Menus in SML Dialogs using XML

SML dialog windows set up using an XML specification can include custom menus with your own set of menu options and associated actions. The menu can be shown on the window by the menu name (Point Action ... menu in the sample window shown here) or by a designated icon (File

menu in the sample). The complete XML specification for this sample dialog window is shown below.

■Process Select	ed Point 🔤 🗆 🗙	
Point Action		
Buffer Distance	200	
0	OK Cancel	

■Process Selected Point <mark></mark>	Process Selected Point
Point Action	Point Action
Close Text File Cancel	Buff Save Coordinates to Text File Compute Buffer Zone
	Find Enclosing Polygon

```
<?xml version="1.0"?>
<root>
   <dialog id="procpts" title="Process Selected Points" OnOK="OnOK()">
      cyane Orientation="Horizontal"> // start-tag for the first menubutton element with icon to use for the button
         <menubutton id="file" Icon="DESKTOP FILE" ToolTip="Text File..." OnSelection="OnFile()">
            <item Value="open">Open Text File</item>
                                                           item list for the selections on the first menu
            <item Value="close">Close Text File</item>
                                        >start-tag for the second menubutton element with name to use for the button
         </menubutton>
         <menubutton id="ptaction" Name= "Point Action..." OnSelection="OnAction()">
            <item Value="save">Save Coordinates to Text File</item>
            <item Value="buffer">Compute Buffer Zone</item>
                                                                     item list for the selections on the second menu
            <item Value="polygon">Find Enclosing Polygon</item>
         </menubutton>
      </pane>
      <pane Orientation="Horizontal">
         <label>Buffer Distance:</label> label for the buffer distance editnumber field
         <editnumber id="buffdist"_MinVal="0" MaxVal="1000" Precision="0" OnActivate="OnBuffDist()"/>
      </pane>

    numeric field for entering buffer distance

   </dialog>
</root>
```

Strict Syntax Checking in SML

The geospatial scripting language (SML) in the TNT products provides a level of syntax checking to ensure correct interpretation of complex code prevent inadvertent overwriting of variable values. When you check the sy of an SML script or query, warnings of violations of the following highsyntax rules are automatically shown:

- 1) All variables must be declared before they are used in a statement.
- 2) Assigned variable values must match the declared variable type.
- 3) All statements must end in a semicolon (";").

Violations of these rules are not interpreted as errors, and will not prevent sc

from running. Warnings are presented in a separate Script Warnings window that appears only if no lower-level syntax errors were detected. Strict syntax checking is invoked automatically when you choose Syntax / Check from the SML Editor or Query Editor window, but is not invoked when you run the script. The \$warnings keyword is available from the Keyword list in the Script Reference window and can be inserted into a script to specify a lower level of syntax checking if desired.

Spatial nanipulation Language
File Edit Insert Syntax Help
applynsk.sml
\$warnings 3
<pre>GetInputRaster(Input); numlins = Numlins(Input); numcols = NumCols(Input); rasttype\$ = RastType(Input);</pre>
GetInputRaster(Mask,numlins,numcols);
<pre>GetOutputRaster(Output,numlins,numcols,rasttype\$);</pre>
<pre>for each Input begin a = Input; b = Mask; c = 0 if (b == 1) then c = a Output = c end</pre>
used without being declaredassuning 'raster'. Is used without being declaredassuning 'numeric'. Is used without being declaredassuning 'numeric'. Used without being declaredassuning 'raster'. It used without being declaredassuning 'raster'. It used without being declaredassuning 'raster'. It used without being declaredassuning 'numeric'. It wi

Close