

# Customizing the VI Window

## Goal

Affect the attributes of a VI by using Property Nodes and Invoke Nodes.

## Scenario

You can set the appearance properties of a VI statically by using the VI properties page. However, robust user interfaces often must modify the appearance of a front panel while the program runs.

You must create a VI that can perform the following tasks on demand:

- Show or hide its title bar
- Show or hide its menu bar
- Become transparent so that objects behind the VI can be seen
- Move to the center of the screen

## Design

### Inputs and Outputs

Type	Name	Default Value
Push Button	Show Menu Bar?	True
Push Button	Show Title Bar?	True
Push Button	Make VI Transparent?	True
OK Button	Center	False
Stop Button	Stop	False



**Tip** Use the Push Button controls because their default mechanical action is Switch When Pressed. Use the OK button because its default action is Latch When Released.

## Properties

Use the following properties and methods on the VI class:

- **ShowMenuBar**—When this property is true, the menu bar of the VI is visible.

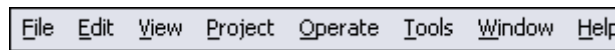


Figure 1. VI Menu Bar

- **TitleBarVisible**—When this property is true, the title bar of the VI is visible.



Figure 2. VI Title Bar

- **RunVITransparently**—When this property is true, the transparency of the VI can vary. The default value of this property is false, so you must write a true value to this property before varying the transparency of the VI.
- **Transparency**—This property varies the transparency of the VI. The property accepts any value between 0 and 100. A value of 0 makes the VI completely opaque (normal behavior), and a value of 100 makes the VI completely transparent (invisible). For this exercise, you set the value to 50 when the **Make VI Transparent?** button is clicked.

## Methods

Unlike properties, a method has an effect every time you call it. Therefore, you should call methods only when you want to perform an action. For example, if you call the `Fp.Center` method during each iteration of a loop, the VI is continually centered, thereby preventing the user from moving it. You can use a Case structure to control calling the method in a given iteration of a loop. Use the following method on the VI class:

- **Center**—Each time this method is called, the VI moves to the center of the screen.



**Tip** Use the Context Help window to view descriptions of each property and method.

## VI Structure

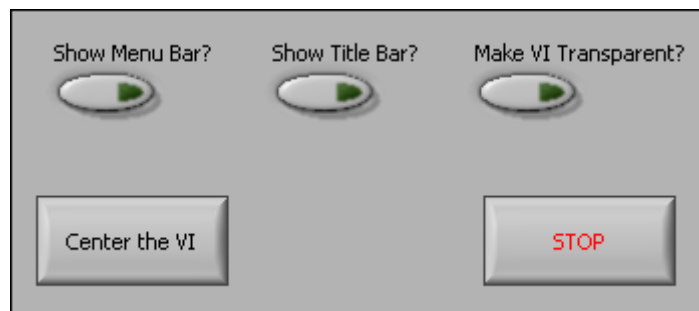
The VI polls the front panel controls every 50 ms and sets the value of the properties based on the current value of the controls. A Case structure controls the execution of the `Center` method.

## Implementation

The files that you need to complete this exercise are here:

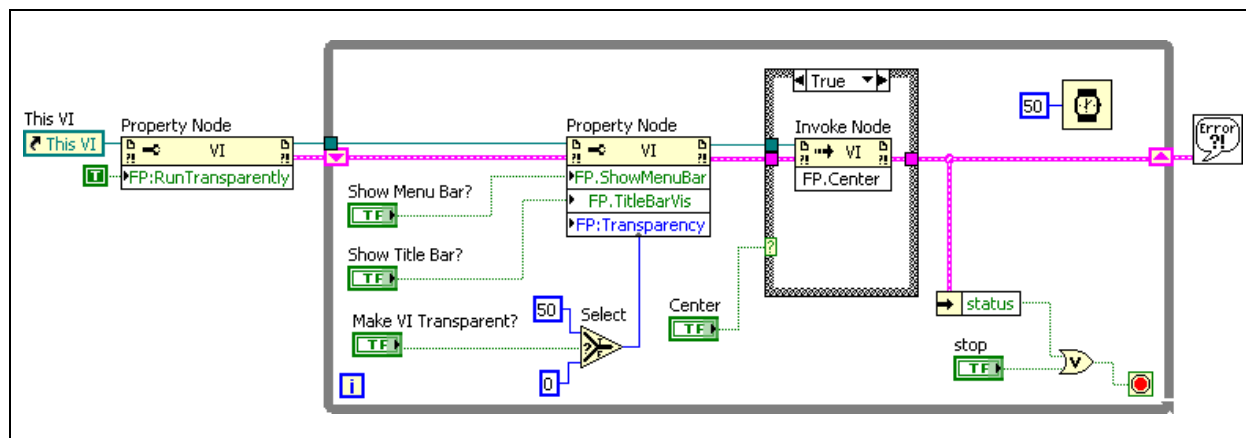
<NI eLearning>\LV Core 2\Invoke Nodes\Exercise.

1. Open *Customize VI Window.vi* in the <Exercise> directory. The front panel of this VI is shown in Figure 3.

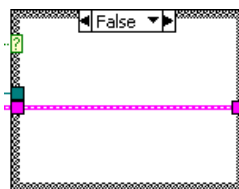


**Figure 3.** Customize VI Window Front Panel

Complete the following steps to create the block diagram as shown in Figure 4.



**Figure 4.** Customize VI Window Block Diagram



**Figure 5.** False Case for Center Method

2. Add a While Loop around the terminals.

## 3. Create a reference to the VI.



- ☐ Add a **VI Server Reference** to the block diagram to the left of the While Loop.
- ☐ Set the VI Server Reference to **This VI** if it is not already.



**Note** Using the This VI reference you can access all the methods and properties of the current VI without having to explicitly open and close a reference.

## 4. Create a Property Node with the RunTransparently property.

- ☐ Right-click the **This VI** reference and select **Create»Property for VI Class»Front Panel Window»Run VI Transparently** from the shortcut menu to create a Property Node.
- ☐ Move the Property Node to the right of the This VI reference, outside of the While Loop.
- ☐ Right-click the Property Node and select **Change All to Write** from the shortcut menu.
- ☐ Right-click the **FP:RunTransparently** property and select **Create»Constant** from the shortcut menu.
- ☐ Change the value of the constant to `True`.

## 5. Create a Property Node with the ShowMenuBar, TitleBarVis, and Transparency properties.

- ☐ Right-click the **This VI** reference and select **Create»Property for VI Class»Front Panel Window»Show Menu Bar** from the shortcut menu to create another Property Node.
- ☐ Expand the Property Node to show three elements.
- ☐ Click the second item in the Property Node and select **Front Panel Window»Title Bar Visible**.
- ☐ Click the third item in the Property Node and select **Front Panel Window»Transparency**.
- ☐ Right-click the Property Node and select **Change All to Write** from the shortcut menu.



- ☐ Move the Property Node inside the While Loop.
- ☐ Add a **Select** function inside the While Loop.

- ☐ Add two **numeric constants** with values 0 and 50 to the left of the Select function.
- ☐ Wire the **0** numeric constant to the f input of the Select function.
- ☐ Wire the **50** numeric constant to the t input of the Select function.
- ☐ Wire the Boolean controls to the appropriate properties, as shown in Figure 4.

6. Create an Invoke Node with the Center method.

- ☐ Right-click the This VI reference and select **Create»Method for VI Class»Front Panel»Center** from the shortcut menu to create an Invoke Node.



7. Add a Case structure around the FP.Center Invoke Node.

8. Add a 50 ms wait to the loop.



- ☐ Add a **Wait (ms)** function to the While Loop.
- ☐ Right-click the **milliseconds to wait** input and select **Create»Constant** from the shortcut menu.
- ☐ Enter 50 in the constant.

9. Set the While Loop to stop when the user clicks the Stop button or when an error occurs.



- ☐ Add an **Unbundle By Name** function to the While Loop.



- ☐ Add an **Or** function to the While Loop.

10. Wire the diagram as shown in Figure 4 and Figure 5. Make sure to replace the error cluster tunnel with a shift register.

11. Display any errors that may occur to the user.



- ☐ Add a **Simple Error Handler VI** to the right of the While Loop.
- ☐ Wire the Simple Error Handler VI to the error cluster output shift register from the While Loop.

12. Save the VI.

## **Test**

1. Switch to the front panel window of the VI.
2. Run the VI.
3. Try each of the buttons and observe the results.

## **End of Exercise**

## Notes

---