Programming with Event Driven in C#

Advanced Programming

Chalew Tesfaye
Department of Computer Science
Debre Berhan University
2017

Outline

- ✓ Create controls on the fly
- ✓ Add help to a project
- ✓ Connect C# application with MS Office
- ✓ C# Socket program
- ✓ Multithreading
  - > Reading assignment
- ✓ COM, ActiveX controls and Unmanaged Code
  - > Reading assignment
- ✓ Test a project
- ✓ Setup project and Deploying an application

Creating Controls On-the-Fly

- creating controls dynamically may require when we don’t know until runtime the number of controls that may be available
- For example, the right amount of textboxes and labels for a user to enter data when the number of questions depends on a database tables.
- **Creating controls on the fly**
- Use normal object creation syntax to create controls
  - ✓ TextBox NewTextBox = new TextBox();
  - ✓ Button NewButton = new Button();

Creating Controls On-the-Fly

- **Initialize the controls**
- Initialization depends on what control you are using and why you are using it
  - ✓ Button NewButton = new Button();
  - ✓ NewButton.Text = "Click Me!";
  - ✓ NewButton.Location = new Point(56, 32);//location on the form
  - ✓ NewButton.Size = new Size(64, 21);//size of the control
- Adding to the container
  - ✓ Controls.Add(NewButton);
- If you are adding multiple controls, use the Suspendlayout and resumeLayout methods
  - ✓ SuspendLayout();
  - ✓ Controls.Add(NewButton1);
  - ✓ Controls.Add(NewButton2);
  - ✓ ResumeLayout();

Creating Controls On-the-Fly…

- Handling events for the control at run time
- NewButton.Click += New EventHandler(this.Button_OnClick);
  - ✓ // This instantiates a delegate and assigns the delegate to the event
- NewButton.Click += this.Button_OnClick;
  // This is simply an abbreviated syntax
- Accessing the controls once created
- Using the same handler for multiple controls is the easiest way

```
private void NewButton_OnClick(Object sender, EventArgs e){
    Button EventControl = (Button)sender;
    // Once we have a strongly typed reference, we can access
    // the control's properties.
    EventControl.Enabled = false;
    EventControl.Text = "I've been clicked!";
}
```

Creating Controls On-the-Fly…

- Depending on your needs, the control reference can be stored in a variable, an array, a class or structure

```
CheckBox[] chkOptionBoxes = null;
private void CreateCheckBoxes(string[] options){
    chkOptionBoxes = new CheckBox[options.Length];

    // This variable stores the Y location
    // of the checkboxes and adds 20 pixels
    // for each one.
    int yPosition;

    SuspendLayout();

    foreach(string Option in options){
        CheckBox newbox = new CheckBox();
        newbox.Left = 20;
        newbox.Top = yPosition;
        yPosition += 20;
        newbox.Size = new Size(100, 20);
        newbox.Text = Option;

        this.Controls.Add(newbox);
    }
    ResumeLayout();
}
```

Creating Controls On-the-Fly…

- Once the controls created and done, it may require removing them

```
private void DeleteOptionCheckBoxes(){
    SuspendLayout();
    foreach(CheckBox c In chkOptionBoxes){
        Controls.Remove(c);
        c.Dispose();
    }
    ResumeLayout();

    // Forget about CheckBoxes
    chlOptionBoxes = null;
}
```

Adding Help to Your Programs

- One way to add help information to a form is to add a Help menu
- Then, you can add item to that menu for various topics.
- When the user selects one of these items, you can display a dialog box with the appropriate information.
- You can also add help information to a form by using
  - ✓ **tool tips**
  - ✓ **Adding context sensitive help**

To add tool tip

- Add ToolTip control to a form, a control named ‘toolTip1’ will appear in the component design tray
- The ToolTip controls makes “**ToolTip on toolTip1”** property becomes available for the **form** and each of its **controls**
- Set this property to the text you want displayed when the user places the mouse pointer over the form or control

To provide context sensitive help

- Add **HelpProvider** control to the form
- A control named helpProvider1 will appear in the component designer tray
- This control makes several additional properties available for the control and each control it contains.
- To display a text string when a user presses **F1** key the control that has the focus, enter the text for the “HelpString on helpProviider1” property of the control
- You can enter help text for HelpString property of the form, then that text is displayed at the location of the mouse pointer if a help string isn’t specified for the control that has the focus
- When you enter text for ‘HelpString on helpProvider1’ property, the “ShowHelp on helpProvider1” property automatically changes from false to true.

Connect C# with Microsoft Office

- Automation of an office file allows us to doing various operations from C#.
- For example, we can automate an Excel file from C# using Excel Object Model or using Microsoft Jet Engine to connect Excel from Csharp
- Through the automation from C# we can achieve
  - ✓ creating a new workbook
  - ✓ adding data to a workbook
  - ✓ creating charts etc, ..

Connecting C# to Excel

- add the Microsoft Excel Object Library to you project.
- Use project->Add reference command to add MS excel object Library
- Search and select Microsoft Excel 14.0 Object Library, click OK

Connecting C# to Excel

```
private void button1_Click(object sender, EventArgs e)
        {
            Excel.Application xlApp;
            Excel.Workbook xlWorkBook;
            Excel.Worksheet xlWorkSheet;
            object misValue = System.Reflection.Missing.Value;
            xlApp = new Excel.Application();
            xlWorkBook = xlApp.Workbooks.Add(misValue);
            xlWorkSheet = (Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
            xlWorkSheet.Cells[1, 1] = "Excel form C#";
            xlWorkBook.SaveAs("D:\\csharp-Excel.xls", Excel.XlFileFormat.xlWorkbookNormal, misValue, misValue, misValue, misValue,
Excel.XlSaveAsAccessMode.xlExclusive, misValue, misValue, misValue, misValue, misValue);
            xlWorkBook.Close(true, misValue, misValue);
            xlApp.Quit();
            releaseObject(xlWorkSheet);
            releaseObject(xlWorkBook);
            releaseObject(xlApp);
            MessageBox.Show("Excel file created , you can find the file D:\\csharp-Excel.xls");
        }
```

Connecting C# to Excel

```
private void releaseObject(object obj)
    {
        try
        {
            System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);
            obj = null;
        }
        catch (Exception ex)
        {
            obj = null;
            MessageBox.Show("Exception Occured while releasing object " + ex.ToString());
        }
        finally
        {
            GC.Collect();
        }
    }
```

Creating Programs for the Internet

- The Microsoft .NET framework provides two namespaces
  - ✓ **System.Net** and
  - ✓ **System.Net.Sockets** for managed implementation of Internet protocols that applications can use to send or receive data over the Internet.
- Network programming in windows is possible with sockets
  - ✓ peer-to-peer Microsoft Windows applications that act as servers and clients to send and receive data
- The System.Net classes provide functionalities that allows
  - ✓ classes to communicate with other applications by using the
    - > Hypertext Transfer Protocol (HTTP),
    - > Transmission Control Protocol (TCP),
    - > User Datagram Protocol (UDP), and
    - > Socket Internet protocols.

Socket Programming

- A Socket is an End-Point of To and From (Bidirectional) communication link between two programs
  - ✓ Server Program and
  - ✓ Client Program running on the same network.
- We need two programs for communicating a socket application in C#.
- **C# Server Socket Program**:
  - ✓ Program running on a computer has a socket that bound to a Port Number on the same computer and listening to the client's incoming requests.
- **C# Client Socket Program**:
  - ✓ Programs have to know the IP Address (Hostname) of the computer that the C# Server Socket Program resides and the Port Number assign for listening for client's request.

Server Socket Program

- This program act as a Server and listening to clients request.
- Here we assign a Port No. 8888 for the Server Socket,
  - ✓ it is an instance of the C# Class TcpListener, and
  - ✓ call its start() method.
- TcpListener serverSocket = new TcpListener(8888);
- serverSocket.Start();
- The next step is to create an infinite loop for monitoring the request from Client's side
- When the Server Socket accepts a request from the Client side,
  - ✓ it reads the data from NetworkStream and
  - ✓ also it writes the response to NetworkStream.

Client Socket Program

- the second part of the C# Server Socket Program.
- The Client is connected to
  - ✓ the Port 8888 of the C# Server Socket Program, and
  - ✓ the IP Address (Computer Name) here we give as 127.0.0.1, localhost
- clientSocket.Connect("127.0.0.1", 8888);
- When the C# Client program starts,
  - ✓ it will connect to the C# Server Socket Program and
  - ✓ start to reads data from NetworkStream, and
  - ✓ also write to the NetworkStream.

Localization

- Localization means to display the Web site in a different way when a different culture is used
- ASP.NET supports localization through resource files
  - ✓ They have a .resx extension
  - ✓ Can be edited with Visual Studio
- Resource files are a collection of name-value pairs
- We can edit them through Visual Studio
- Create a separate file for each culture you want supported
  - ✓ Each resource file should include the locale in its name before the .resx
  - ✓ ASP.NET automatically picks the resource file corresponding to the UI culture of the user

Localization steps in C#

- Create a Resource file for
  - ✓ Ddefault language/locale
  - ✓ Required language/locale
  - ✓ E.g.
    - > GlobalResourse.resx
    - > GlobalResourse.am-ET.resx
- Define name/Value pair variables/parameter in the recourse file
  - ✓ E.g
    - > MsgBoxTitle/የተማሪዎች መረጃ ቁት
    - > lblCollgeText/ኮሌጅ
    - > btnSaveText/አስቀምጥ
- Use/Call the variables
  - ✓ E.g.
    - > GlobalResourse. MsgBoxTitle;
    - > GlobalResourse. lblCollgeText;

Add language selection menu

- Thread.CurrentThread.CurrentCulture = new System.Globalization.CultureInfo(cmbLanguage.Text.ToString());
- Thread.CurrentThread.CurrentUICulture = new System.Globalization.CultureInfo(cmbLanguage.Text.ToString());

Testing a project

- When you test a project, you run it and make sure the application works correctly
- As you test your project
  - ✓ try every possible combination of input data and user actions to be certain that the project works correctly in every case
  - ✓ **In other words, your goal is to make the project fail.**
- Test the user interface
  - ✓ Visually check all the controls to make sure they are displayed properly with the correct text.
- Test valid input data
  - ✓ For example, enter data that you would expect a user to enter
- Test invalid data or unaccepted user action
  - ✓ For example, leave a required field blank, enter a text data in to numeric input fields, and use negative numbers where they are not appropriate
  - ✓ Try everything you can think of to make the application fail.

Deploying Applications

- At point during the process of developing a Windows application,
  - ✓ you need to deploy the application
  - ✓ so you can test it on the target system and finally your users can run it.
- Way to deploy Window application:
  - ✓ XCopy
  - ✓ ClickOne
  - ✓ Setup program
- Using third party program
  - ✓ Advanced Installer

XCOPY

- The oldest and easiest way to deploy a windows application is to copy files that are required by the application to the user's computer.

How to use ClickOnce works

- Build your project
- Use Project Properties and click on publish tab
- Set Installation Mode and setting
- Select application file required by the application
- Select prerequisites file
- Choose option from where you download prerequisite files
- Set application update option
- Set version, option like publisher, product name,…
- Click Purplish
- Go to your DataDirectory\publish folder and look at the setup files

Using Setup program

- Add a new setup project to your solution,
- File->New Project or right Click on the solution and use
- Add->new project Select Othe project type->Setup and Deployment->Visula Studio Installer->Setup project
- Select project output file, right Click on Application folder->Add->Project Output, click OK
- Add required file like database file Application folder->Add->File, browse to file
- Create shortcut Users' Desktop ->Create shortcut to users' desktop from the primary output file
- Create Shortcut to Users programs menu
- Build your project
- Go to your setup project folder and have a look at the setup files
- Install your application on other computer and test it.

Advanced Installer

- Advanced Installer is a Windows Installer authoring tool.
- It offers a friendly and easy to use Graphical User Interface for creating and maintaining installation packages (EXE, MSI, etc.) based on the Windows Installer technology.
- Download and install Advanced Installer
- Start the program
  - ✓ Start a "Visual Studio Application" project type
  - ✓ Enter details about your product
  - ✓ Select a distribution type
  - ✓ Set project and package paths
  - ✓ Select Visual Studio file
  - ✓ Detected configurations
  - ✓ Detected files
  - ✓ Create shortcuts for your applications
  - ✓ Configure prerequisite or merge modules
  - ✓ Application execution
  - ✓ Configure installation UI
  - ✓ Select build languages
  - ✓ License Agreement
  - ✓ Import Visual Studio project
  - ✓ Build the project

Lab Exercise

- **Preparing a Setup file for your project**
- Use ClickOnce way to prepare a setup file for your project
- Use Visual Studio Installer setup project to deploy your application to user system
- Download, install and use the InstallShield limited edition product for preparing a setup program to deploy your application.
  - ✓ InstallShield is a setup program developed by Flexera software and provide more advanced option for prepare setup program for large windows application.
- Install and Use Advanced Installer to deploy your application

For more information

- Deitel, C#-How to Program. USA, 2010
- Svetlin Nakov *et al*. Fundamentals of Computer Programming With C#. Sofia, 2013
- Joel Murach, Anne Boehm. Murach C# 2012, Mike Murach & Associates Inc USA, 2013
- Murachs ADO.NET.4 Database Programming With C# 2010 4th Edition

QUESTION?