

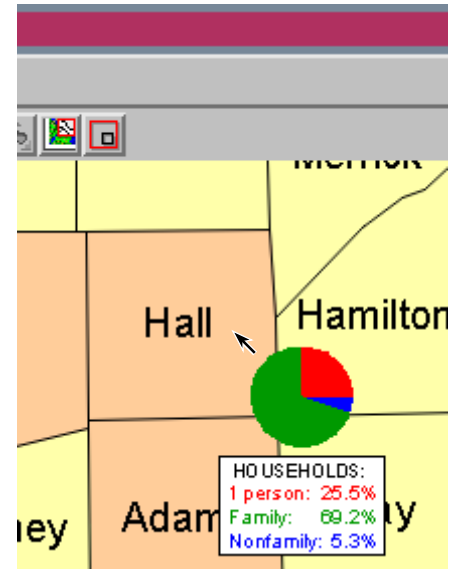
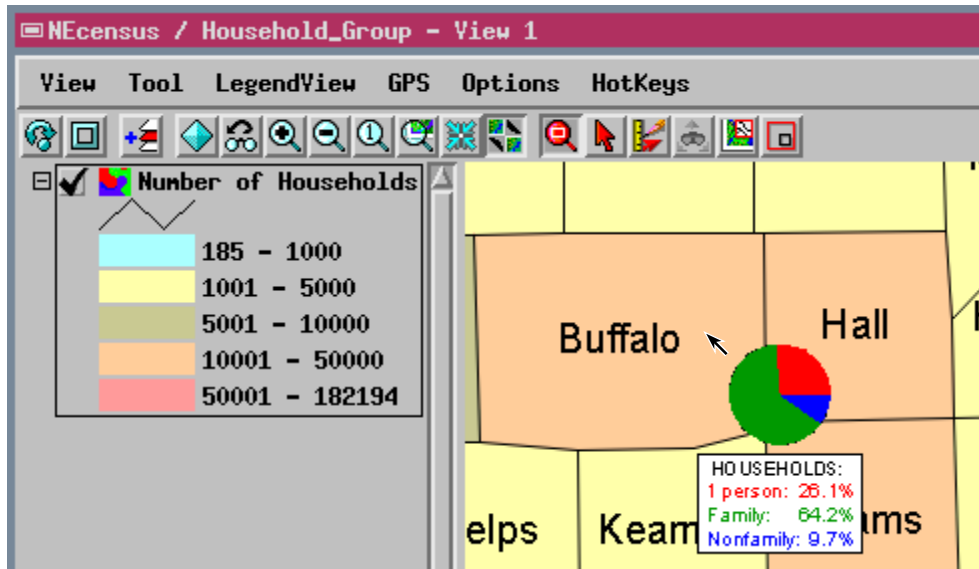
## Sample GraphTip Script

# Pie Chart and Bar Graph

A common use of GraphTips is to pop-in a graphical presentation of the attributes of the geographic position of the cursor. This is similar to a DataTip, but the graphical presentation in the GraphTip makes it easier to compare relative values. GraphTips are the dynamic equivalent of a pin-map, but without the clutter of presenting all the pins simultaneously, which might obscure or distract from the spatial layers in the view. Like a DataTip, a GraphTip can be restricted to operating

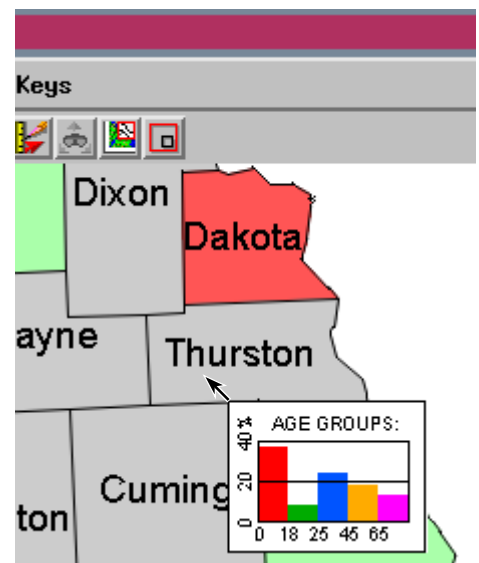
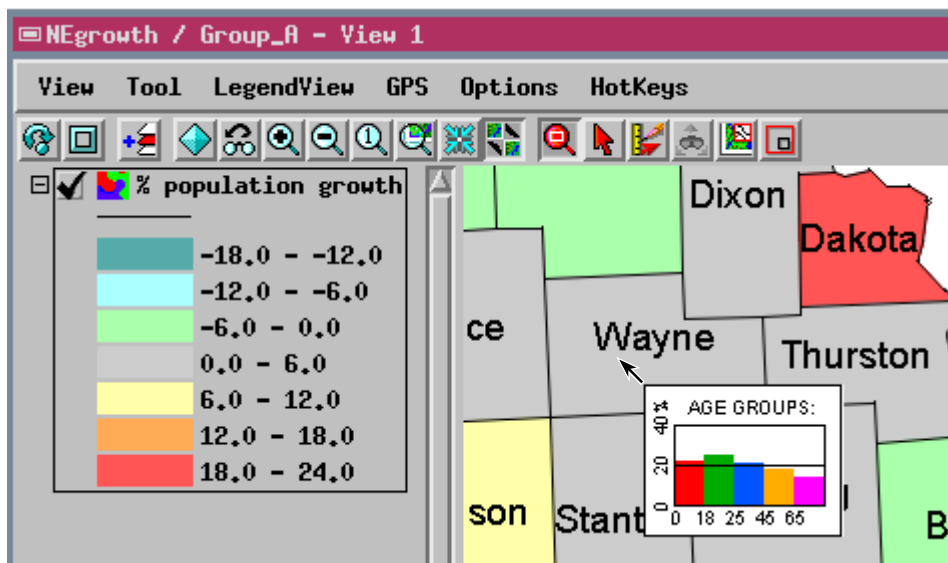
only when the scale of the view is in an appropriate range.

The illustrations below show two GraphTip examples that plot categorical breakdowns of population values for counties (polygons in a vector object). Excerpts of the Display Control Scripts that create these GraphTips are shown on the opposite side of this page, and the complete scripts are available for download.



The county polygons in this view are Theme Mapped by the number of households in the county. The GraphTip shows a pie chart for each county with the percentage of these households in different categories. The box in the lower part

of the Graph-Tip lists the categories and their percentages, with the text colors providing the legend for the pie chart above. The Theme Map and GraphTip values are all found in a single table associated with the polygons.



The Theme Mapped county polygons in this view show percentage population growth (or loss) between 1990 and 2000; these values are produced by a computed field in a table with 1990 and 2000 population figures for each county. The GraphTip shows a bar chart with the percentages of the

population in each of five age categories, with these values coming from a different table. The combination of GraphTip and Theme Map enables an easy visual assessment of the relationship between population age breakdown and the long-term population growth patterns for these counties.

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](http://www.microimages.com/freestuf/scripts.htm).

## Script Excerpt for Pie Chart GraphTip

```

func OnViewDataTipShowRequest (
class GRE_VIEW view,
class POINT2D point,
class TOOLTIP datatip
) {
numeric retval = 1;

trans = view.GetTransLayerToScreen(cntyLayer, 1);
ptLayer = trans.ConvertPoint2DFwd(point);

polyNum = FindClosestPoly(CntyVec, ptLayer.x, ptLayer.y, vecGeoref, 0);

total = CntyVec.poly[polyNum].Household.Total;

onePerson = 360 * CntyVec.poly[polyNum].Household._1_person_house / total;
family = 360 * CntyVec.poly[polyNum].Household.Family_househd / total;
nonfamily = 360 * CntyVec.poly[polyNum].Household.Nonfamily_house / total;

gc = imagedev.CreateGC();
gc.SetColorName("white");
gc.SetLineWidth(1, "pixels");
gc.FillRect(1,60,84,50);
gc.SetColorName("black");
gc.DrawRect(1,60,82,49);

gc.DrawTextSetFont("ARIAL.TTF");
gc.DrawTextSetHeightPixels(9);
gc.TextStyle.RoundWidth 1;
color.Name = "black";
gc.SetColor(color);
gc.DrawTextSetColors(color);
gc.DrawTextSimple("HOUSEHOLDS:", 8, 71);

gc.DrawTextSetHeightPixels(10);
color.Name = "red";
gc.SetColor(color);
gc.FillArcWedge(42, 30, 25, 25, 0, onePerson);
gc.DrawTextSetColors(color);
gc.DrawTextSimple("1 person:", 5, 82);
percent$ = sprintf("%.1f%", onePerson / 3.6);
start = 80 - gc.TextGetWidth(percent$);
gc.DrawTextSimple(percent$, start, 82);

color.red = 0; color.green = 60; color.blue = 0;
gc.SetColor(color);
gc.FillArcWedge(42, 30, 25, 25, onePerson, family);
gc.DrawTextSetColors(color);
gc.DrawTextSimple("Family:", 6, 94);
percent$ = sprintf("%.1f%", family / 3.6);
start = 80 - gc.TextGetWidth(percent$);
gc.DrawTextSimple(percent$, start, 94);

color.Name = "blue";
gc.SetColor(color);
gc.FillArcWedge(42, 30, 25, 25, onePerson + family, nonfamily);
gc.DrawTextSetColors(color);
gc.DrawTextSimple("Nonfamily:", 6, 106);
percent$ = sprintf("%.1f%", nonfamily / 3.6);
start = 80 - gc.TextGetWidth(percent$);
gc.DrawTextSimple(percent$, start, 106);

datatip.SetImageTip(imagedev, maskdev);

return retval;
}

```

called when DataTip event is triggered

predefined class variables

in GraphTip use only what is created by script

find cursor position in layer coordinates

get values for households for polygon

create graphics context for GraphTip

fill white rectangle with black border for label background

draw title for label area

draw pie slice for one-person households in red

draw label and percentage in same color

draw pie slice for family households in dark green

draw label and percentage in same color

draw pie slice for non-family households in blue

draw label and percentage in same color

set the rendered image and mask as source for the GraphTip

## Script Excerpt for Bar Chart GraphTip

```

func OnViewDataTipShowRequest (
class GRE_VIEW view,
class POINT2D point,
class TOOLTIP datatip
) {
numeric retval = 1;
polyNum = 0;

trans = view.GetTransLayerToScreen(cntyLayer, 1);
ptLayer = trans.ConvertPoint2DFwd(point);
polyNum = FindClosestPoly(CntyVec, ptLayer.x, ptLayer.y, vecGeoref, 0);

pcUnder18 = CntyVec.poly[polyNum].NEcntyPopAge.pc_under_18;
pc18to24 = CntyVec.poly[polyNum].NEcntyPopAge.pc_18_to_24;
pc25to44 = CntyVec.poly[polyNum].NEcntyPopAge.pc_25_to_44;
pc45to64 = CntyVec.poly[polyNum].NEcntyPopAge.pc_45_to_64;
pc65andOver = CntyVec.poly[polyNum].NEcntyPopAge.pc_65_and_over;

gc = imagedev.CreateGC();
gc.SetColorName("white");
gc.FillRect(0, 0, 100, 75);
gc.SetColorName("black");
gc.DrawRect(0, 0, 98, 75);

color.red = 100; color.green = 0; color.blue = 0;
gc.SetColor(color);
gc.FillRect(15, 60 - pcUnder18, 14, pcUnder18);

color.red = 0; color.green = 67; color.blue = 0;
gc.SetColor(color);
gc.FillRect(30, 60 - pc18to24, 14, pc18to24);

color.red = 0; color.green = 33; color.blue = 100;
gc.SetColor(color);

gc.FillRect(45, 60 - pc25to44, 14, pc25to44);

color.red = 100; color.green = 67; color.blue = 0;
gc.SetColor(color);
gc.FillRect(60, 60 - pc45to64, 14, pc45to64);

color.red = 100; color.green = 0; color.blue = 100;
gc.SetColor(color);
gc.FillRect(75, 60 - pc65andOver, 14, pc65andOver);

gc.DrawTextSetFont("ARIAL.TTF");
gc.DrawTextSetHeightPixels(9);
gc.TextStyle.RoundWidth = 1;
color.Name = "black";
gc.SetColor(color);
gc.DrawTextSetColors(color);
gc.DrawTextSimple("AGE GROUPS:", 22, 15);

gc.DrawRect(15, 20, 75, 40);
gc.MoveTo(15, 40);
gc.DrawTo(90, 40);

gc.DrawTextSimple("0", 12, 63, 90);
gc.DrawTextSimple("20", 12, 45, 90);
gc.DrawTextSimple("40%", 12, 25, 90);

gc.DrawTextSimple("0", 14, 70);
gc.DrawTextSimple("18", 26, 70);
gc.DrawTextSimple("25", 41, 70);
gc.DrawTextSimple("45", 56, 70);
gc.DrawTextSimple("65", 71, 70);

datatip.SetImageTip(imagedev, maskdev, offset);
return retval;
}

```

called when DataTip event is triggered

predefined class variables

in GraphTip use only what is created by script

find cursor position in layer coordinates

read values from database for polygon under cursor

create graphics context for GraphTip

fill white rectangle with black border for background

fill rectangles with different colors left to right to create vertical bars using population category value as height

set font attributes for labels

draw black frame and grid line for chart

vertical axis labels

draw labels for age group boundaries on horizontal axis (boundaries of bars)

set the rendered image as source for the GraphTip