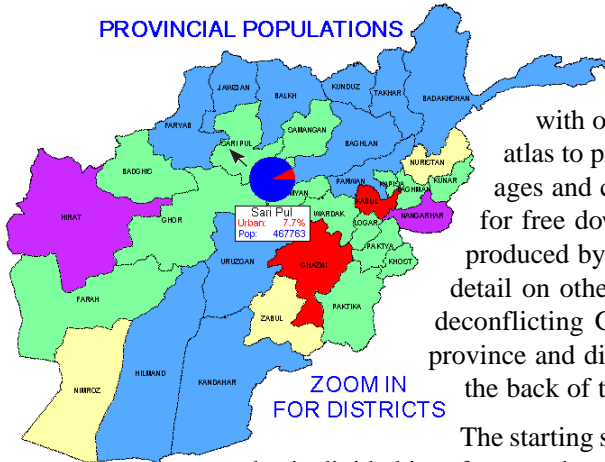
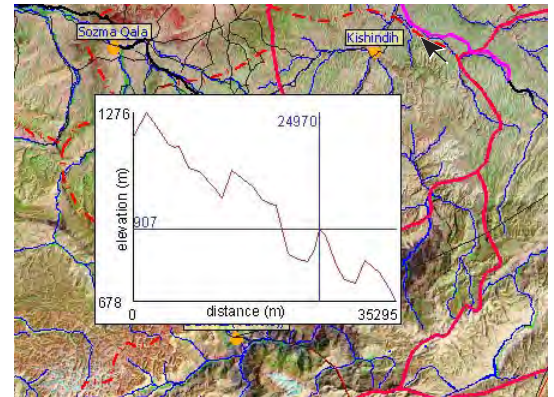


GraphTips in the Afghanistan Atlas



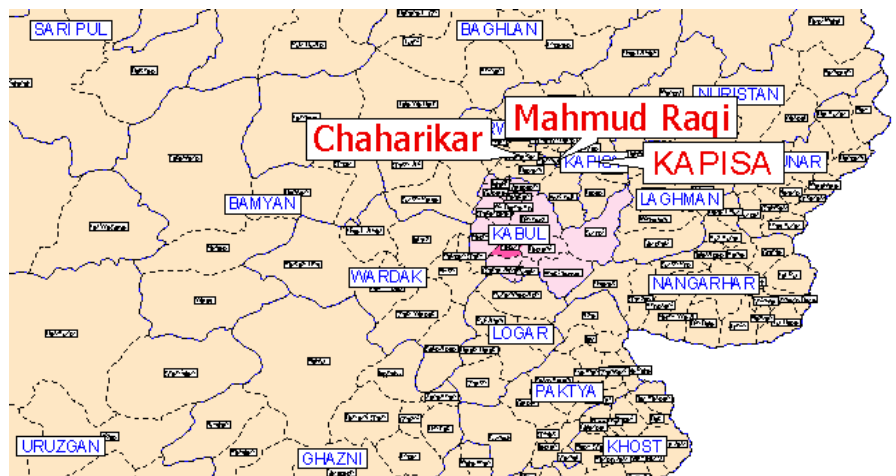
The sample CD atlas of Afghanistan illustrates the use of Display Control Scripts to pop in GraphTips of various types. Four sample Display Control Scripts can be tried, three of which were initially developed for use with other data. These scripts were quickly adapted for use with the Afghanistan atlas to provide an example of how readily the sample scripts prepared by MicroImages and clients can be modified for use with your data. These scripts are available for free download from www.microimages.com/freestuf/scripts.htm. The GraphTips produced by the original scripts that were adapted for use in this atlas are described in detail on other color plates that provide script excerpts on the reverse side. The label deconflicting GraphTip script was developed specifically for use with the Afghanistan province and district data and is illustrated here with an excerpt of the script provided on the back of this page.

The starting screen for the Afghanistan atlas is divided into four quadrants, all of which are linked to different spatial layouts that use a different Display Control Script to provide the GraphTip for each quadrant. For additional information about the components of the Afghanistan atlas, see the color plate entitled *Afghanistan Atlas on CD*. The DEMOGRAPHIC & HAZARDS quadrant is linked to two spatial layouts that use the same Display Control Script, which pops in a GraphTip that provides the province name, the urbanization percentage of the province presented as a pie chart and text, and the total population of the province. The urbanization percentage is the same color (red) as the urbanization representation in the pie chart. Province-level information is displayed by the GraphTip whether you are viewing province or district outlines. For more details on this GraphTip, see the color plate entitled *Pie Chart and Bar Graph*.



for the real-time slope and aspect calculations again comes from the bottom layer in the group. For more information on this GraphTip, see the color plate entitled *Enhanced DataTips and GraphTips*.

The PLANIMETRIC MAPS quadrant is linked to a spatial layout that provides access through HyperIndex links from the province and district polygons to province and district maps in PDF format prepared by the Afghan Information Management Service. The Display Control Script for this layout (see back) zooms individual labels to readable size when they are too small to be legible and enlarges and relocates, or deconflicts, overlapping labels. The original labels remain in position. Note also that the text color for the temporarily zoomed labels is different so they stand out from the originals.



The COUNTRY IMAGE MAPS quadrant is linked to a spatial layout that uses a Display Control Script to pop in a GraphTip that highlights the road segment line closest to the cursor and presents its elevation information as a profile graph. The road's elevation information is obtained from the bottom layer in the group being viewed, which is a 16-bit unsigned elevation raster object. See the color plate entitled *Profile of Nearest Line* for more information on this GraphTip.

The KABUL MAPS quadrant is linked to a spatial layout with a Display Control Script that pops in a GraphTip that presents the slope and aspect at the cursor location graphically and as text. The elevation information

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 Excerpt for Deconflicting Labels (LabelDecon.sml)

```

proc OnLayoutDrawEnd (
  class GRE_LAYOUT layout,
  class GRE_VIEW view
) {
  group = layout.GetGroupByName("Group 1");
  g_group = group;
}
func OnViewDataTipShowRequest (
  foregroundcolor.name = "red2";
  offset.x = 10;
  offset.y = 10;
  imagegc = imgdev.CreateGC();
  while (layer) {
    labels = layer.GetLabels(view);
    labels.GetItemAtLocation(point, place, 10);
    reg = place.GetRegion();
    places = labels.GetItemsOverlappingRegion(reg);
    numeric number = places.GetNumItems();

    if (number > 1) {
      groupreg.Clear();
      for q = 1 to number {
        groupreg.UnionRegion(places[q].GetRegion());
      }
      groupextents = groupreg.Extents;
      groupcenter = groupextents.center;
      groupextents.Expand(10, 10);
      reg.Clear();
      for q = 1 to number {
        place = places[q];
        center = place.GetRegion().Extents.center;
        angle = groupcenter.GetAngle(center);
        place.SetCopyOnWrite(1);
        while (place.GetNumAttachments()) {
          place.DeleteAttachment(place.GetAttachment(1).Element);
        }
        zoom = 30 / place.GetRegion().Extents.GetHeight();
        if (zoom < 1.0) zoom = 1.0;
        place.Zoom(zoom);

        dTheta = 0;
        SolutionFound = false;
        while (!SolutionFound && dTheta < 360) {
          delta.x = cos(angle);
          delta.y = sin(angle);
          extents = place.GetRegion().Extents;
          offset.x = groupcenter.x - extents.center.x;
          offset.y = groupcenter.y - extents.center.y;
          extents += offset;

          while (groupextents.Overlaps(extents)) {
            offset += delta;
            extents += delta;
          }
          SolutionFound = reg.TestRect(extents, "FullOutside");
          if (!SolutionFound) {
            dTheta += 10;
            angle += 10 / deg;
          }
        }
        offsets[q] = offset;
        place.Move(offset);
        reg.UnionRegion(place.GetRegion());
      }
      groupdelta.x = -reg.Extents.x1;
      groupdelta.y = -reg.Extents.y1;
      imgdev.Create(reg.Extents.GetHeight()+5,reg.Extents.GetWidth()+5);
      imgdev.SetPixelSizeMM(view.PixelSizeMillimeters,
        view.PixelSizeMillimeters);
      imgdev.Clear("white");
      imagegc = imgdev.CreateGC();
      imagegc.SetOutputScale(view.GetMapScale());
      maskdev.Create(imgdev.GetHeight(),imgdev.GetWidth());
      maskdev.SetPixelSizeMM(view.PixelSizeMillimeters,
        view.PixelSizeMillimeters);
      maskdev.ClearAll();
      maskgc = maskdev.CreateGC();
      maskgc.SetOutputScale(view.GetMapScale());
      maskgc.SetPixelFunction("Set");
      place.DrawSample(maskgc, foregroundcolor);
      newpt.x = oldcenter.x - imgdev.GetWidth() / 2;
      newpt.y = oldcenter.y - imgdev.GetHeight() / 2;
      offset = newpt - point;
      datatip.SetImageTip(imgdev, maskdev, offset);
      return (1);
    }
  }
}

```

procedure called after layout is drawn

function called when DataTip event is triggered

set text color and offset so label is not under cursor

find the extents of the labels in the view to deconflict

expand the extents for good looking balloon leaders

compute zoom and position to deconflict

exclude leader line from zoom calculation

exclude leader line from zoom calculation

new label position

show the deconflicted labels

how to handle a single label (no overlap)

move out from center until label no longer intersects actual label extents

ensure enlarged labels do not overlap

get new extents and move the label copy

draw the label to be used

draw the label to be used

new label position

show the DataTip