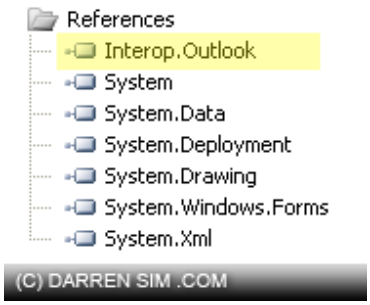
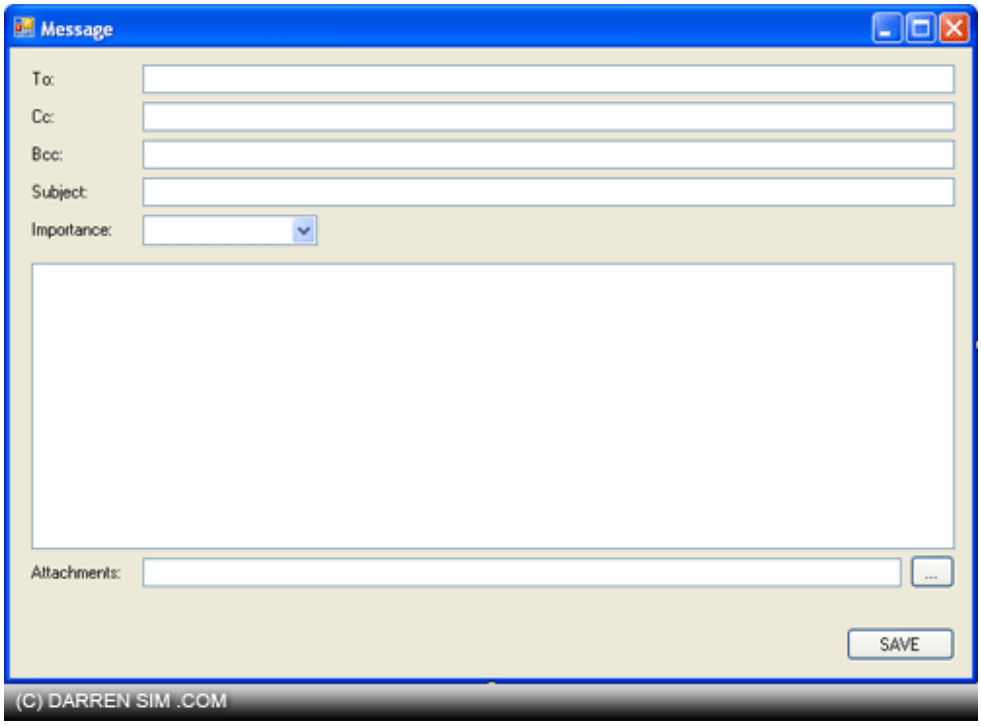
1. To begin, let's first create a C# Winforms Project in Visual Studio.

2. Next, let's add the reference to the interop DLL (attached on this blog post). You should see something like the following.



3. For the purpose of this tutorial, we'd create a simple windows form which takes in the **mail receipient**, **subject**, **message** and **attachment** fields. Once you're familiar with how the code (which we'd discuss later), you can work more magic around this (e.g. creating automation processes)



Note: The **Importance ComboBox** should have the following values "High","Normal" and 'Low".

4. Next step, is to wire up the code-behind for the **Save Button**. It's optional whether you want to register the Outlook Interop in the namespace or not. In this example, I do not so so, thus qualifying the full path in the code (below).

```
// Creates a new Outlook Application Instance
Outlook.Application objOutlook = new Outlook.Application();


// Creating a new Outlook Message from the Outlook Application Instance
Outlook.MailItem mic = (Outlook.MailItem)(objOutlook.CreateItem(Outlook.OlItemType.olMailItem));


// Assigns the "TO", "CC" and "BCC" Fields
mic.To = toTextBox.Text;
mic.CC = ccTextBox.Text;
mic.BCC = bccTextBox.Text;


// Assigns the Subject Field
mic.Subject = subjectTextBox.Text;


// Switch the Importance ComboBox to identify the Mail Message Importance Level
switch (importanceComboBox.SelectedItem.ToString())
{
    case "High":
        mic.Importance = Outlook.OlImportance.olImportanceHigh;
        break;     case "Normal":
        mic.Importance = Outlook.OlImportance.olImportanceNormal;
        break;


    case "Low":
        mic.Importance = Outlook.OlImportance.olImportanceLow;
        break;
}


// Define the Mail Message Body. In this example, you can add in HTML content to the mail message
body
mic.HTMLBody = messageTextBox.Text;


// Adds Attachment to the Mail Message.
// Note: You could add more than one attachment to the mail message.
// All you need to do is to declare this relative to the number of attachments you have.
mic.Attachments.Add(attachmentOneTextBox.Text,Outlook.OlAttachmentType.olByValue,1,"Attachment
Name");


// Save the message to C:\demo.msg. Alternatively you can create a SaveFileDialog to
// allow users to choose where to save the file
mic.SaveAs(@"C:\demo.msg", Outlook.OlSaveAsType.olMSG);
```

Easy isn't it? Besides creating mail messages, you can also create other outlook items such as task, calendar objects and more. More to come in the future posts. Do let me know your comments/views on this post.