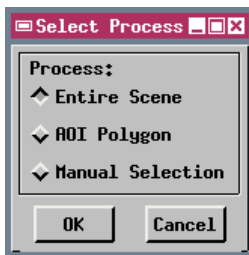


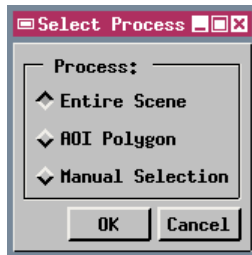
Creating SML Dialogs

The Spatial Manipulation Language (SML) in the TNT products provides two ways for you to create custom dialog windows for your SML scripts. To create dialogs that work interchangeably in SML in the X Windows environment (SML/X) or in SML for Windows (SML/W), you can use a simple set of tags and attributes in XML-format. The dialog specification in XML is interpreted by SML and mapped to a set of generic GUI dialog classes that utilize either Motif dialog components in SML/X or Windows dialog components in SML/W. You can also build dialogs directly using Motif classes, but they are only useable in SML/X. These subsections of SML scripts show examples of the two dialog coding methods for a simple dialog window with a set of radio buttons inside a groupbox (frame).

Using Motif classes



Using XML



Generic GUI Dialog via XML (for SML/X or SML/W)

```
xml$='<?xml version="1.0"?>
<root>
  <dialog id="select" title="Select Process" OnOK="OnOK()">
    <groupbox Name=" Process: " ExtraBorder="4">
      <radiogroup id="processgp" Default="entire">
        <item Value="entire" Name="Entire Scene"/>
        <item Value="polygon" Name="AOI Polygon"/>
        <item Value="manual" Name="Manual Selection"/>
      </radiogroup>
    </groupbox>
  </dialog>
</root>';
```

XML string variable specifying the dialog layout and components. Alternatively the XML code can be read from a separate text file.

```
class XMLDOC dlgdoc;
err = dlgdoc.Parse(xml$);
if (err < 0) {
  PopupError(err);
  Exit();
}
```

parse XML text string into memory. If there are XML syntax errors, show an error dialog; pressing "Details" button on the error dialog lists the XML syntax errors.

```
class XMLNODE dlgnode;
dlgnode = dlgdoc.GetElementByID("select");
if (dlgnode == 0) {
  PopupMessage("Could not find dialog node in XML document");
  Exit();
}
```

get the dialog element from the parsed XML
show an error message if the dialog element not found

```
class GUI_DLG dlgwin;
dlgwin.SetXMLNode(dlgnode);
ret = dlgwin.DoModal();
```

set the XML dialog element as the source for the GUI_DLG class instance being used for the dialog window; open window as a modal dialog

NOTE: procedure definitions not included in the script extracts. More callback procedures are required for the Motif method in this example.

X Windows Dialog via OSF/Motif (for SML/X only)

```
class XmForm dlgwin, frameform, btnrow;
class XmToggleButton btn1, btn2, btn3;

dlgwin = CreateFormDialog("Select Process");
dlgwin.MarginHeight = 5;
dlgwin.MarginWidth = 5;

class XmFrame frame;
frame = CreateFrame(dlgwin);
frame.TopWidget = dlgwin;
frame.LeftWidget = dlgwin;
frame.RightWidget = dlgwin;
frame.ShadowType = "SHADOW_ETCHED_IN";

frameform = CreateForm(frame);
frameform.TopWidget = frame;
frameform.LeftWidget = frame;
frameform.RightWidget = frame;
frameform.MarginWidth = 4;

class XmLabel label;
label = CreateLabel(frameform, "Process:");
label.TopWidget = frameform;
label.LeftWidget = frameform;
label.RightWidget = frameform;
```

create dialog

create box around other components

create container inside box to hold other components

create text label

```
btn1 = CreateToggleButton(frameform, "Entire Scene");
btn1.TopWidget = label;
btn1.LeftWidget = frameform;
btn1.InidicatorType = "ONE_OF_MANY";
btn1.Set = 1;
WidgetAddCallback(btn1.ValueChangedCallback, btn1Set);

btn2 = CreateToggleButton(frameform, "AOI Polygon");
btn2.TopWidget = btn1;
btn2.LeftWidget = frameform;
btn2.TopOffset = 4;
btn2.InidicatorType = "ONE_OF_MANY";
WidgetAddCallback(btn2.ValueChangedCallback, btn2Set);

btn3 = CreateToggleButton(frameform, "Manual Selection ");
btn3.TopWidget = btn2;
btn3.LeftWidget = frameform;
btn3.BottomWidget = frameform;
btn3.TopOffset = 4;
btn3.BottomOffset = 4;
btn3.InidicatorType = "ONE_OF_MANY";
WidgetAddCallback(btn3.ValueChangedCallback, btn3Set);

class PushButtonItem okBtn, cancelBtn;
okBtn = CreatePushButtonItem("OK", OnOK);
cancelBtn = CreatePushButtonItem("Cancel", OnCancel);
btnrow = CreateButtonRow(dlgwin, okBtn, cancelBtn);
btnrow.TopWidget = frame;

DialogOpen(dlgwin);
DialogWaitForClose(dlgwin);
```

create toggle button 1

create toggle button 2

create toggle button 3

create pushbuttons at bottom of window

open the dialog window