



GIS Labeler

Description:

The *ODOT_GIS_Labeler.mvba* is a Visual Basic Application developed to create updateable labels of GIS attributes. This works with shape file attachments as well as Bentley Map XFM features.

The application file can be found in the following location:

ODOTcadd Standards: ..\ODOTcadd\Standards\vba\ODOT_GIS_Labeler.mvba.

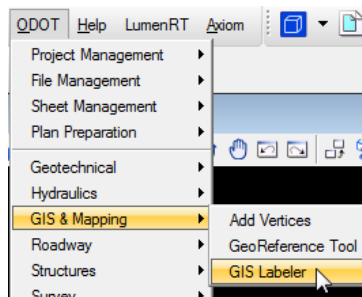
Follow the [link](#) to watch the How To YouTube video on this application.

The main purpose of this documentation is to explain how to use this application.

Loading the Application:

To load the *ODOT_GIS_Labeler.mvba* open MicroStation and select **ODOT > GIS & Mapping > GIS Labeler** from the MicroStation menu or by the following key-in command:

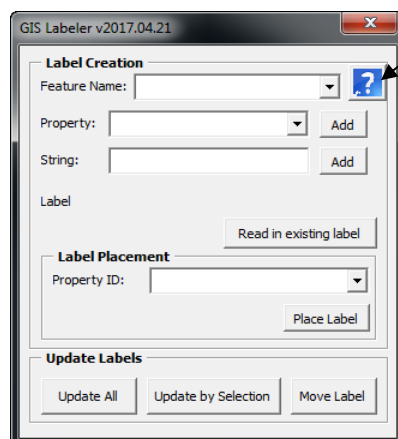
```
vba load ODOT_GIS_Labeler.mvba; vba run modLabelMain.StartGISLabeler
```



Application Operation:

When loaded the dialog shown to the right will appear.

This dialog is comprised of two main sections; [Label Creation](#) and [Update labels](#).



This button will open this PDF.



Label Creation:

The label creation section is used to build a label and place labels. In order to place a label a [Feature Name](#) must be selected and there must be at least one character for the label.

Feature Name Dropdown List:

This list is populated with all the features found in the active model at the time this application is opened, this includes references. Once a feature is selected the [Property Dropdown List](#) will be populated with all the properties (attributes) found for that features.

Property Dropdown List:

This list gets populated when a feature is selected from the [Feature Name Dropdown List](#). To add a selected property to your label select the [Add Button](#) next to the property dropdown list.

Property Add Button:

This button, when selected will add the selected property in the property dropdown list to your label. The syntax for the label will be this, [PropertyName].

String Textbox:

This textbox is used to add non property values to your label. This could be used to add spaces between properties or any text desired to be in the label. To add a desired string to your label select the [Add Button](#) next to the string textbox. Due to the syntax used to denote a Property in the label “[” and “]” characters are restricted and cannot be added into the string textbox.

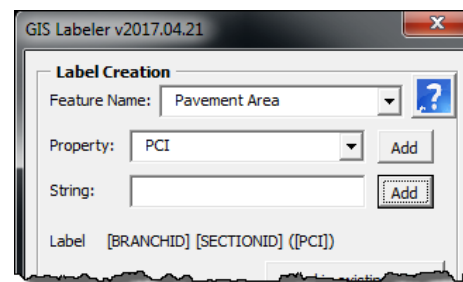
String Add Button:

This button, when selected will add the text from the [String Textbox](#) to your label. Once this button is selected the [String Textbox](#) is cleared of its value.

Label:

This will display your current label. The screen shot below shows an example label. The example label consists of a *BRANCHID* property, a space, *SECTIONID* property, a space and (, *PCI* property, and a).

To clear out a label simply click on the word Label.



Read in existing label Button:

This button can be used to read in a label from a previously placed label. Note that this only works with labels placed by this application, see [Place Label Button](#) for details on this restriction.

Label Placement:

PropertyID listbox:

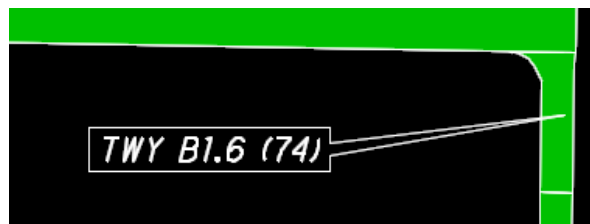
This dropdown box is for selecting a unique identifier property. It is important that you choose a property that will be unique for each element. This is because the label is not directly linked to the element itself. This was done because features can be referenced or reimported causing the element to be redrawn or deleted. The PropertyID will allow this program to search for the



element to update it at a later time. This way if the element was reimported from a shape file you don't have to replace the labels because it wasn't directly linked to the element.

Place Label Button:

Selecting this button will start the place label function as well as a feature name, propertyID, and a label consisting of at least one character are set. This function consists of three steps. The first step is to use the cursor to select the feature you want to label. The feature should highlight. Next you can data point (left click) to accept your selection or reset (right click) to unselect the element and you can then select a different element. The second step (after accepted a selection) is to click at the location you want your label callout to start at. After this step the label will dynamically follow your cursor. You can reset click to select a different location to start the label callout at or data point to place the label. At this point the label is drawn to the model. See example below.



The label consists of 2 elements, a shape and a text element. The text height and width are taken from the active settings in the design file. The shape element is sized to add a 0.5 * text height border around the text. If annotation scaling is true that will be taken into account. The shape element also is filled with a black background. Both elements are placed in the SH_Text level.

The text element is also tagged with a tag set name vbODOTBMLabels. This tagset consists of three tags, PropertyID, LabelSyntax, and PropertyIDValue. The PropertyID tag value is set to the selected Property in the [PropertyID listbox](#). The LabelSyntax tag value is set to the [Label](#). The PropertyIDValue tag is set to the actual value of the selected [PropertyID](#) for the selected element. These three tags are used to update the label at a later time. These tags replace the need to directly attach the label to the element being label. This way elements can be deleted and redrawn and the label will still be able to find the element. Again it is important the PropertyID is a unique property that does not change.

This application will work in rotated views.

Update Labels:

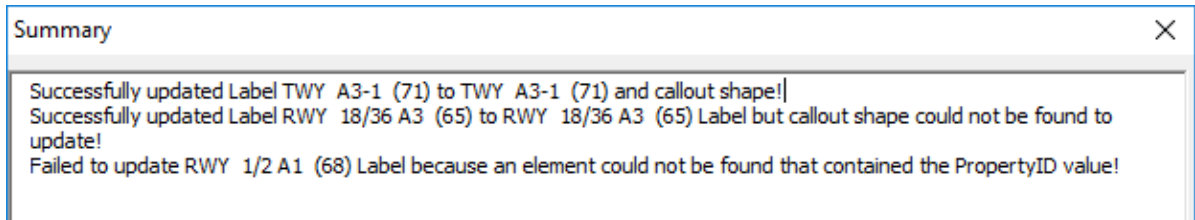
This section consists of 3 buttons, [Update All](#), [Update by Selection](#), and [Move Label](#).

Update All Button:

The update all button, when selected will search the active model looking for any text elements that have the vbODOTBMLabels tagset. Once a text element is found with these tags it then looks for a feature element that contains the same property that's set in the PropertyID tag and that properties value is the same as the PropertyIDValue tags value. Once found the LabelSyntax tag is used to recreate the labels values and then updates the text elements value. The shape around the text element is then updated for the new size and if the view rotation has changed the label will be rotated. When the



application is done processing a dialog as shown below will open listed all the labels and if they were successfully updated or not.



Update by Selection Button:

This button is very similar to the [Update All Button](#) but instead of updating all labels you can select a specific label to update. After selecting the button data point to select a text element. Once a valid text element is selected the element will be highlighted. At this point you can data point to accept the selection and the label will update or reset to deselect the text element. The label is updated the same way as specified in the [Update All Button](#) paragraph above.

Move Label Button:

This button is used to move a label to a different location. Once the button is selected the user simply data point on a valid text element. Once this is done the label will dynamically move with your cursor. Move your cursor to the desired location and data point to redraw the label at that location. Reset click to leave the label at its current position.

Contacts

For any questions, suggestions, or problems with this document please contact the ODOT Office of CADD and Mapping Services by use of the following form on the ODOT website:

https://odot.formstack.com/forms/cadd_servicerequest
