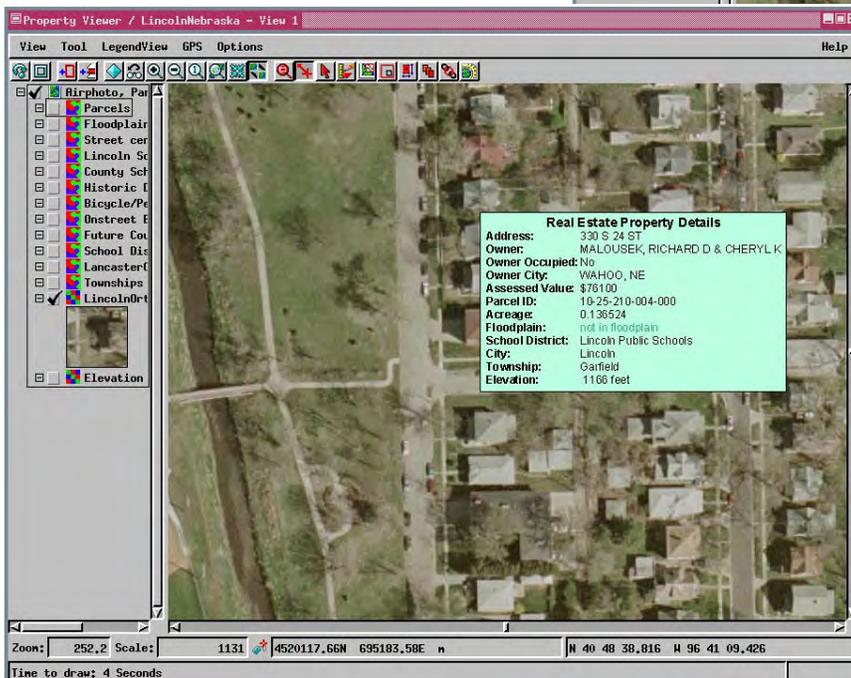
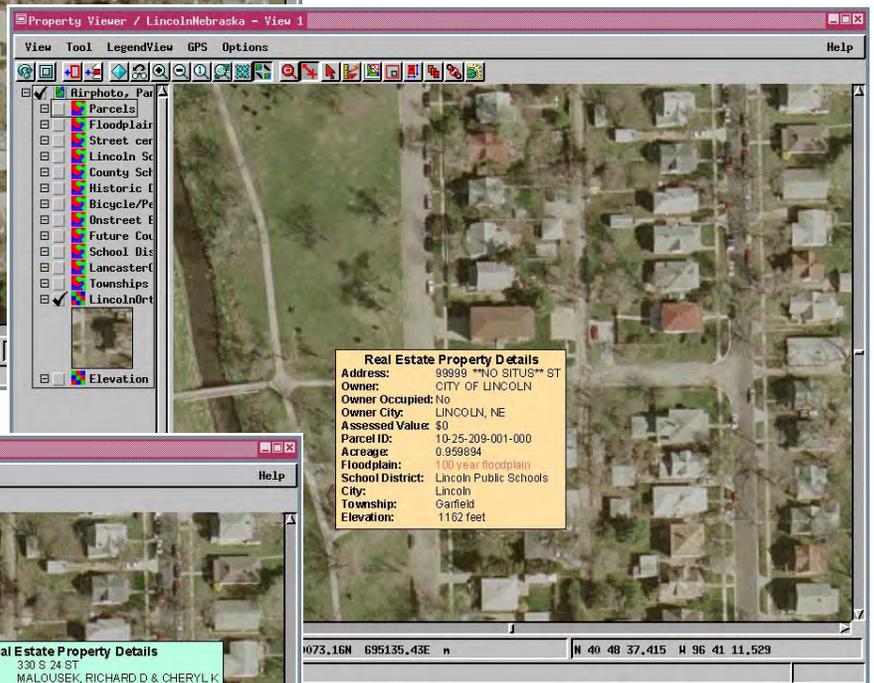
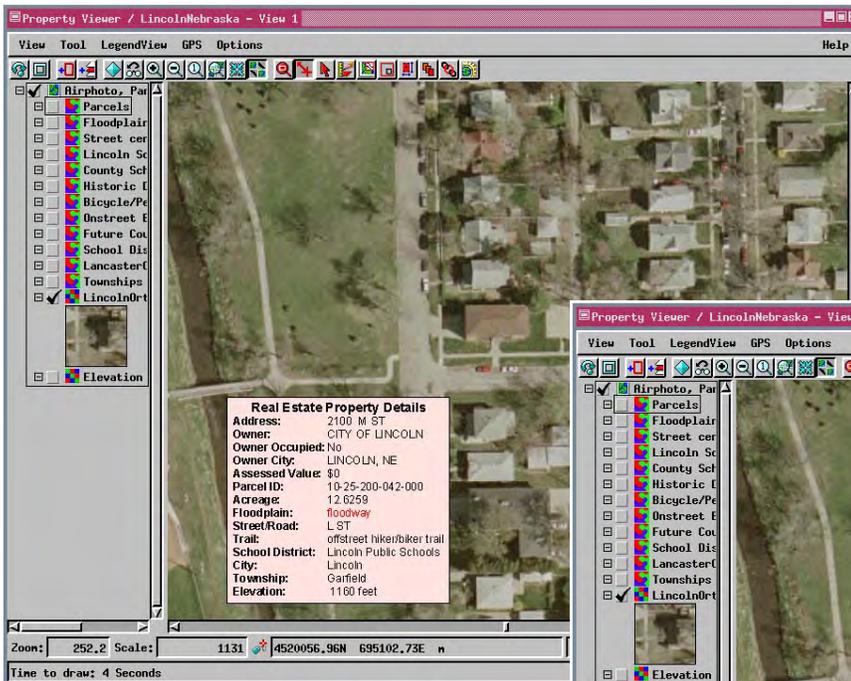


Add Styling to DataTips

You can extend the functionality of your DataTips with a Control Script. The DataTip comes from the layers in the view, and a Control Script can modify its appearance. You may set up a DataTip for every layer in a layout or group and choose whether to show the DataTips for all layers, for visible layers, for the active or top layer, or not to show any DataTips. The content of the DataTip is set through the Layer Controls window. DataTips may come from any field in the tables for a layer including virtual fields. Using virtual fields allows you to create multiline DataTips with information from different fields, to incorporate style codes (see back of this page), and to convert database values into publicly understandable values. For example, the first seven lines in the DataTips illustrated here come from a

single parcels layer; the DataTips incorporate bold and normal weight text, as well as different color text; and a coded floodplain field has been converted from codes 0, 1, 2, or 5 to: not in floodplain, 100 year floodplain, floodway, and 500 year floodplain, respectively. The control script applies to all DataTips regardless of their content.



In the illustrations, a Control Script (Layout/Edit Control Script or Group/Edit Control Script in the Layout or Group Controls window) is used to change the background color of the DataTip depending on the floodplain status of the current cursor geolocation. Note also that the DataTips have bold text for the prefix only and that there is tab alignment for data values. Also note that the floodplain status is shown in a different color when over the floodplain as a means to alert the viewer of the reason for the change in DataTip color. The number of lines in the DataTip will vary with the number of layers that have elements within the search distance when you have All Layers selected as your DataTip viewing option.

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 to Provide Heading and Vary Background Color by Attribute

```

func OnViewDataTipShowRequest (
class GRE_VIEW view,
class POINT2D point,
class TOOLTIP datatip
){
datatip.String = "{~CJ~TS14~FARIALBD.TTF}Real Estate Property Details\n{~LJ~TS12}";
return (false);
}“
func OnViewDataTipShowText (
class GRE_VIEW view,
class POINT2D point,
class TOOLTIP datatip
){
datatip.MarginHeight = 5;
datatip.MarginWidth = 5;
datatip.BackgroundColor.name = "sea foam";
if (datatip.String contains "floodway")
    datatip.BackgroundColor.name = "misty rose"; ;
else if (datatip.String contains "100 year")
    datatip.BackgroundColor.name = "light orange";
else if (datatip.String contains "500 year")
    datatip.BackgroundColor.name = "lemon chiffon";
return (true);
}

```

function name predefined for when DataTips are called

class instances for DataTips in 2D views

sets the heading text

allows preset DataTip text to be used

function name predefined for when DataTips are called

class instances for DataTips in 2D views

set the size of the margin between the DataTip text and frame border in pixels

determine frame background color by attribute using color names from rgb.txt

use the DataTip text

Text Formatting Control Codes

The formatting control codes used to create bold text, change the font, insert tabs, and create other effects are listed below. You can enter any number of codes within a single set of curly brackets. In a computed field, the formatting codes should be within the quotes used to specify the text or be in quotes by themselves if the text is coming from a database field.

		{~BX}	boxed
		{~/BX}	not boxed
{~LJ}	text flush left		
{~RJ}	text flush right		
{~CJ}	center text		
{~FJ}	justify text (even left and right margins)		
		{~S+}	superscript
		{~S-}	subscript
		{~S0}	turns off superscript / subscript while maintaining other set styles
		{~AN}	establishes text anchor position for subsequent overprint
		{~OS}	text following this command will overprint preceding text, starting at the last anchor established by {~AN}
		{~SM}	smoothing
		{~/SM}	no smoothing
		{~K}	kerning
		{~/K}	no kerning
		{~GLRx}	glyph rotation in degrees
		{~IG}	vertical flip of glyph
		{~/IG}	return to normal text orientation
		{~TSx}	sets text size to x points
		{~LSx}	sets line spacing to x points
		{~Cr,g,b}	sets color to RGB values (from 0 to 100)
		{~C[BG]r,g,b}	sets text background color (values from 0 to 100)
		{~Fname}	sets font to "name" font
{~TABS xj}	sets tabs where x is the character position to put the tab stop, j (optional) indicates the type of tab justification (L, R, or C for left, right, or center aligned tabs, respectively; L is the default), and l (optional) indicates the type of leader line that precedes the tab stop (the default is no leader line; the options are ".", "-", or "_" where the character in quotes is a sample of the leader line type (dot, dash, or underline). If you specify the interval for only one tab, that interval is also used for other tabs (for example, you specify three tabs but only give an interval of 0.5 inches for the first tab; all three tabs occur at intervals of 0.5 inches). You can also specify multiple tabs at the same time by separating the tab positions with commas (for example, {~TABS 0.5, 1.5, 3R}).		