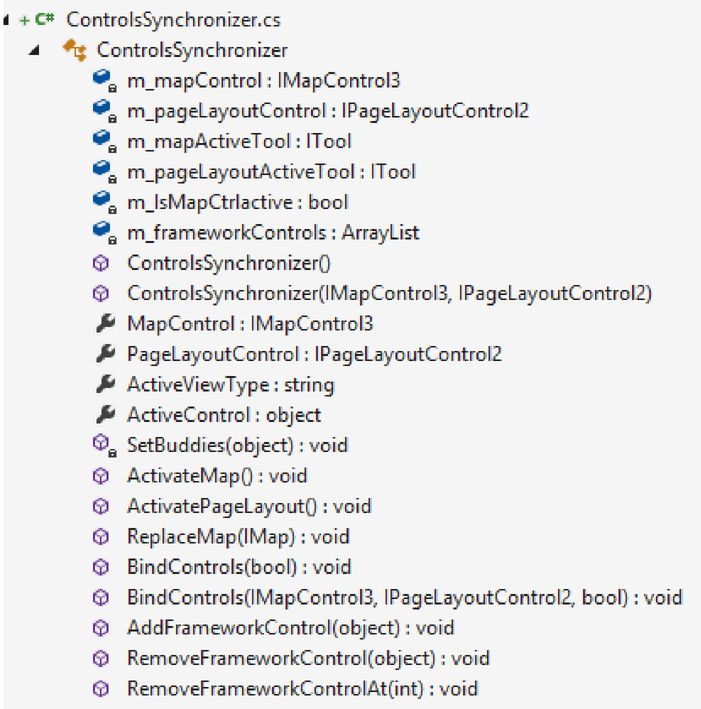
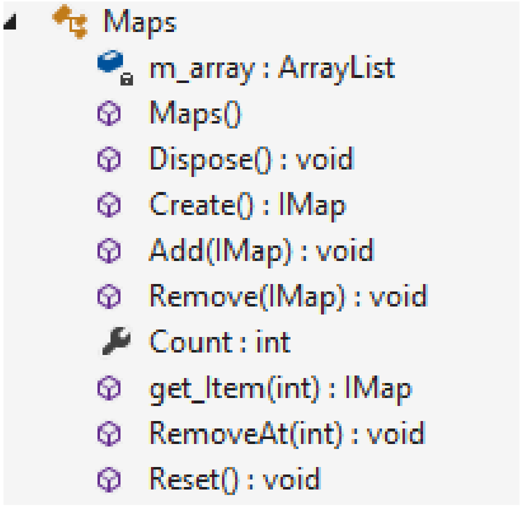
Chapter 2 Controls Sychronizer

1. Add ControlsSynchronizer class and Map Class to the project, copy the code from “ControlsSynchronizer.cs”, “Maps.cs”.

The following graphs show the member and property of the class:



```
ControlsSynchronizer.cs
  ControlsSynchronizer
      m_mapControl : IMapControl3
      m_pageLayoutControl : IPageLayoutControl2
      m_mapActiveTool : ITool
      m_pageLayoutActiveTool : ITool
      m_IsMapCtrlactive : bool
      m_frameworkControls : ArrayList
      ControlsSynchronizer()
      ControlsSynchronizer(IMapControl3, IPageLayoutControl2)
      MapControl : IMapControl3
      PageLayoutControl : IPageLayoutControl2
      ActiveViewType : string
      ActiveControl : object
      SetBuddies(object) : void
      ActivateMap() : void
      ActivatePageLayout() : void
      ReplaceMap(IMap) : void
      BindControls(bool) : void
      BindControls(IMapControl3, IPageLayoutControl2, bool) : void
      AddFrameworkControl(object) : void
      RemoveFrameworkControl(object) : void
      RemoveFrameworkControlAt(int) : void
```

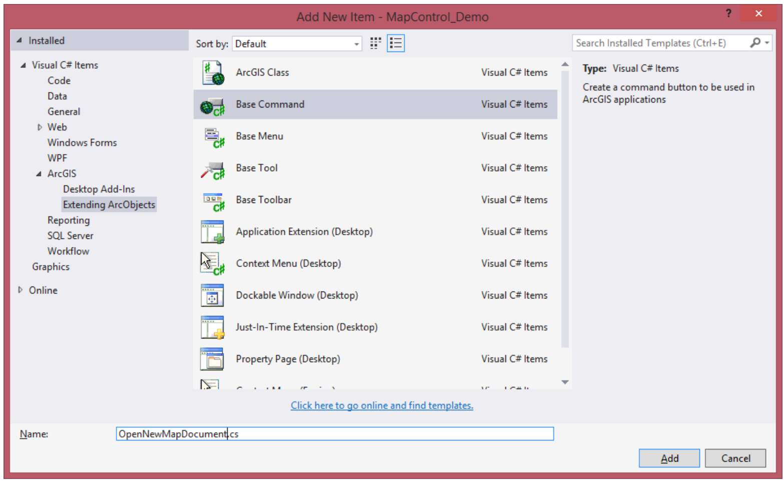


```
Maps
    m_array : ArrayList
    Maps()
    Dispose() : void
    Create() : IMap
    Add(IMap) : void
    Remove(IMap) : void
    Count : int
    get_Item(int) : IMap
    RemoveAt(int) : void
    Reset() : void
```

2. New a class named OpenNewMapDocument:

Because the default open icon will not synchronize the data view and layout view, so we have to rewrite an OpenNewMapDocument class to implement open map function.

In the Dialog, choose Extending ArcObjects -> Base Command, then input file name: “OpenNewMapDocument.cs”

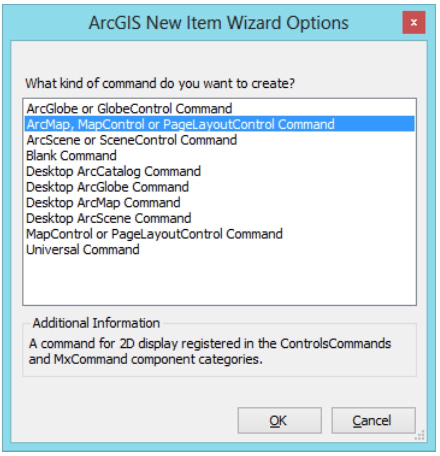


Add New Item - MapControl_Demo
Installed
Visual C# Items
Code
Data
General
Web
Windows Forms
WPF
ArcGIS
Desktop Add-Ins
Extending ArcObjects
Reporting
SQL Server
Workflow
Graphics
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
ArcGIS Class — Visual C# Items
Base Command — Visual C# Items
Base Menu — Visual C# Items
Base Tool — Visual C# Items
Base Toolbar — Visual C# Items
Application Extension (Desktop) — Visual C# Items
Context Menu (Desktop) — Visual C# Items
Dockable Window (Desktop) — Visual C# Items
Just-In-Time Extension (Desktop) — Visual C# Items
Property Page (Desktop) — Visual C# Items
Click here to go online and find templates.
Add
Cancel

Type: Visual C# Items

Create a command button to be used in ArcGIS applications

Name: OpenNewMapDocument.cs



ArcGIS New Item Wizard Options
ArcGlobe or GlobeControl Command
ArcMap, MapControl or PageLayoutControl Command
ArcScene or SceneControl Command
Blank Command
Desktop ArcCatalog Command
Desktop ArcGlobe Command
Desktop ArcMap Command
Desktop ArcScene Command
MapControl or PageLayoutControl Command
Universal Command
OK
Cancel

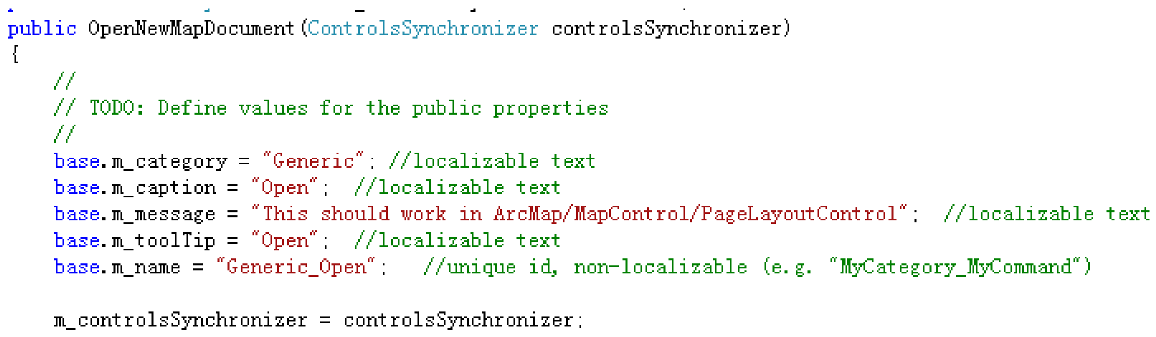
What kind of command do you want to create? ArcMap, MapControl or PageLayoutControl Command

Additional Information

A command for 2D display registered in the ControlsCommands and MxCommand component categories.

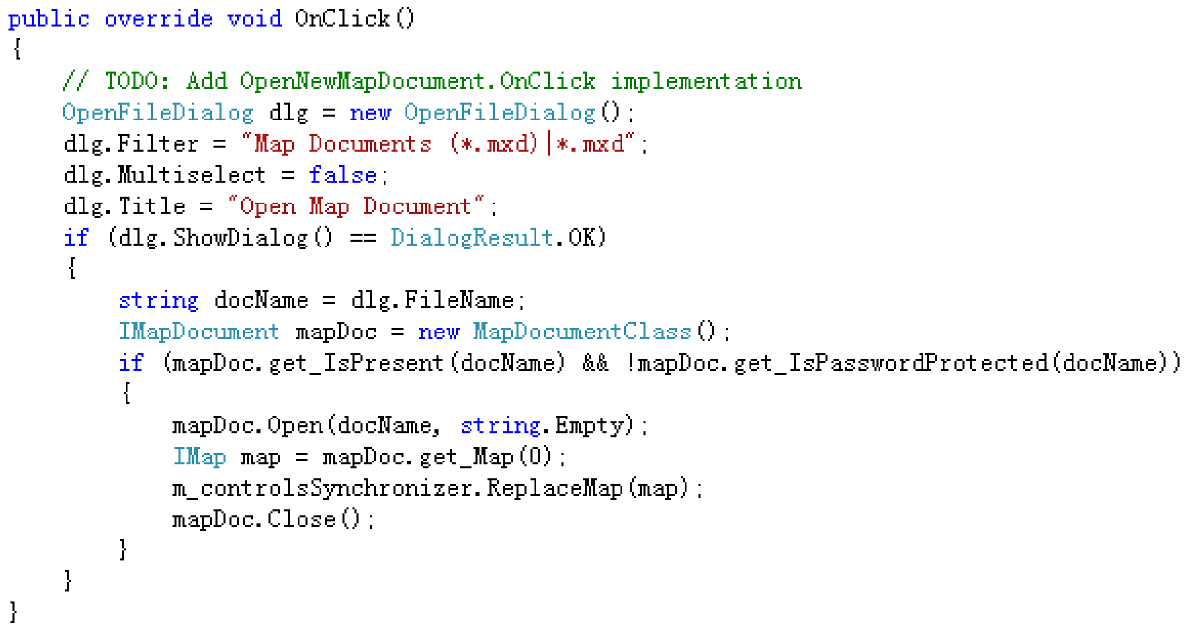
In the new Item Wizard, choose” Command that works in ArcMap/Map/PageLayout”.

- Add class member:
  ```
  private ControlsSynchronizer m_controlsSynchronizer = null;
  ```
- Change the default construct function:



```
public OpenNewMapDocument(ControlsSynchronizer controlsSynchronizer)
{
    //
    // TODO: Define values for the public properties
    //
    base.m_category = "Generic"; //localizable text
    base.m_caption = "Open";  //localizable text
    base.m_message = "This should work in ArcMap/MapControl/PageLayoutControl";  //localizable text
    base.m_toolTip = "Open";  //localizable text
    base.m_name = "Generic_Open";   //unique id, non-localizable (e.g. "MyCategory_MyCommand")

    m_controlsSynchronizer = controlsSynchronizer;
```

- Add Onclick implementation:



```
public override void OnClick()
{
    // TODO: Add OpenNewMapDocument.OnClick implementation
    OpenFileDialog dlg = new OpenFileDialog();
    dlg.Filter = "Map Documents (*.mxd)|*.mxd";
    dlg.Multiselect = false;
    dlg.Title = "Open Map Document";
    if (dlg.ShowDialog() == DialogResult.OK)
    {
        string docName = dlg.FileName;
        IMapDocument mapDoc = new MapDocumentClass();
        if (mapDoc.get_IsPresent(docName) && !mapDoc.get_IsPasswordProtected(docName))
        {
            mapDoc.Open(docName, string.Empty);
            IMap map = mapDoc.get_Map(0);
            m_controlsSynchronizer.ReplaceMap(map);
            mapDoc.Close();
        }
    }
}
```

3. In the MainForm class:

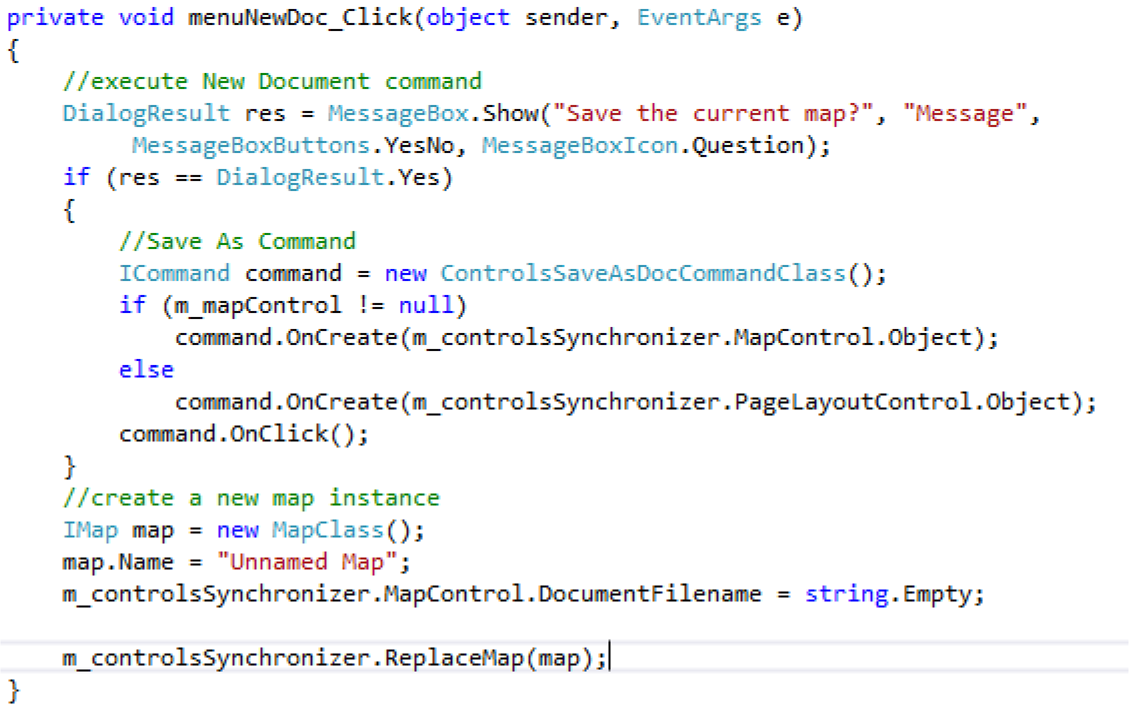
- Add class private member: ControlsSynchronizer and IPageLayoutControls:

```
public sealed partial class MainForm : Form
{
    #region class private members
    private IMapControl3 m_mapControl = null;
    private string m_mapDocumentName = string.Empty;
    IMapDocument m_MapDocument = new MapDocumentClass();
    private ControlsSynchronizer m_controlsSynchronizer = null;
    private IPageLayoutControl3 m_pageLayoutControl = null;//Layout View
    #endregion
```

- Load method implementation:

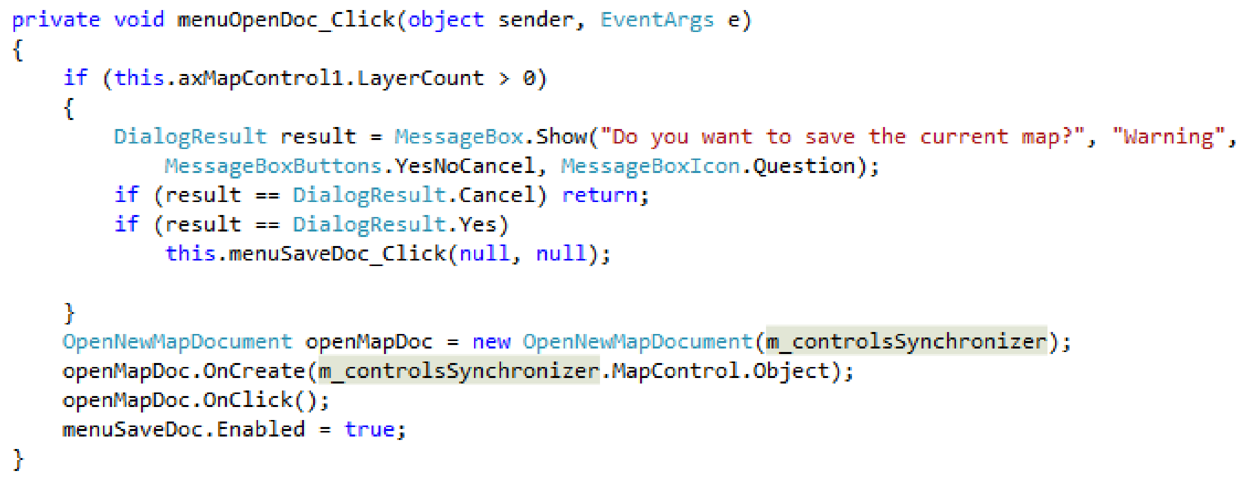
```
        private void MainForm_Load(object sender, EventArgs e)
        {
            //get the MapControl
            m_mapControl = (IMapControl3)axMapControl1.Object;
            //disable the Save menu (since there is no document yet)
            menuSaveDoc.Enabled = false;
            m_pageLayoutControl = (IPageLayoutControl3)axPageLayoutControl1.Object;
            //Initialize controls synchronization calss
            m_controlsSynchronizer = new ControlsSynchronizer(m_mapControl,
m_pageLayoutControl);
            //Bind MapControl and PageLayoutControl(Point to the same Map), Set
MapControl as the active Control
            m_controlsSynchronizer.BindControls(true);
            //In order to switch MapControl and PageLayoutControl, we need to add
Framework Control
            m_controlsSynchronizer.AddFrameworkControl(axToolbarControl1.Object);
            m_controlsSynchronizer.AddFrameworkControl(this.axTOCControl1.Object);

            OpenNewMapDocument openMapDoc = new
OpenNewMapDocument(m_controlsSynchronizer);
            axToolbarControl1.AddItem(openMapDoc, -1, 0, false, -1,
esriCommandStyles.esriCommandStyleIconOnly);
        }
```

- Add follow code in the menuNewDoc_Click method:



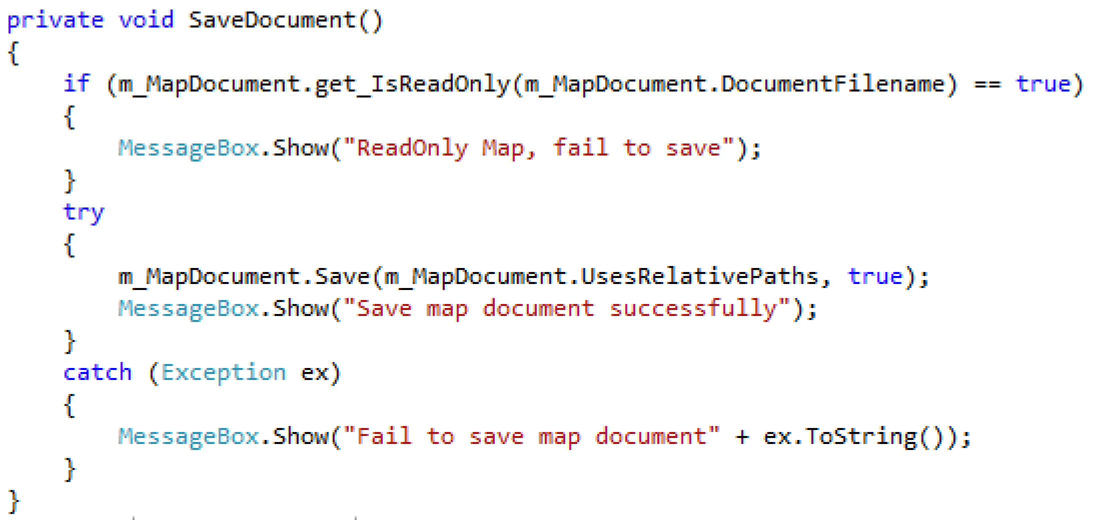
```
private void menuNewDoc_Click(object sender, EventArgs e)
{
    //execute New Document command
    DialogResult res = MessageBox.Show("Save the current map?", "Message",
         MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (res == DialogResult.Yes)
    {
        //Save As Command
        ICommand command = new ControlsSaveAsDocCommandClass();
        if (m_mapControl != null)
            command.OnCreate(m_controlsSynchronizer.MapControl.Object);
        else
            command.OnCreate(m_controlsSynchronizer.PageLayoutControl.Object);
        command.OnClick();
    }
    //create a new map instance
    IMap map = new MapClass();
    map.Name = "Unnamed Map";
    m_controlsSynchronizer.MapControl.DocumentFilename = string.Empty;

    m_controlsSynchronizer.ReplaceMap(map);
}
```

- Add follow code in the menuOpenDoc_Click method:



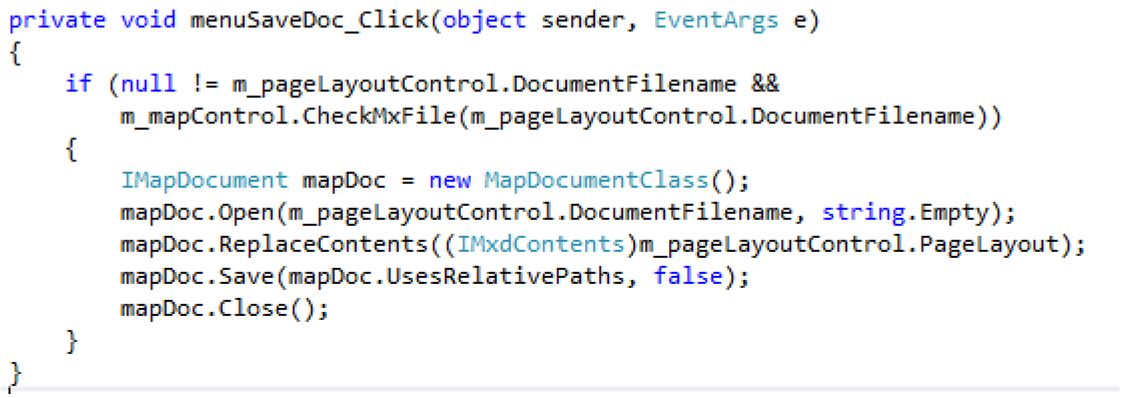
```
private void menuOpenDoc_Click(object sender, EventArgs e)
{
    if (this.axMapControl1.LayerCount > 0)
    {
        DialogResult result = MessageBox.Show("Do you want to save the current map?", "Warning",
            MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
        if (result == DialogResult.Cancel) return;
        if (result == DialogResult.Yes)
            this.menuSaveDoc_Click(null, null);

    }
    OpenNewMapDocument openMapDoc = new OpenNewMapDocument(m_controlsSynchronizer);
    openMapDoc.OnCreate(m_controlsSynchronizer.MapControl.Object);
    openMapDoc.OnClick();
    menuSaveDoc.Enabled = true;
}
```

- Add SaveDocument method:



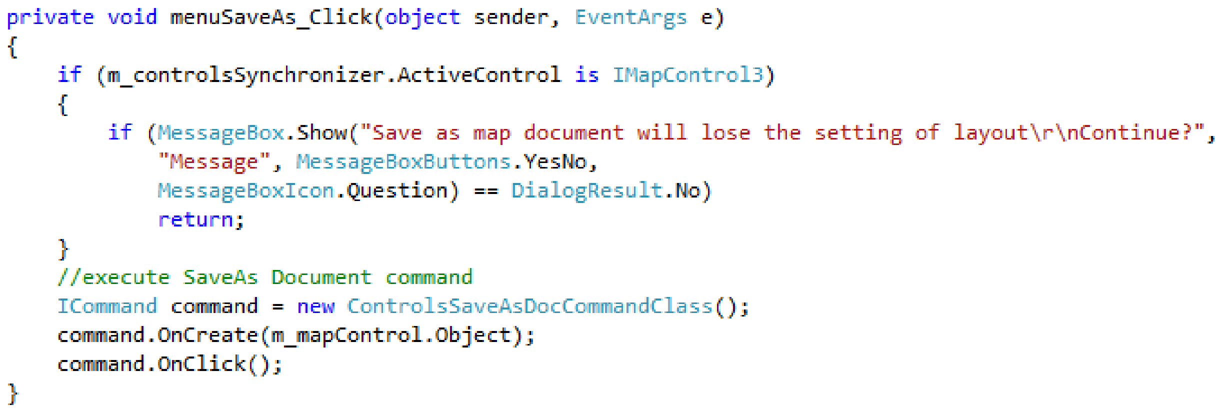
```
private void SaveDocument()
{
    if (m_MapDocument.get_IsReadOnly(m_MapDocument.DocumentFilename) == true)
    {
        MessageBox.Show("ReadOnly Map, fail to save");
    }
    try
    {
        m_MapDocument.Save(m_MapDocument.UsesRelativePaths, true);
        MessageBox.Show("Save map document successfully");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Fail to save map document" + ex.ToString());
    }
}
```

- Add follow code in the menuSaveDoc_Click method:



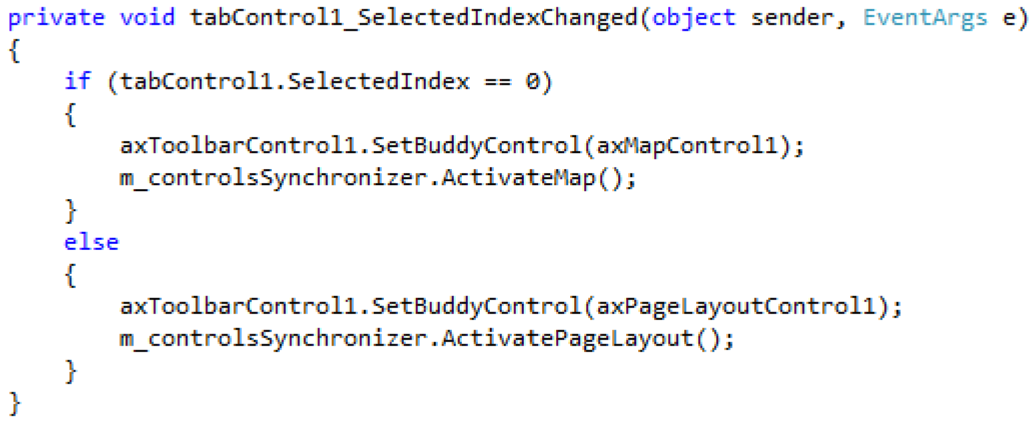
```
private void menuSaveDoc_Click(object sender, EventArgs e)
{
    if (null != m_pageLayoutControl.DocumentFilename &&
        m_mapControl.CheckMxFile(m_pageLayoutControl.DocumentFilename))
    {
        IMapDocument mapDoc = new MapDocumentClass();
        mapDoc.Open(m_pageLayoutControl.DocumentFilename, string.Empty);
        mapDoc.ReplaceContents((IMxdContents)m_pageLayoutControl.PageLayout);
        mapDoc.Save(mapDoc.UsesRelativePaths, false);
        mapDoc.Close();
    }
}
```

- Add follow code in the menuSaveAsDoc_Click method:



```
private void menuSaveAs_Click(object sender, EventArgs e)
{
    if (m_controlsSynchronizer.ActiveControl is IMapControl3)
    {
        if (MessageBox.Show("Save as map document will lose the setting of layout\r\nContinue?",
            "Message", MessageBoxButtons.YesNo,
            MessageBoxIcon.Question) == DialogResult.No)
            return;
    }
    //execute SaveAs Document command
    ICommand command = new ControlsSaveAsDocCommandClass();
    command.OnCreate(m_mapControl.Object);
    command.OnClick();
}
```

- Implement SelectedIndexChanged method of tabControl:



```
private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (tabControl1.SelectedIndex == 0)
    {
        axToolbarControl1.SetBuddyControl(axMapControl1);
        m_controlsSynchronizer.ActivateMap();
    }
    else
    {
        axToolbarControl1.SetBuddyControl(axPageLayoutControl1);
        m_controlsSynchronizer.ActivatePageLayout();
    }
}
```

Now, the synchronize between PageLayoutControl and MapControl has been finished. When changing the item of one view, the other one will also change.