Course: EGDF20
Module: EGE202 Application Programming

Practical 6: Media Player Application: Handling Multimedia

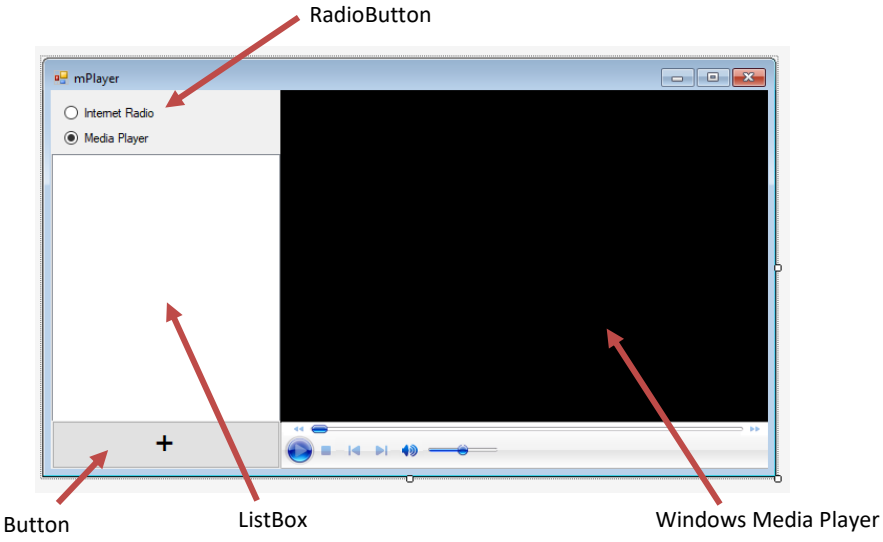
Objectives: At the end of this lab, the student should be able handle multimedia content such as audio/video files and also URI resources.

Exercise 1 – Developing Application that Handles Multimedia Resources

Part 1: Creating a Form with Dual Size Support

1. Under the ***File*** menu, click ***New Project*** or use the ***New Project*** button to create a new project. Alternatively, use the ***Create New Project*** link in the ***Get Started*** *popup* dialog.
2. From the pop-up dialog, select **"C#"** for the ***Language filter***, **"Windows"** for ***the Platform filter*** and **"Desktop"** for the ***Project type filter***.
3. Then choose ***Windows Forms App (.Net Framework)*** and click the ***Next*** button.
4. Type the name of your new project as ***mPlayer*** and keep the Solution name the same as Project name.
5. **Do not** tick on the check-box of [☐ **Place solution and project in the same directory**].
6. Click the **Create** button to start your project.

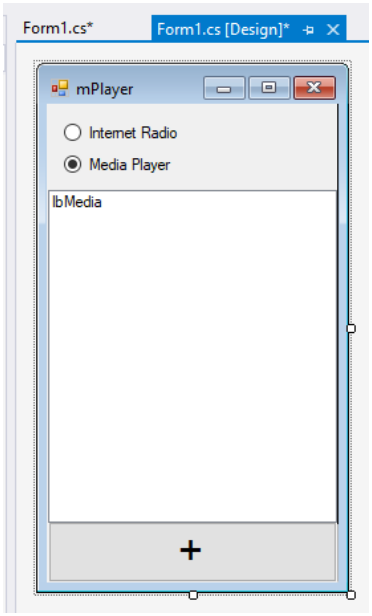
7. In the ***Properties*** window of the ***Form*** control, change the ***TopMost*** property of the ***Form1*** to 'True'.
8. At the ***Form Designer***, select the Form control and set the ***Text*** property of the ***Form*** from '*Form1'* to '*mPlayer'* and ***Size (Width & Height)*** to 700 by 400.
9. Then drag from the ***Toolbar*** 2 *RadioButtons*, 1 *ListBox*, 1 *Button* and 1 *WindowsMediaPlayer* control into the *Form* window.
   (Note: If you cannot find *WindowsMediaPlayer* control in your **ToolBox**, refer to the end of this lab sheet on ***how to Add WindowsMediaPlayer to ToolBox***).
10. Resize and arrange them as shown in the diagram below.

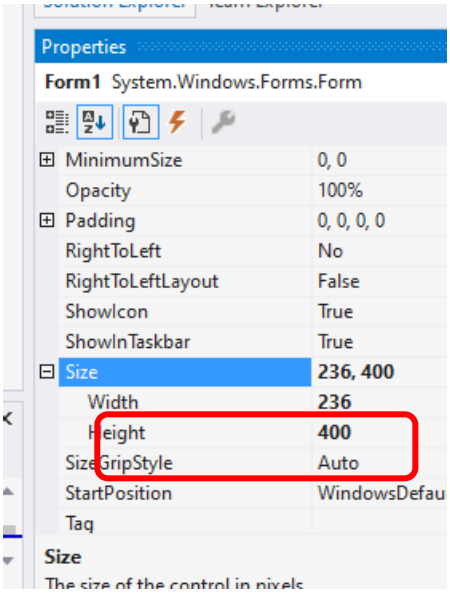


11. Modify the properties of the controls based on the table below

{Name} From	{Name} To	{Text}	{Font->Size}	{Checked}
radiobutton1	rRadio	Internet Radio		
radiobutton2	rMedia	Media Player		True
listBox1	lbMedia			
Button1	btnAdd	+	24	
axWindowsMediaPlayer1	wmpPlayer			

12. Next try to experiment by resizing the *Form* object smaller until on the *WindowsMediaPlayer* object is just nicely hidden. From the ***Properties*** tab, take note of the *Form* ***Width*** (**this value will be used in the next step**). Remember to revert back the ***Width*** to '*700*' before proceed to next step.





13. At the ***Form Designer***, double click on Internet Radio *RadioButton* to create a ***CheckedChanged*** event handler. Add the following codes and note that you need to replace the value 236 with your own value from previous step.

```
private void rRadio_CheckedChanged(object sender, EventArgs e)
{
    this.Size = new Size(236, 400);
}
```

14. Similarly repeat the previous step for Media Player *RadioButton*

```
private void rMedia_CheckedChanged(object sender, EventArgs e)
{
    this.Size = new Size(700, 400);
}
```

15. Build and test the application.

Part 2: Playback Internet Radio Stream from URI Resources

1. Let's modify the Internet Radio ***CheckedChanged*** event to disable the *Button* and also populate the *ListBox* with 2 radio channels. Add the following codes

```
        private void rRadio_CheckedChanged(object sender, EventArgs e)
        {
            this.Size = new Size(236, 400);

            btnAdd.Enabled = false;
            stations.Clear();
            stations.Add(new KeyValuePair<string, string>("Class 95",
@"https://19183.live.streamtheworld.com/CLASS95_SC"));
            stations.Add(new KeyValuePair<string, string>("Yes 933",
@"https://19183.live.streamtheworld.com/YES933_SC"));
            lbMedia.DataSource = stations;
            lbMedia.DisplayMember = "Key";
            lbMedia.ValueMember = "Value";

        }
```

2. Notice that there is a syntax error on *stations* object. This is a ***KeyValuePair*** object. ***KeyValuePair*** object consist of 2 values. The 2 values correspond to a Key and Value objects. Commonly used to store a pair of values such as in areas such as Login/Password, Admin/Name and etc.
3. We need to add the declaration for stations as follows:

```
        public partial class Form1 : Form
        {
            List<KeyValuePair<string, string>> stations = new
List<KeyValuePair<string, string>>();

            public Form1()
            {
                InitializeComponent();
            }
```

4. Build and test the application. Ensure that when Internet Radio is selected, the ListBox will have 2 channels.
5. Next, lets add the ***SelectedIndexChanged*** event to the *ListBox*, such that once the selection has been made, it will play the media.
6. Double click on the *ListBox* and add the following codes to ***SelectedIndexChanged*** event handler:

```
    private void lbMedia_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (rRadio.Checked)
         wmpPlayer.URL = ((KeyValuePair<string, string>)lbMedia.SelectedItem).Value;
    }
```

7. Build and test to ensure the radio is streamed and played,

Part 3: Playback Videos from Local Resources

1. Let's modify the Media Player ***CheckedChanged*** event to enable the *Button* and also clear any items/contents inside the ListBox by adding the following codes

```
private void rMedia_CheckedChanged(object sender, EventArgs e)
{
    this.Size = new Size(700, 400);

    btnAdd.Enabled = true;
    lbMedia.DataSource = null;
    lbMedia.Items.Clear();
}
```

2. Next let's add the ***Clicked*** event handler to the button and add the following codes:

```
private void btnAdd_Click(object sender, EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.Multiselect = true;
    if (ofd.ShowDialog() == DialogResult.OK)
    {
        filenames = ofd.SafeFileNames;
        paths = ofd.FileNames;

        for (int i = 0; i < filenames.Length; i++)
        {
            lbMedia.Items.Add(filenames[i]);
        }
    }

}
```

3. We need to add the declaration for *filenames* and *paths* as follows:

```
public partial class Form1 : Form
{
    List<KeyValuePair<string, string>> stations = new
List<KeyValuePair<string, string>>();

     string[] paths, filenames;
```

4. Next, we need to modify the ***SelectedIndexChanged*** event

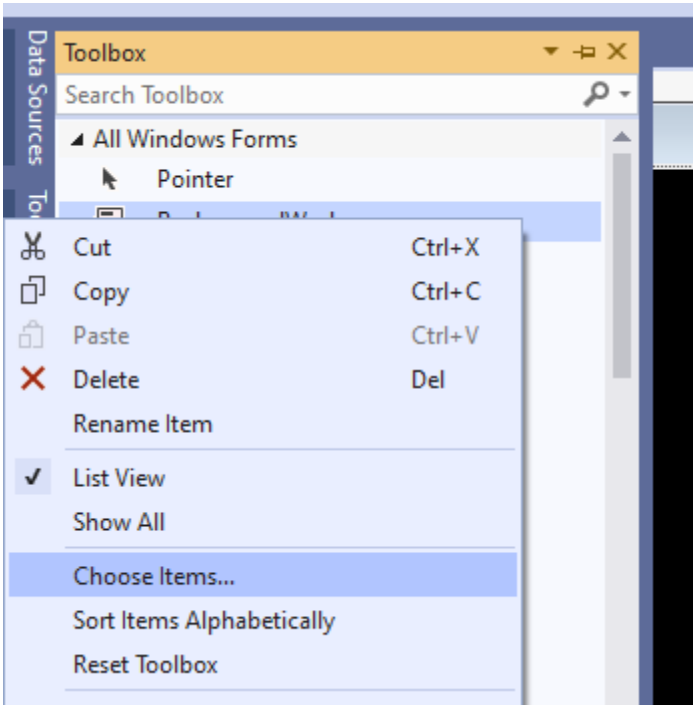
```
private void lbMedia_SelectedIndexChanged(object sender, EventArgs e)
{
  if (lbMedia.SelectedIndex == -1)
    return;

  if (rRadio.Checked)
   wmpPlayer.URL = ((KeyValuePair<string, string>)lbMedia.SelectedItem).Value;
  else
   wmpPlayer.URL = paths[lbMedia.SelectedIndex];
}
```

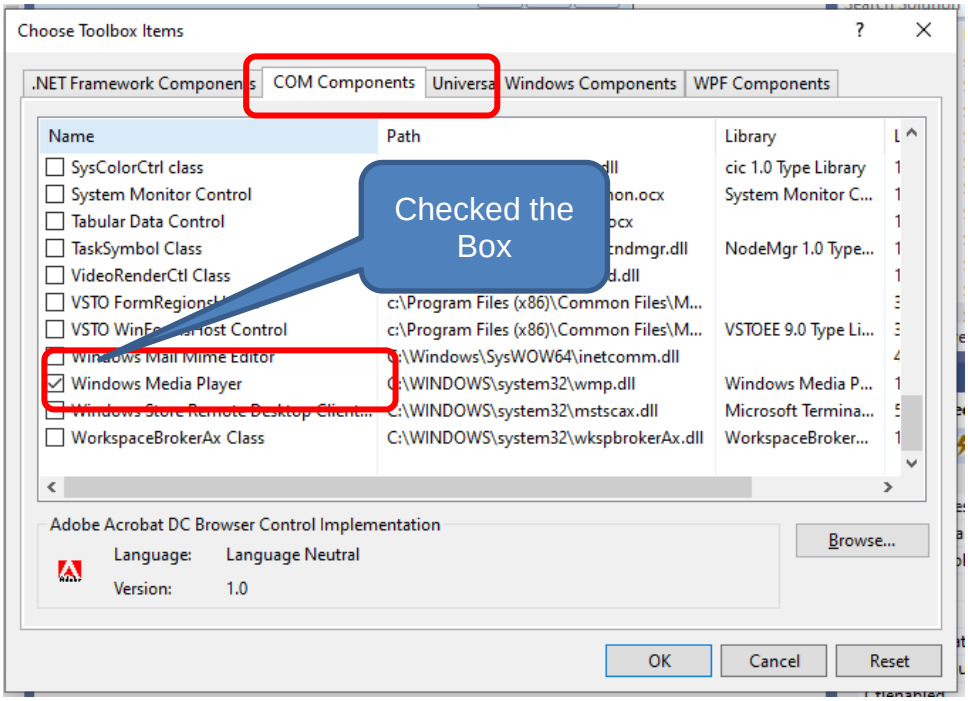
5. Build and test your application.

Add WindowsMediaPlayer to ToolBox

1. Right click on Toolbox and Select Choose Items



2. Next under COM Components, select {CHECKED} the Windows Media Player checkbox



Setting for Control Items (Sample)

If you cannot get your control items size and location correctly, use the following:

```
        /*
            form1.size          : 560, 440
            form1.size          : 0,0

            wmpPlayer.Size      : 570, 380
            wmpPlayer.Location  : 205, 10

            lbMedia.Size        : 183, 277
            lbMedia.Location    : 10, 54

            btnAdd.Size         : 182, 60
            btnAdd.Location     : 10, 334
        */

        private void rRadio_CheckedChanged(object sender, EventArgs e)
        {
            this.Size = new Size(220,440);

        }

        private void rMedia_CheckedChanged(object sender, EventArgs e)
        {
            this.Size = new Size(800,440);
        }
```

Additional Radio Station

```
      stations.Add(new KeyValuePair<string, string>("Gold 905",
@"https://19183.live.streamtheworld.com/GOLD905_SC"));
```