT:** If you are using existing TPI themes in this analysis, then you cannot use criteria sets that use Standard Deviations or Variance units. If you try to load an existing criteria set that uses these units, you will see a message explaining that this is an invalid criteria set and the selected criteria set will not load:

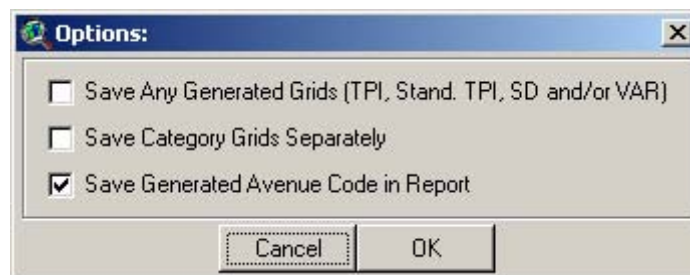
Saving Criteria Sets: If you expect to use your criteria set over again or wish to share it with others, it may make sense to save it so it can be distributed and easily reloaded. Click the "Save" button to open the "Save Criteria Set" dialog:



Give your new criteria set a name and click “Save”, and it will be saved to the “Landform Classification Saved Criteria Set” table (../tpi\_landform\_criteria). See the discussion on Saved Criteria Tables (p. 37) for more information on this table and how to share saved criteria sets.

**Important:** The saved criteria set will reference any Elevation or TPI grid the user wants to use. The saved sets don’t reference specific grids, but rather any grid that the user designates as Elevation or TPI. HOWEVER, if any additional grids are added to the saved set, then those are saved by name. For example, the Weiss 10-class Landform criteria set references a slope grid as an additional grid, and therefore it requires a grid theme named “Slope”. If the extension doesn’t find a grid named “Slope” in your view, it will halt operation and ask you to identify that grid before continuing.

Options: The “Options” button allows you to save some of the intermediate data and some of the code used to generate the data.



- a) *Save Any Generated Grids:* If you are generating your TPI grids on-the-fly, or if you are using standardized units, standard deviation units or variance units, then this option will save those grids and add them as new themes to your view. If you do not choose this option, these grids are erased after use.
- b) *Save Category Grids Separately:* This option saves each separate class in a separate grid and adds them to your view. Cells in these category grids will have either the class value or null values.
- c) *Save Generated Avenue Code in Report:* This extension generates the category grids by building an Avenue script containing the necessary parameters, then executing that script. This option lets you save the respective Avenue scripts to the final report so you can examine exactly what the extension is doing. A sample script would look something like the following:

```
-----<<<<<< Begin Avenue Script >>>>>>-----
theTPISmallGrid = self.Get(0)
theTPILargeGrid = self.Get(1)
theTPIUnitsGrid = self.Get(2)
```

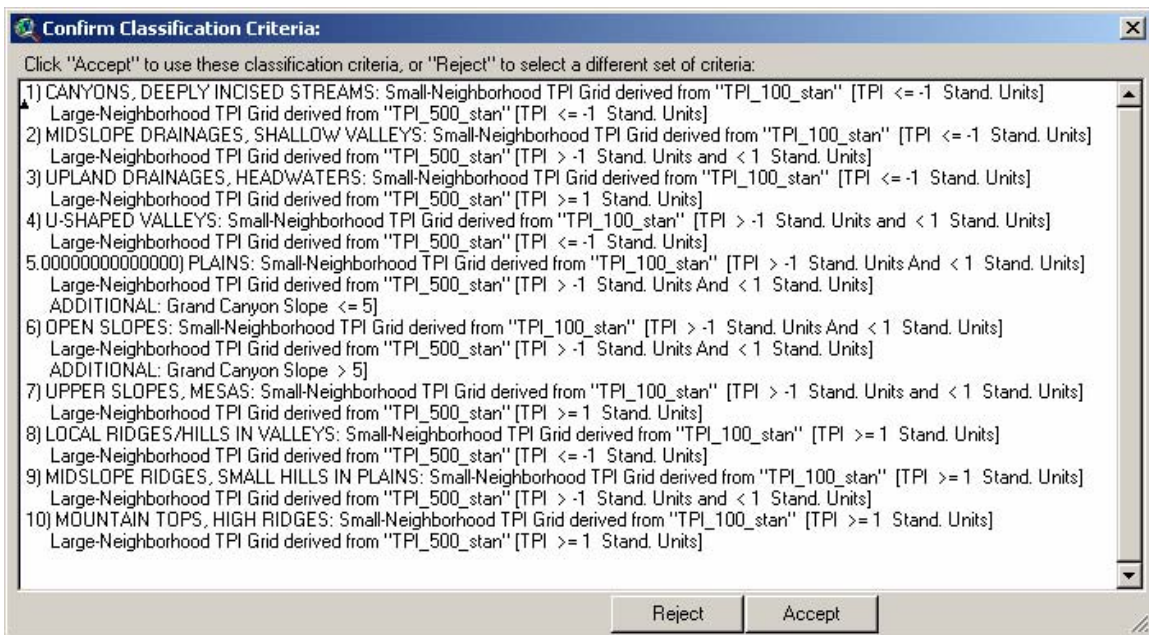
```

theTPI1SDGrid = self.Get(3)
theTPI1VARGrid = self.Get(4)
theTPI1StdTPIGrid = self.Get(5)
theTPI2UnitsGrid = self.Get(6)
theTPI2SDGrid = self.Get(7)
theTPI2VARGrid = self.Get(8)
theTPI2StdTPIGrid = self.Get(9)
UseOther = self.Get(10)
theOtherGrids = self.Get(11)
GridNumber1 = theOtherGrids.Get(0).Get(1)
theClassGrid = (((theTPI1StdTPIGrid >(-1.00000000000000)) And (theTPI1StdTPIGrid
<(1.00000000000000))) and ((theTPI2StdTPIGrid >(-1.00000000000000)) And
(theTPI2StdTPIGrid <(1.00000000000000)))) and ((GridNumber1 <= 5))
return {theClassGrid}

```

-----<<<<<<<< End Avenue Script >>>>>>>-----

- After you have selected or created your classification regime, click the "Next..." button on the "Landform Analysis" dialog and confirm your selected classification definitions:



- If you are generating TPI values on-the-fly, then you will next be asked to define your small and large neighborhood (see "Neighborhood Options" on p. 29 for a discussion of neighborhood types). If you are using existing TPI grids, then the extension will proceed directly to the analysis.
- Upon completion, the extension will add your new Landform grid to your view and generate a report detailing exactly what it did:



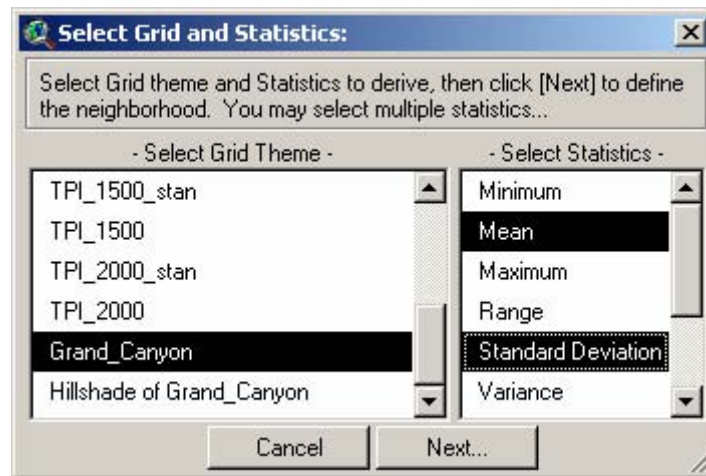


## ADDITIONAL FUNCTIONS:

### CALCULATING NEIGHBORHOOD STATISTICS

This option generates a new grid representing neighborhood statistics from an existing grid. Each cell value in the new grid represents some statistic (i.e. mean, min, max, standard deviation, etc.) from some region of cells surrounding the original cell in the original grid, where the region is called a Neighborhood and can come in a variety of shapes and sizes. The standard Spatial Analyst Neighborhood Statistics function includes options for rectangles, circles, wedges and doughnuts (see ArcView Help files for more information; especially “FocalStats Discussion”). The Neighborhood Statistics function included with this extension is very similar to the original Spatial Analyst function except that this tool allows you to generate custom-shaped neighborhoods and also generate multiple sets of statistics with one operation.

Click the “Neighborhood Statistics” menu item to start the process. You will be asked to identify the grid theme you wish to analyze and the statistics you wish to generate. You may select multiple statistics. This window is resizable by dragging on a corner.



### Statistic Options:

If your Grid Theme is a Floating Point theme (i.e. with decimal numbers, and therefore without an attribute table), then your statistics options will include:

1. Minimum: The lowest value in the neighborhood around the original cell.
2. Mean: The average value in the neighborhood.
3. Maximum: The maximum value in the neighborhood.
4. Range: The range of values (i.e. the maximum – the minimum) in the neighborhood.
5. Standard Deviation: The standard deviation of values in the neighborhood.
6. Variance: The variance of values in the neighborhood.
7. Sum: The sum of values in the neighborhood.

If your Grid Theme is an Integer theme, you have a few more options:

- a) Majority: The value that occurs most often in the neighborhood.
- b) Minority: The value that occurs least often in the neighborhood.
- c) Median: The median value in the neighborhood.

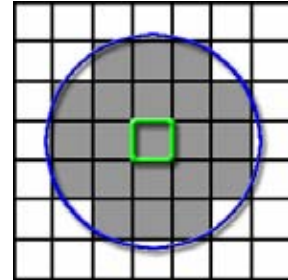


- d) Variety: The number of unique values in the neighborhood.

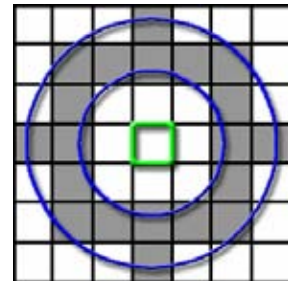
**Neighborhood Options:**

You will next be asked to define your neighborhood. You can choose between a circle, annulus (doughnut-shape), rectangle, wedge, or custom shape. You can also enter your neighborhood parameters in units of either grid cells or map units (i.e. meters, feet, etc.).

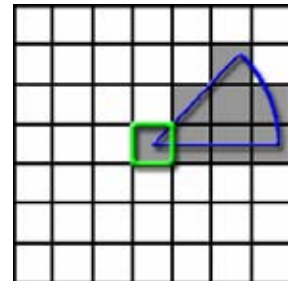
- a) Circle: A circular neighborhood defined by a radius length extending outward from the cell center. In theory, this neighborhood should be composed of all grid cells whose cell centers lie within that distance of the focal cell center, but Spatial Analyst appears to have some kind of number rounding issue that sometimes includes or excludes cells that do not exactly fit this definition. If possible, test the neighborhood function to make sure it is including the all the cells you need it to. If not, you may be able to use the Custom Neighborhood function to create a correct neighborhood.



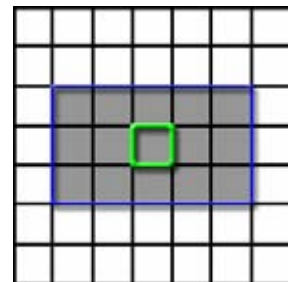
- b) Annulus: An annular neighborhood looks like a ring or doughnut defined by an inner and outer radius length extending outward from the cell center. This neighborhood should be composed of all cells whose cell centers lie within this ring, but this can also be affected by the rounding issues that affect the Circular neighborhood. If testing reveals that the annular neighborhood is not working exactly as desired, you may be able to generate a custom neighborhood that will work correctly.



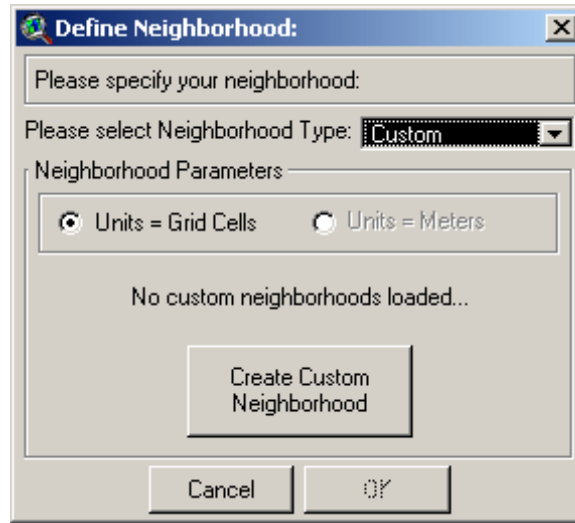
- c) Wedge: A wedge-shaped neighborhood looks like a slice of pie cut out of a circular neighborhood and is defined by a starting angle, and ending angle, and a radius. As always, it is advisable to test and confirm that your neighborhood is correct, and possibly generate a custom neighborhood if necessary.



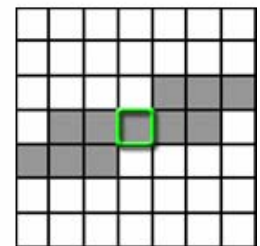
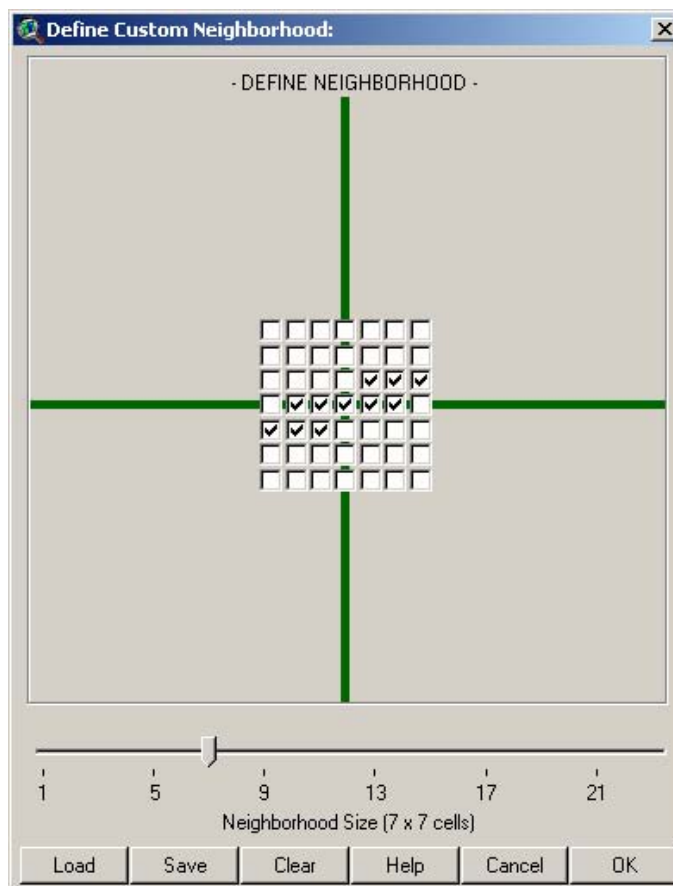
- d) Rectangle: A square or rectangular neighborhood defined by width and height, which will be centered around your focal cell center. Cells will be included in the neighborhood if the cell centers lie within this rectangle.



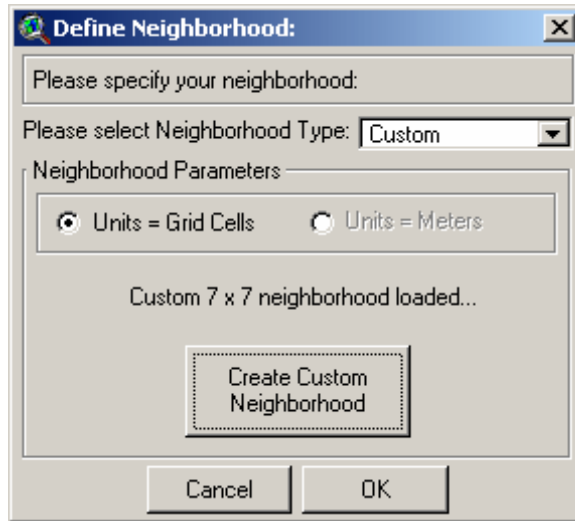
- e) Custom: Custom neighborhoods can be of any shape or arrangement and are defined by explicitly stating which cells, relative to the focal cell, will be included in the neighborhood. To generate a custom neighborhood, first select the “Custom” option from the drop-down list of Neighborhood Types:



Next click the “Create Custom Neighborhood” button to open the custom neighborhood definition window. Select your general neighborhood size by moving the slider to the correct value, and then click in the boxes to specify which cells you wish to be included in the analysis:



Click OK to go back to the “Define Neighborhood” window and notice that it now says “Custom 7 x 7 Neighborhood Loaded...”.



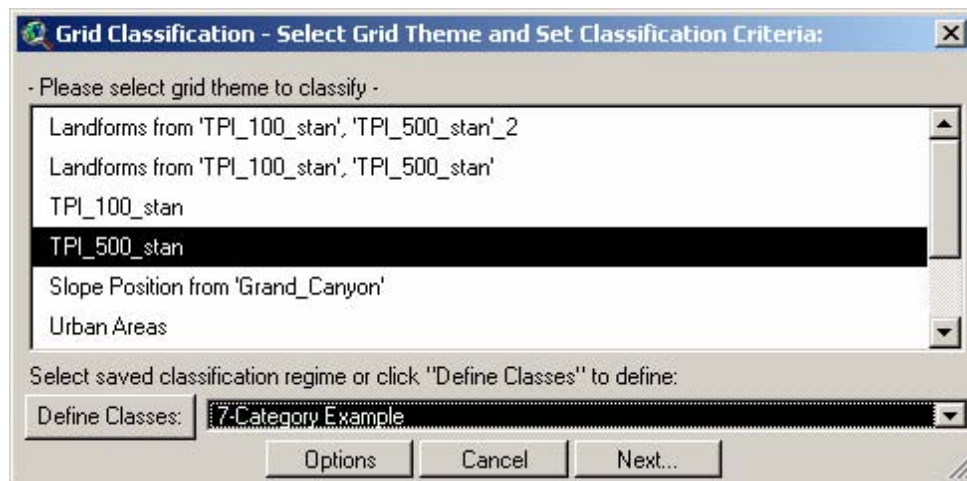
Click "OK" to generate your neighborhood statistics grid(s).

### GENERAL GRID CLASSIFICATION

**Classifying by Multiple Grids:** This extension is basically a tool to classify grids, focused on a particular manipulation and combination of elevation and slope values to attempt to identify topographic features. However, although the dialogs are designed to facilitate classification of slope position and landform based on Weiss' algorithms, they can be used more generally. For example, the Landform functions can be used to create a new classification grid based on any 2-grid combination (with options to add additional grids if necessary; see p. 22). Start by choosing the "Use Existing TPI Themes" option, and then select any two grids you want in the next dialog. If you have a classification based on a single grid, but which references a second grid for some classifications, use the Slope Position functions with the "Use Existing TPI Themes" option.

As you go through the dialogs, you may notice that they refer to "Slope Grid" and "TPI Grid" here and there, and the final output grid will be called "Slope Position" or "Landform", but those are just cosmetic issues intended to make the process more user-friendly. If you are classifying other types of grids, just ignore the Slope and Landform comments and remember to rename your final output grid.

**Classifying a Single Grid:** If you wish to classify a single grid based only on values within that grid, this extension does offer a separate function to help. Click the "Grid Classification" menu item in the "TPI" menu to open the initial dialog.



Begin by selecting the grid to classify, and the classification regime you wish to use.

Selecting and Defining a Classification Regime: Your classification regime is the set of rules determining how your grid values will be classified. You may use an existing criteria set or you may generate a new one. You may also load an existing one and modify it.

This extension comes with 2 general criteria sets pre-loaded. They are fairly generic and will probably be most useful as a template to modify. The first set has 4 classes and the second set has 7 classes. You can also save any classifications you create yourself.

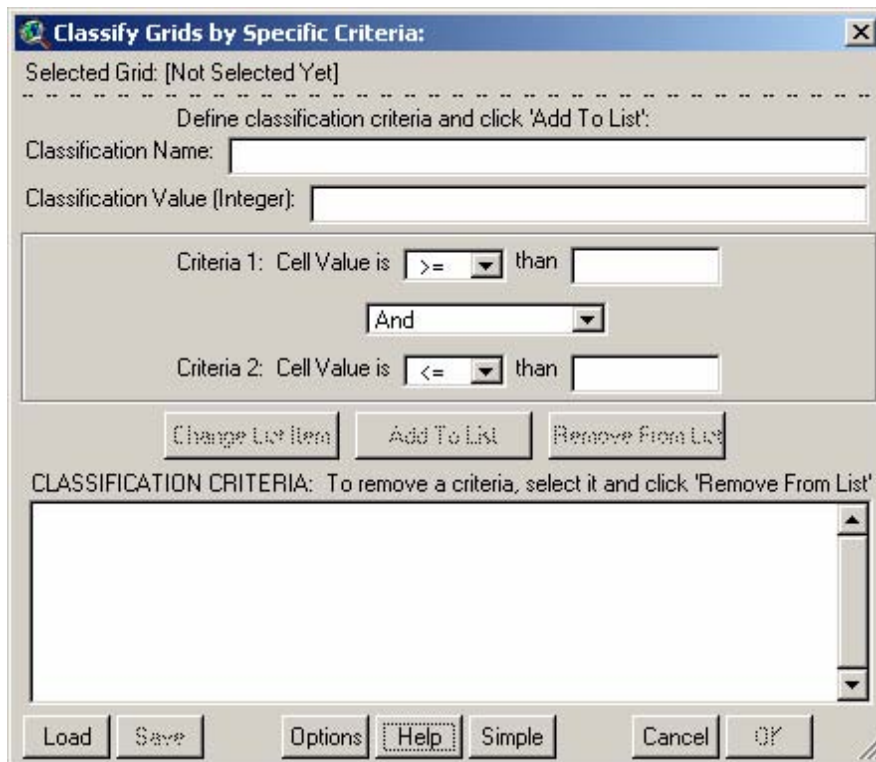
#### Sample 4-Class General Classification

**Value < -1:** [Value < -1]  
**Value Between -1 And 0:** [Value >= -1 and < 0]  
**Value Between 0 And 1:** [Value >= 0 and < 1]  
**Value > 1:** [Value >= 1]

#### Sample 7-Class General Classification

**Value < -10:** [Value < -10]  
**Value Between -10 And -5:** [Value >= -10 and < -5]  
**Value Between -5 And 0:** [Value >= -5 and < 0]  
**Value = 0:** [Value = 0]  
**Value Between 0 And 5:** [Value > 0 and < 5]  
**Value Between 5 And 10:** [Value >= 5 and < 10]  
**Value > 10:** [Value >= 10]

If this criteria set, or any other saved criteria set appearing in the drop-down list, is appropriate for your analysis, simply select it from the list and click the "Next" button to continue. If you need to modify the criteria set or create a new one, click the "Define Classes" button to open the "Classify Grids by Specific Criteria" dialog:



**Criteria Label and Value:** The “Classification Name” at the top is the text label for any particular classification. The Classification Value is the numeric value associated with that classification. Both of these values will appear in the final attribute table for the classification grid.

**IMPORTANT:** The classification values also determine the rank or precedence level of a classification in cases where you may have overlapping criteria. If multiple classification criteria reference the same values, then the classification with the lowest Classification Value will take precedence. For example, if 2 criteria both reference Grid A, and are defined as follows:

Criteria with Classification Value = 1; defined as  $[5 \leq \text{Grid A} < 10]$   
Criteria with Classification Value = 2; defined as  $[8 \leq \text{Grid A} < 15]$

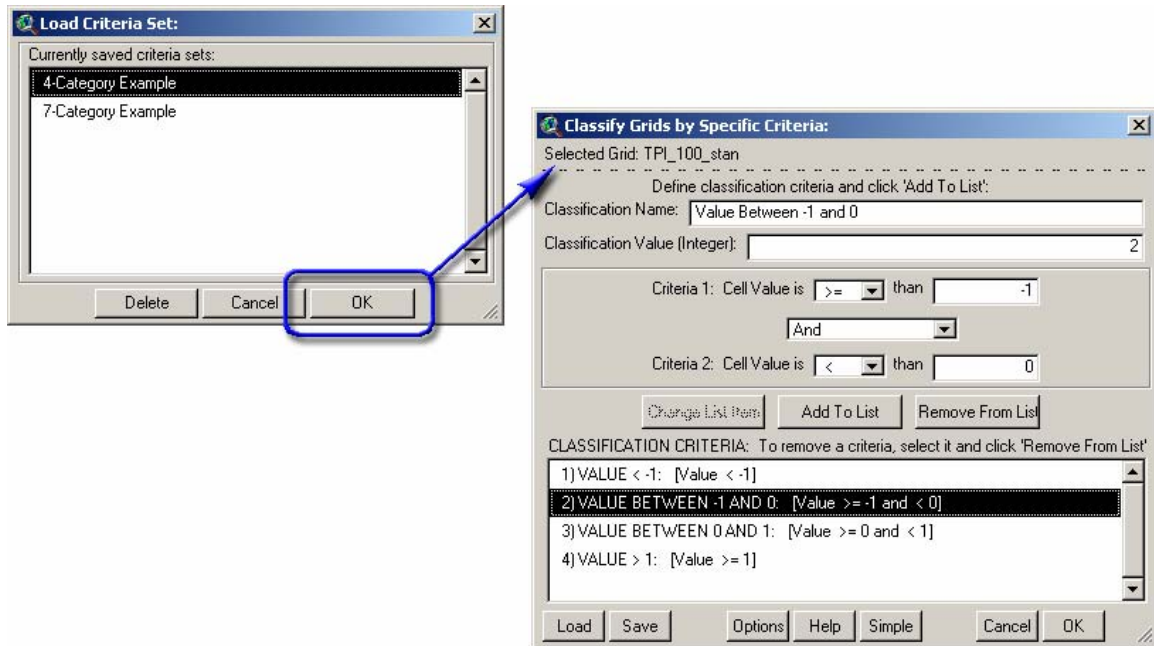
Then, all Grid A Cells with values 8 and 9 meet both criteria. Because the first criteria has the lowest Classification Value, then these cells would be classified as Criteria 1.

**Classification Criteria List:** The box at the bottom of the dialog labeled “CLASSIFICATION CRITERIA” lists all the classification definitions established for this criteria set. Assign the range of values that apply to this classification definition. Notice that you may opt to skip the second set of criteria if you only have one relationship that applies to this class (click the drop-down box containing the word “and” and select “Skip Criteria 2”). Click “Add to List” to add it to the definition for this class.

If you want to edit existing grid threshold values, select that grid in the “CLASSIFICATION CRITERIA” listbox. The current threshold values will appear above, so make any changes you wish and click “Change List Item” to save the changes.

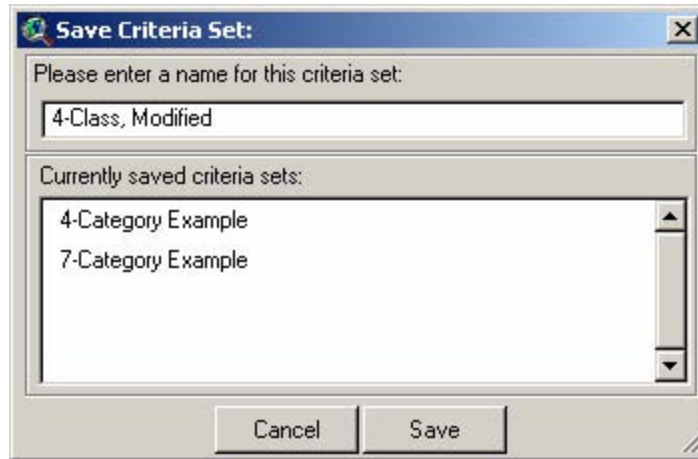
You may remove any existing grid threshold values by selecting it in the “CLASSIFICATION CRITERIA” listbox and clicking the “Remove from List” button.

**Loading Criteria Sets:** You can load pre-existing or saved criteria sets by clicking the “Load” button. Simply select the criteria set to load and click “OK”:



**Saving Criteria Sets:** If you expect to use your criteria set over again or wish to share it with others, it may make sense to save it so it can be distributed and easily reloaded. Click the “Save” button to open the “Save Criteria Set” dialog:



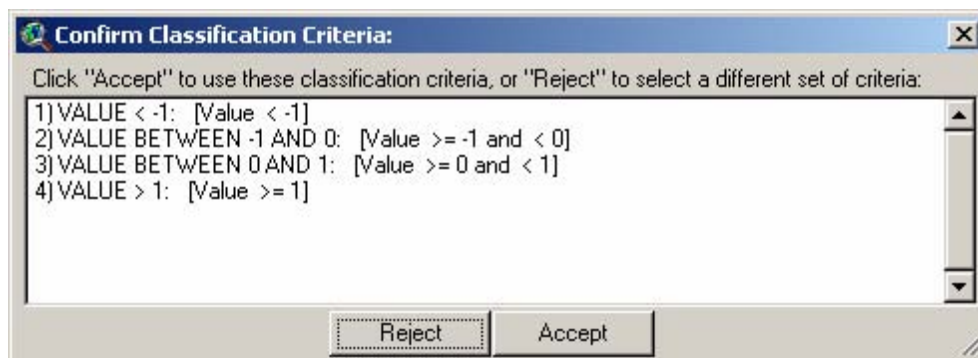


Give your new criteria set a name and click “Save”, and it will be saved to the “General Classification Saved Criteria Set” table (./tpi\_class\_criteria.dbf). See the discussion on Saved Criteria Tables (p. 37) for more information on this table and how to share saved criteria sets.

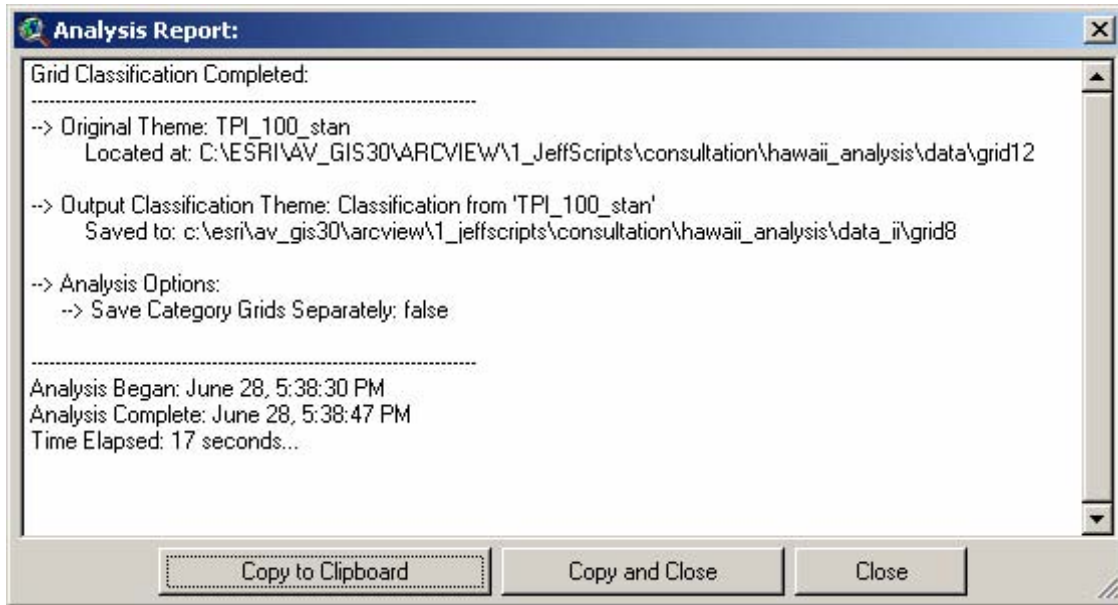
Options: The “Options” button allows you to save the individual category grids separately, as well as generating the new classification grid.



After you have selected or created your classification regime, click the “Next...” button on the main dialog and confirm your selected classification definitions:



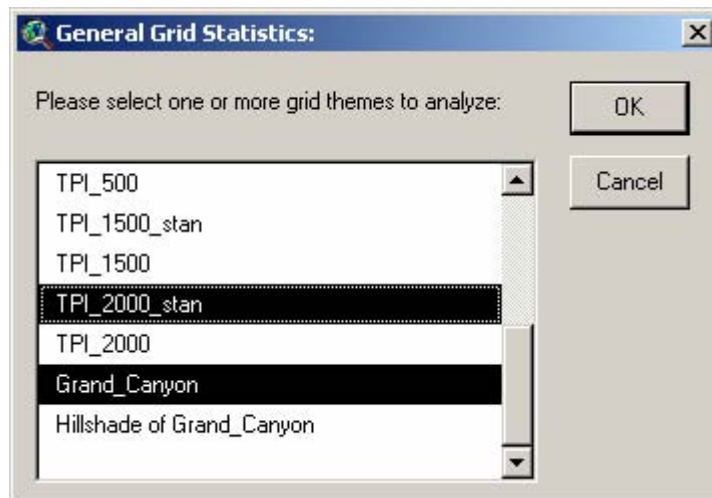
Upon completion, the extension will add your new Landform grid to your view and generate a report detailing exactly what it did:



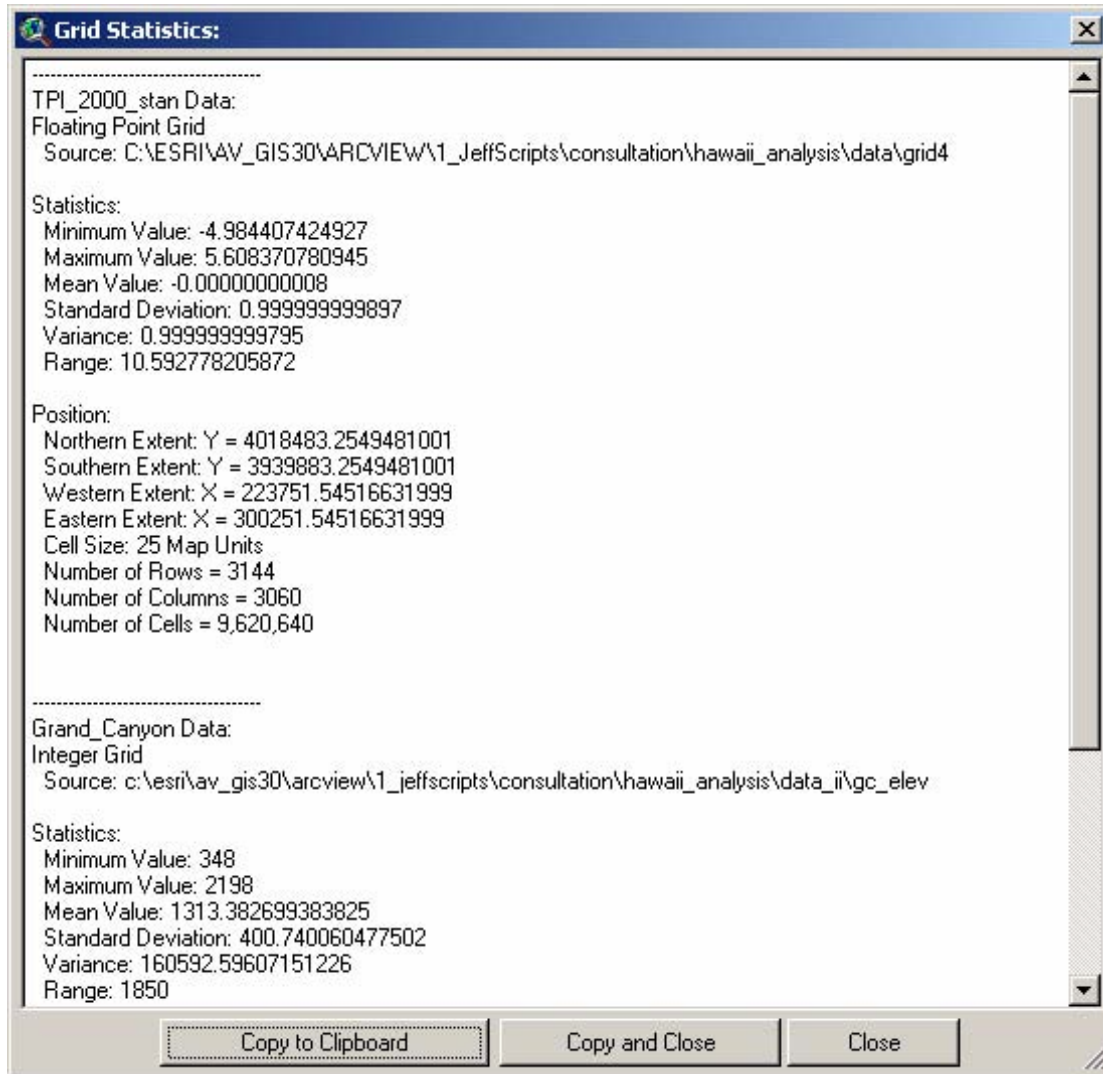
### GENERAL GRID STATISTICS

This function produces a report detailing some general statistics for grid themes, including the *minimum, maximum, mean, standard deviation, variance* and *range* for all grid cell values in the grid. The report also describes the geographic extent, the number of rows, columns and cells, and the hard drive location of the grid.

Select the "Grid Descriptive Statistics" menu item and then select the grids you would like to describe:



Click "OK" and you will see a report window with the grid description:



### STANDARDIZING GRIDS

In this case, to standardize a grid means to transform the grid cell values such that the grid has a mean = 0 and a standard deviation = 1. This may be useful in some analyses where the grids may have widely different numeric scales. For example, Andrew Weiss, in his discussion of Topographic Position Indices, describes how it may be desirable to standardize TPI grids before using them for landform identification. This process is described in a poster presented at the 2001 ESRI User Conference.

Technically, all that happens in this operation is that the original grid is subtracted from the grid mean, then divided by grid standard deviation:

$$\text{Standardized Grid} = \frac{\text{Original Grid} - \text{Grid Mean}}{\text{Grid Standard Deviation}}$$

## SAVED CRITERIA SETS

### DISCUSSION

This extension is essentially a tool to classify data from 1 or more grids into a single categorical class grid. This can always be done manually using a series of Map Calculator and Map Query functions, but that process can be very time-consuming, error-prone and painful to replicate. Therefore the author feels that the real power of this extension is the ability to save, re-use and share classification criteria sets.

In order to make the classification criteria sets transportable, this extension stores those criteria in dBASE tables. The extension generates default tables the first time you run it, and each default table includes one or more pre-loaded classification sets. There are 3 default tables:

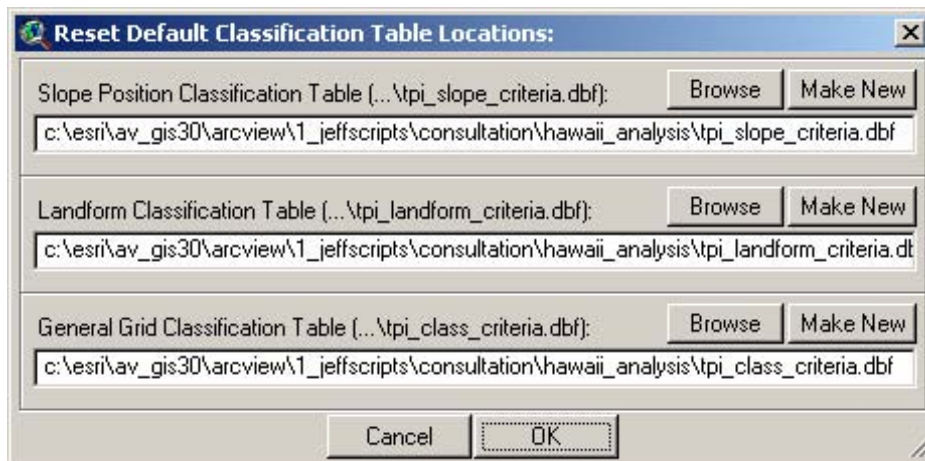
1. **tpi\_slope\_criteria.dbf**: Stores classification criteria for the Slope Position classification functions (see p. 12). This table comes preloaded with Weiss' 6-class Slope Position Classification and Dickson & Beier's 4-class Slope Position Classification.
2. **tpi\_landform\_criteria.dbf**: Stores classification criteria for the Landform classification functions (see p. 19). This table comes preloaded with Weiss' 10-class Landform Classification.
3. **tpi\_class\_criteria.dbf**: Stores classification criteria for general single-grid classification functions (see p. 31). This table comes preloaded with a based 4-Class and a basic 7-Class Classification.

### SETTING DEFAULT TABLE LOCATIONS

The first time you run the extension, it generates the 3 default tables in the current work directory and then remembers where they are located (see "Technical Notes" for details on how it remembers things). If it ever expects to find a default table in a particular location, and that table does not exist, then it regenerates the default table in the work directory.

At any point you can reset the default location to a different table. The table must have the correct name (**tpi\_slope\_criteria.dbf**, **tpi\_landform\_criteria.dbf** or **tpi\_class\_criteria.dbf**), but it can be located anywhere.

To reset the default locations, click the "Reset Default Classification Tables" menu item in the "TPI" menu to open the "Reset Locations" dialog:



This dialog allows you to browse for different tables or to create new default ones. Note that you must use the Browse button to find new criteria tables because the text boxes are set to "read-only".

## SHARING CRITERIA SETS

The default criteria tables are dBASE tables which may be moved around like any other file. The person who receives the file only needs to use the “Reset Default Classification Table Locations” dialog to reference the new table.

## TECHNICAL DATA: FIELDS AND VALUES

(Precision numbers should be read as “Field Width” – dot – “Decimal Precision”)

### tpi\_slope\_criteria.dbf (21 fields):

- |  |   |   |
|--|---|---|
| 1) SAVE_NAME: String, Char<br>[#FIELD_CHAR; Precision = 200.0]       | [#FIELD_CHAR; Precision = 5.0]  | 16) CRIT_SLOP2: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 20.8] |
| 2) DESCRIPT: String, Char<br>[#FIELD_CHAR; Precision = 50.0]         | 9) CRITERIA_2: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 20.8]  | 17) USE_OTHER: String, Logical<br>[#FIELD_LOGICAL; Precision = 5.0]   |
| 3) UNITS: String, Char<br>[#FIELD_CHAR; Precision = 20.0]            | 10) UNITS_2: String, Char<br>[#FIELD_CHAR; Precision = 16.0]          | 18) OTHER: String, Char<br>[#FIELD_CHAR; Precision = 254.0]           |
| 4) OP_1: String, Char<br>[#FIELD_CHAR; Precision = 5.0]              | 11) USESLOPE: String, Logical<br>[#FIELD_LOGICAL; Precision = 5.0]    | 19) FIRST_ROW: String, Char<br>[#FIELD_CHAR; Precision = 6.0]         |
| 5) CRITERIA_1: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 20.8] | 12) OP_SLOPE: String, Char<br>[#FIELD_CHAR; Precision = 5.0]          | 20) GROUP: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 11.0]      |
| 6) UNITS_1: String, Char<br>[#FIELD_CHAR; Precision = 16.0]          | 13) CRIT_SLOPE: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 20.8] | 21) ORDER: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 6.0]       |
| 7) AND_OR: String, Char<br>[#FIELD_CHAR; Precision = 25.0]           | 14) SL_AND_OR: String, Char<br>[#FIELD_CHAR; Precision = 25.0]        |   |
| 8) OP_2: String, Char  | 15) OP_SLOPE2: String, Char<br>[#FIELD_CHAR; Precision = 5.0]         |   |

### tpi\_landform\_criteria.dbf (22 fields):

- |   |  |   |
|---|--|---|
| 1) SAVE_NAME: String, Char<br>[#FIELD_CHAR; Precision = 200.0]      | [#FIELD_DECIMAL; Precision = 20.8]                                   | 16) TPI2_UNIT2: String, Char<br>[#FIELD_CHAR; Precision = 16.0]     |
| 2) DESCRIPT: String, Char<br>[#FIELD_CHAR; Precision = 50.0]        | 9) TPI1_UNIT2: String, Char<br>[#FIELD_CHAR; Precision = 16.0]       | 17) USE_OTHER: String, Logical<br>[#FIELD_LOGICAL; Precision = 5.0] |
| 3) TPI1_OP_1: String, Char<br>[#FIELD_CHAR; Precision = 5.0]        | 10) TPI2_OP_1: String, Char<br>[#FIELD_CHAR; Precision = 5.0]        | 18) OTHER: String, Char<br>[#FIELD_CHAR; Precision = 254.0]         |
| 4) TPI1_VAL1: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 20.8] | 11) TPI2_VAL1: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 20.8] | 19) FIRST_ROW: String, Char<br>[#FIELD_CHAR; Precision = 6.0]       |
| 5) TPI1_UNIT1: String, Char<br>[#FIELD_CHAR; Precision = 16.0]      | 12) TPI2_UNIT1: String, Char<br>[#FIELD_CHAR; Precision = 16.0]      | 20) GROUP: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 11.0]    |
| 6) TPI1_A_OR: String, Char<br>[#FIELD_CHAR; Precision = 25.0]       | 13) TPI2_A_OR: String, Char<br>[#FIELD_CHAR; Precision = 25.0]       | 21) ORDER: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 6.0]     |
| 7) TPI1_OP_2: String, Char<br>[#FIELD_CHAR; Precision = 5.0]        | 14) TPI2_OP_2: String, Char<br>[#FIELD_CHAR; Precision = 5.0]        | 22) UNITS: String, Char<br>[#FIELD_CHAR; Precision = 20.0]          |
| 8) TPI1_VAL2: Number, Decimal                                       | 15) TPI2_VAL2: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 20.8] |   |

### tpi\_class\_criteria.dbf 11 fields):

- |   |  |  |
|---|--|--|
| 1) SAVE_NAME: String, Char<br>[#FIELD_CHAR; Precision = 200.0]  | 5) CRITERIA_1: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 20.8] | 9) FIRST_ROW: String, Char<br>[#FIELD_CHAR; Precision = 6.0]     |
| 2) NAME: String, Char<br>[#FIELD_CHAR; Precision = 50.0]        | 6) AND_OR: String, Char<br>[#FIELD_CHAR; Precision = 25.0]           | 10) GROUP: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 11.0] |
| 3) VALUE: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 11.0] | 7) OP_2: String, Char<br>[#FIELD_CHAR; Precision = 5.0]              | 11) ORDER: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 6.0]  |
| 4) OP_1: String, Char<br>[#FIELD_CHAR; Precision = 5.0]         | 8) CRITERIA_2: Number, Decimal<br>[#FIELD_DECIMAL; Precision = 20.8] |  |



## TROUBLESHOOTING:

If you encounter some strange crash, please click the menu item "Check 'TPI Extension' Scripts" in either the View, Table or Project Help menu. With any luck, that function will generate a report with enough information for the author to find and fix the problem.

Otherwise, the problem may be found and explained below:

---

**Problem:** Extension Fails to Load, with the following error message:

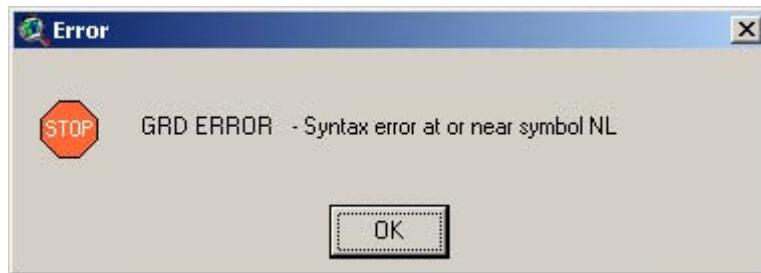


**Solution:** This problem is caused by an outdated version of the Dialog Designer. For some reason, some versions of ArcView 3 were shipped with an older version of Dialog Designer which didn't support this "LISTBOX\_SELECTION\_MULTIRROW" option (which basically means that a listbox on a dialog is set so that you can select multiple items from the list).

ESRI has a newer version of the Dialog Designer available on their website for free download. Please link to:

<http://support.esri.com/index.cfm?fa=downloads.patchesServicePacks.viewPatch&PID=25&MetalD=483>

**Problem:** Extension crashes in mid-operation, producing an obscure message stating that there is a syntax error at or near symbol NL:



This is sometimes followed by the infamous Segmentation Violation message:



Sometimes ArcView crashes completely and vanishes without showing these messages; while other times it vanishes after showing these messages.

**Solution:** There is no simple solution to this problem. It is due to a bug in Spatial Analyst which causes SA to crash after approximately 32,500 grid operations or if SA tries to hold > 50 grids in memory at one time. You can force the error to occur by writing a short script that checks the cell value at a particular point, then loops over 32,500 iterations. You can also trigger it by running the Zonal Statistics function on a point theme containing over 32,500 points. You would probably trigger this error if you tried to classify grids using > 50 classification values because it creates a separate grid for each classification and then merges them for the final output.

I am unaware of a simple way to work around this problem. If possible, use smaller point data sets or try to use fewer grids in your analysis. Alternatively, I hear that ArcGIS 9 was expected to fix this problem.

**Problem:** Unable to find grid in a directory, even though you know it is there.

**Solution:** This is probably due to a space or invalid character in the pathname. For some reason, Spatial Analyst doesn't recognize a grid if it lies in a folder with a space or period in it. For example, if you store your grids in the standard default Windows directory "My Documents," you will probably not see the grid listed in the "Add Theme" dialog at all. The "Add Theme" dialog will show you all the shapefiles and images, but no grids. The only solution is to move your grid to a different file location where it does not lie in a path with invalid characters.

**Problem:** You load your grids but you are unable to do any calculations on them. They don't act like grids.

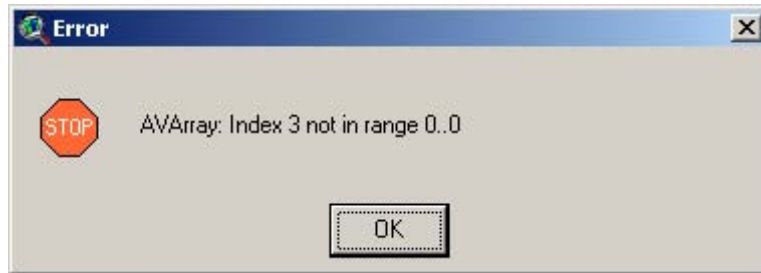
**Solution:** You may have loaded them as images. Grids can be loaded as either images or grids, and you can't do any of the grid functions on them if you have loaded them as images. Try adding them to your view again, but make sure that you have "Grid Data Source" selected instead of "Image Data Source".

**Problem:** Extension crashes in mid-calculation, with the message:



**Solution:** This error may be caused by either a corrupt INFO directory or if your working directory pathname is too long. I am unaware of the exact pathname size that triggers the error, but I think it is around 80 characters. If you have over 80 characters in your pathname and you see this error, then you can probably avoid it by changing your work directory to someplace closer to the root.

**Problem:** Extension crashes when you try to load a saved criteria set, with the message:



**Explanation:** This error can occur if you have classifications in which a single classification requires examination of a large number of grids. This would only be triggered if you saved the classifications and attempted to load them to a new analysis. The problem here is that the dBASE table that holds the saved classification criteria definitions has limitations regarding string fields, in which they can only hold  $\leq 254$  characters. The definition for each classification value is saved as a string, and large numbers of grids will cause this string to exceed 254 characters. If the string exceeds 254 characters, then the string will be clipped to the first 254 characters and almost certainly corrupted so that it is unusable.

**Solution:** I have not thought of a reasonable solution to this problem yet.

#### TECHNICAL NOTES:

This extension adds a new document GUI to your ArcView project called "Placeholder" and sets it to be invisible. It is a clone of the Script document GUI, but only serves as a memory-persistent anchor to attach several items that the author wanted the extension to remember. By attaching them as object tags to the Placeholder GUI, the extension will remember them between ArcView sessions. These items are:

1. The file locations of the 3 default data tables **tpi\_class\_criteria.dbf**, **tpi\_landform\_criteria.dbf** and **tpi\_slope\_criteria.dbf**. These are the data tables that contain the saved classification criteria sets. The filenames for these 3 files are attached as an object tag to the Placeholder GUI menu bar.
2. The last-used settings for the *Slope Position* and *Landform Classification* options are attached to the Placeholder GUI tool bar.
3. The last-used Neighborhood parameters for the Neighborhood Statistics function are attached to the Placeholder GUI button bar.
4. The definition of the last custom neighborhood is attached to the Placeholder GUI Save button.
5. The last-used settings for the *General Grid Classification* options are attached to the Placeholder GUI Cut button.

The Placeholder GUI is invisible and no actual "Placeholder" documents are ever created by this extension, but you will see the Placeholder GUI listed if you open up the "Customize" dialog and then click the "Edit" button.

#### REFERENCES:

- Dickson, B. and P. Beier. In Review. Influence Of Topographic Position On Cougar Movement In Southern California.
- Guisan, A., S. B. Weiss and A. D. Weiss. 1999. GLM versus CCA spatial modeling of plant species distribution. Kluwer Academic Publishers. Plant Ecology. 143:107-122.

Jones , K.B., D.T. Heggem, T.G. Wade, A.C. Neale, D.W. Ebert, M.S. Nash, M.H. Mehaffey, K.A. Hermann, A.R. Selle, S. Augustine, I.A. Goodman, J. Pedersen, D. Bolgrien, J.M. Viger, D. Chiang, C.J. Lin, Y. Zhong, J. Baker And R.D. Van Remortel. 2000. Assessing Landscape Conditions Relative to Water Resources in the Western United States: A Strategic Approach. Environmental Monitoring and Assessment 64: 227 – 245.

Weiss, A. 2001. Topographic Position and Landforms Analysis. Poster presentation, ESRI User Conference, San Diego, CA.

---

Enjoy! Please contact the author if you have problems or find bugs.

Jeff Jenness  
Jenness Enterprises  
3020 N. Schevene Blvd.  
Flagstaff, AZ 86004  
USA

[jeffj@jennessent.com](mailto:jeffj@jennessent.com)  
<http://www.jennessent.com>  
(928) 607-4638

Updates to this extension and an on-line version of this manual are available at

<http://www.jennessent.com/arcview/tpi.htm>

---

Please visit *Jenness Enterprises* [ArcView Extensions](#) site for more ArcView Extensions and other software by the author. We also offer customized ArcView-based [GIS consultation](#) services to help you meet your specific data analysis and application development needs.

