COMP 3602

C# Application Development

Week Nine - Online

This Week's Learning Outcomes

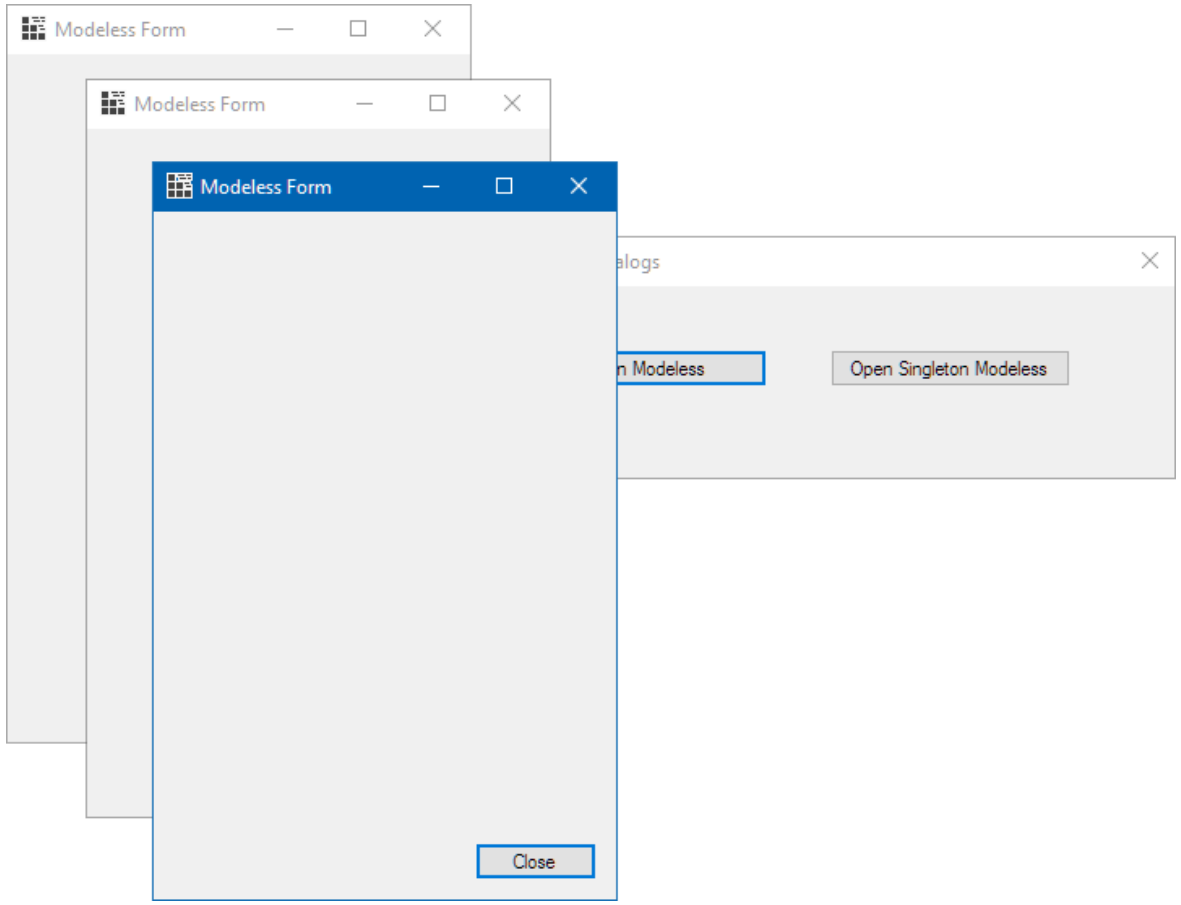
Modeless Dialogs

Modeless dialogs are still child forms but they do not force the user to interact with them; the parent form is still accessible.

Call the Show method to show a form modelessly.

Call either the Hide or Close method to close it.

Close will automatically call Dispose.



Modeless Dialogs - Singleton

Multiple modeless dialogs can be launched from a parent form. Often you will want to limit the number of child forms to one.

```
8 references
public partial class ModelessSingletonForm : Form
{
    private static ModelessSingletonForm instance;

    1 reference
    private ModelessSingletonForm()
    {
        InitializeComponent();
    }

    1 reference
    public static ModelessSingletonForm CreateForm()
    {
        if (instance == null)
        {
            instance = new ModelessSingletonForm();
        }

        return instance;
    }

    1 reference
    private void ModelessSingletonForm_FormClosing(obj
    {
        instance = null; // explicitly set form instar
    }

```

This is accomplished by implementing the “Singleton” pattern for the Modeless form

1) Declare a private static field of the form type
2) Scope the default constructor of the form to private
3) Create a public static method with a return type of the Modeless form
4) Explicitly set the form field to null in the Form_Closing event handler

Create an instance of the form by calling the created static method and calling the Show method

```
private void buttonSingletonModeless_Click(object sender, EventArgs e)
{
    ModelessSingletonForm singleton = ModelessSingletonForm.CreateForm();
    singleton.Show(); // call show method
    singleton.Activate();
}
```

Common Dialogs

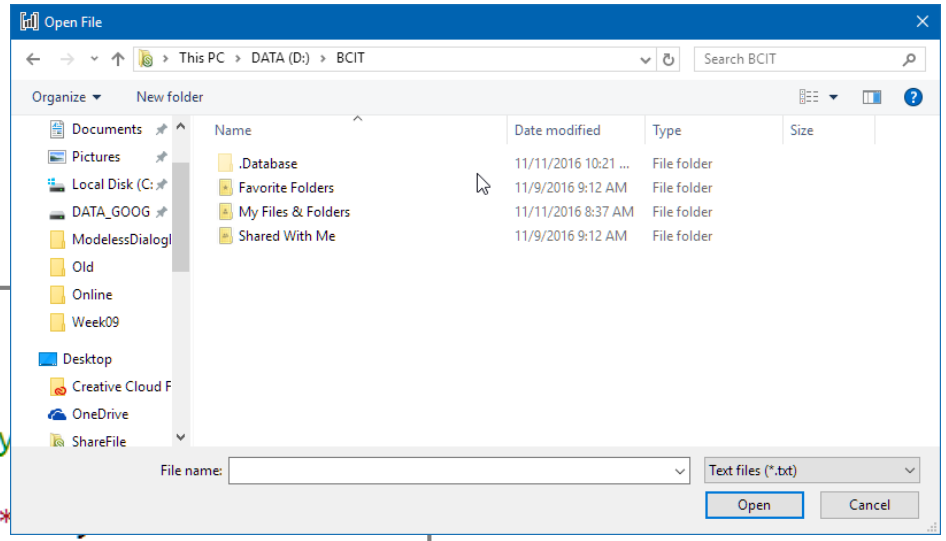
Windows has some built-in dialogs that any application can use. This helps with code reuse and also helps to achieve UI consistency across applications.

In .NET, these common dialogs are available as classes—all derived from the CommonDialog class.

- ColorDialog
- FileDialog (abstract)
  - OpenFileDialog
  - SaveFileDialog
- FolderBrowserDialog
- FontDialog
- PageSetupDialog
- PrintDialog

Common Dialogs - OpenFileDialog

OpenFileDialog contains a lot of functionality but its primary purpose is to get a string from the user




```
24
25      OpenFileDialog dlg = new OpenFileDialog();
26
27      // Initialize the dialog with some values before display
28      dlg.Title = "Open File";
29      dlg.Filter = "Text files (*.txt)|*.txt|All files (*.*)|*
30      dlg.InitialDirectory =
31                  Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
32
33      // Display the dialog.
34      if (dlg.ShowDialog() == DialogResult.OK && dlg.FileName != string.Empty)
35      {
36          try
37          {
38              // Open and display the file.
39              StreamReader reader = new StreamReader(dlg.FileName);
40              textBoxDocument.Text = reader.ReadToEnd();
41              reader.Close();
42          }
43
```