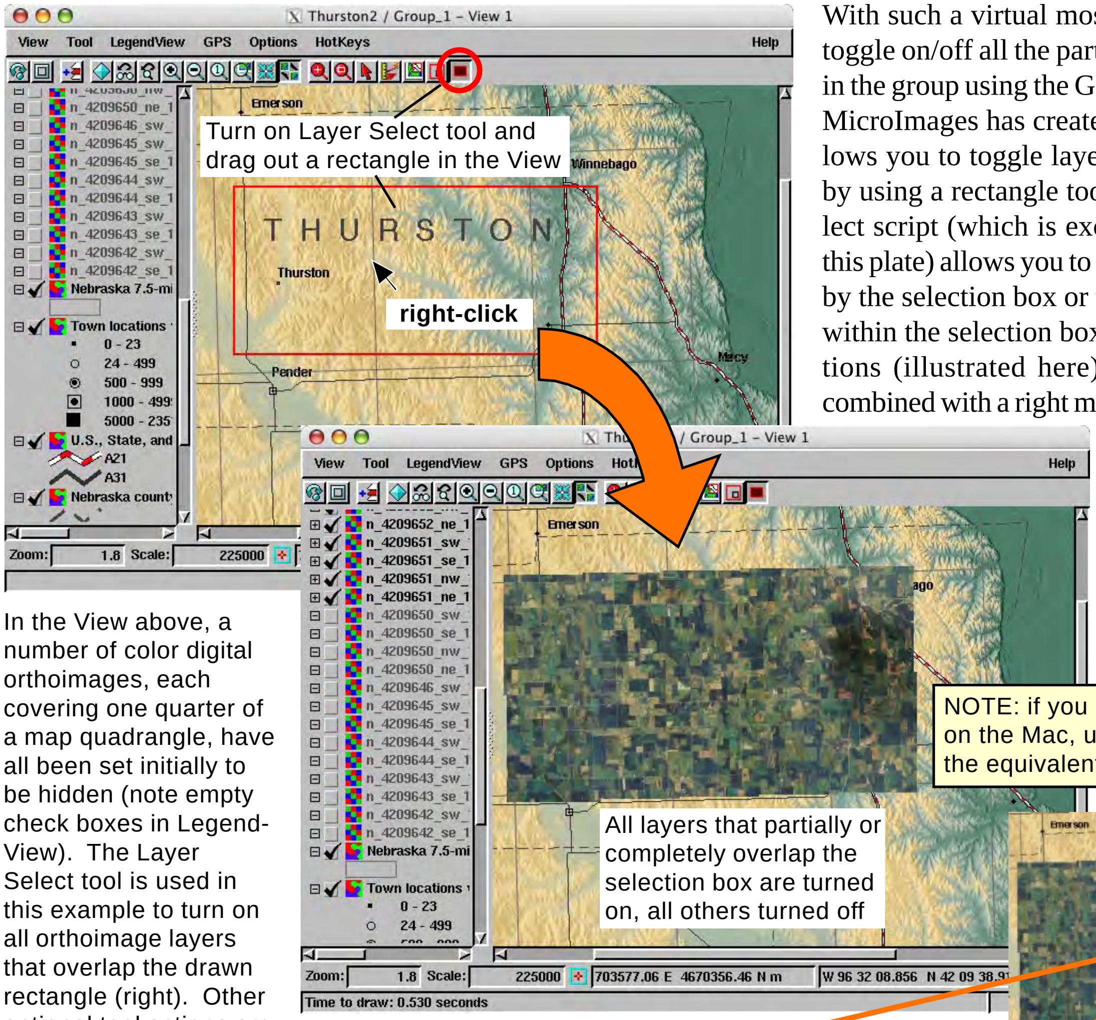
Sample Tool Script

Control Display by Area of Interest

A TNTmips display group can include a “virtual mosaic” of tiled geodata layers, with each of the many (even thousands of) individual layers covering only a portion of the group area. Most commonly the “tiles” would represent data layers such as elevation models, orthoimages, or vector map layers derived from individual map quadrangles. In the example illustrated here, hundreds of color orthoimage tiles (each covering one quarter of a map quadrangle) are arrayed in a display group that also includes several lower-resolution map layers that cover the entire area.

With such a virtual mosaic, it can be cumbersome to toggle on/off all the particular layers in a specific area in the group using the Group Controls or LegendView. MicroImages has created a sample tool script that allows you to toggle layers on or off on a spatial basis by using a rectangle tool in the View. The Layer Select script (which is excerpted on the reverse side of this plate) allows you to toggle on all layers overlapped by the selection box or toggle off all layers contained within the selection box. These and other variant actions (illustrated here) are initiated by keystrokes combined with a right mouse click. The script is implemented to work with any type of object layer, but it also contains sample code to show how the toggle action can be restricted to a particular object type (such as raster layers).

NOTE: if you are using a one-button mouse on the Mac, use ⌘ + click in the View as the equivalent of a right mouse click.

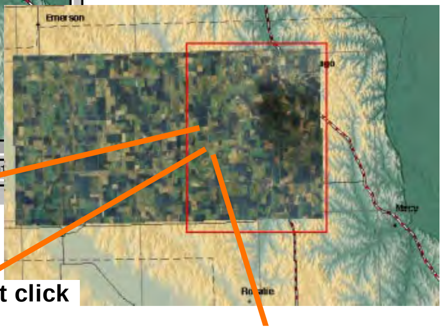


Turn on Layer Select tool and drag out a rectangle in the View

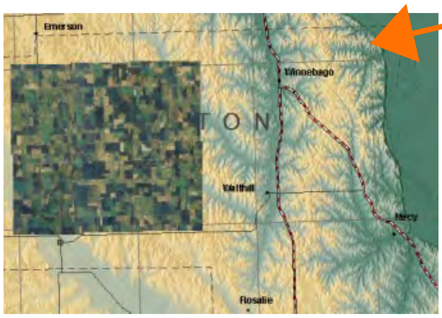
right-click

All layers that partially or completely overlap the selection box are turned on, all others turned off

In the View above, a number of color digital orthoimages, each covering one quarter of a map quadrangle, have all been set initially to be hidden (note empty check boxes in LegendView). The Layer Select tool is used in this example to turn on all orthoimage layers that overlap the drawn rectangle (right). Other optional tool actions are illustrated below.

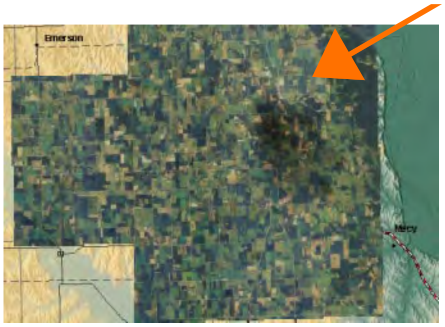


Ctrl + right click



All layers that are completely enclosed in the selection box are turned off. All other layers remain in their previous state.

Shift + right click



All layers that overlap the selection box are turned on. All other layers remain in their previous state.

Ctrl + Shift + right click

All layers are turned on.

Many sample scripts have been prepared to illustrate how you might use the features of the TNT products’ scripting language for scripts and queries. These scripts can be downloaded from www.microimages.com/freestuf/scripts.htm.

Script Excerpts for Layer Toggle Tool Script (LayerSelTool.sml)

Procedure to compare two regions and set the values of overlapping and enclosed to true or false based on the results of the comparison.

```
proc compareRegions(class REGION2D selR, class REGION2D layR) {
    if ((layR.Extents.x1 < selR.Extents.x2) && (layR.Extents.x2 > selR.Extents.x1))
        {if ((layR.Extents.y1 < selR.Extents.y2) &&
            (layR.Extents.y2 > selR.Extents.y1)) {
            overlapping = true;
        } else {
            overlapping = false;
        }
    } else {
        overlapping = false;
    }

    if ((layR.Extents.x1 > selR.Extents.x1) && (layR.Extents.x2 < selR.Extents.x2))
        {if ((layR.Extents.y1 > selR.Extents.y1) &&
            (layR.Extents.y2 < selR.Extents.y2)) {
            enclosed = true;
        } else {
            enclosed = false;
        }
    } else {
        enclosed = false;
    }
}
```

Procedure to transform the extents of the tool and the layer to View coordinates and then call compareRegions() to compare them.

```
proc checkOverlap(class GRE_LAYER testLayer) {
    local class REGION2D testRegion = tool.RegionData;
    local class REGION2D testLayerReg = testLayer.MapRegion;
    local class TRANSPARM tempTrans;

    tempTrans = ViewGetTransViewToScreen(View, true);
    testRegion = RegionTrans(testRegion, tempTrans);

    tempTrans =
        ViewGetTransMapToView(View, testLayer.MapRegion.CoordRefSys);
    testLayerReg = RegionTrans(testLayerReg, tempTrans);

    compareRegions(testRegion, testLayerReg);
}
```

Function to return whether or not the current layer matches the specifications the user wants. In its current form, it only makes sure the layer has a type. The commented-out section is an example of how to specify what type of layers the user wants the script to work with.

```
func checkLayer(class GRE_LAYER checkLayer) {
    if (checkLayer.Type == "") {
        PopupMessage("Layer has no Type");
        return (false);
    }
#   if (checkLayer.Type == "Raster") {
#       if (checkLayer.Name.indexOf("Tile_r", 0) != -1 ) {
#           return (true);
#       }
#   }
#   return (false);
    return true;
}
```

Procedure called when the Right Mouse button is clicked after using the tool to create a rectangle. It calls the appropriate procedures to make the desired geographic selection of layers.

```
proc cbToolApply(class MdispRegionTool tool) {
    local numeric shiftWasPressed = ShiftPressed;
    local numeric ctrlWasPressed = CtrlPressed;
    local class GRE_GROUP currentGroup;
    local class GRE_LAYER currentLayer;

    if (Layout) {
        currentGroup = Layout.FirstGroup;
        currentLayer = Layout.FirstGroup.FirstLayer;
    }
    else currentLayer = Group.FirstLayer;

    View.DisableRedraw = 1;

```

Cycles through all layers

```
    while (currentLayer != 0) {

```

See if layer matches criteria

```
        if (checkLayer(currentLayer)) {
            checkOverlap(currentLayer);
            local numeric visible;
```

Shift + Right Click

```
            if (shiftWasPressed && !ctrlWasPressed) {
                currentLayer.SetVisibleInView(View.GetViewNum(), overlapping ||
                currentLayer.IsVisibleInView(View.GetViewNum() ) );
            }
```

Ctrl + Right Click

```
            else if (!shiftWasPressed && ctrlWasPressed) {
                currentLayer.SetVisibleInView(View.GetViewNum(), !enclosed &&
                currentLayer.IsVisibleInView(View.GetViewNum() ) );
            }
```

Shift + Ctrl + RightClick

```
            else if (shiftWasPressed && ctrlWasPressed) {
                currentLayer.SetVisibleInView(View.GetViewNum(), true);
            }
```

Right Click

```
            else {
                currentLayer.SetVisibleInView(View.GetViewNum(), overlapping);
            }
        }
        currentLayer = currentLayer.NextLayer;
        if (Layout) {
            if ((currentLayer == 0) && (currentGroup.NextGroup != 0) ) {
                currentGroup = currentGroup.NextGroup;
                currentLayer = currentGroup.FirstLayer;
            }
        }
    }

    View.DisableRedraw = 0;
    View.RedrawIfNeeded();
    tool.HasPosition = 0;

```

Uncomment this line to have tool deactivate after each use.

```
    #cbClose();
}
```