**Application of Ink drawing board**

**Overview**

Windows 10 provides API of drawing board application for developers. Developers can construct the sketchpad categories of applications through the basic API. The users can complete the drawing by touching the pen and fingers, and can set up the brush thickness, the brush color and much more drawing resource to complete the creation. Using the remote communication interface provided by Microsoft, developers can easily construct a remote distributed application, to provide the functions of multiple people drawing for users.

**Through the samples you will learn:**

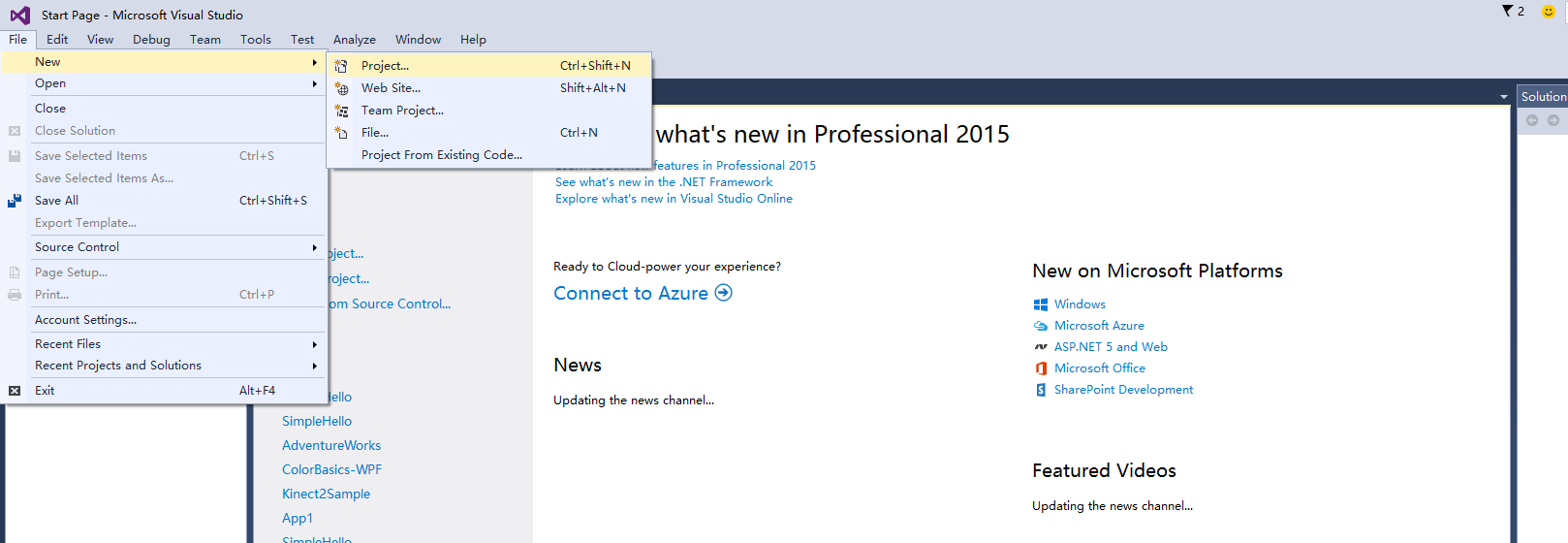
* Basic drawing board. Brush type, brush thickness, brush color, picture loading and open / close touching mode settings;
* Handwriting recognition. Construction of text recognizer and recognition of handwritten characters;
* Ink editor. Cutting, copying and pasting of drawing;
* Multi-devices synchronizations. Multi device collaborative graphics;
* Else. Application supports multi language;

**Challenge**

We will use Visual Studio 2015 to create a new project to study the API related with drawing board.

Run Visual Studio 2015 from the desktop or start menu.

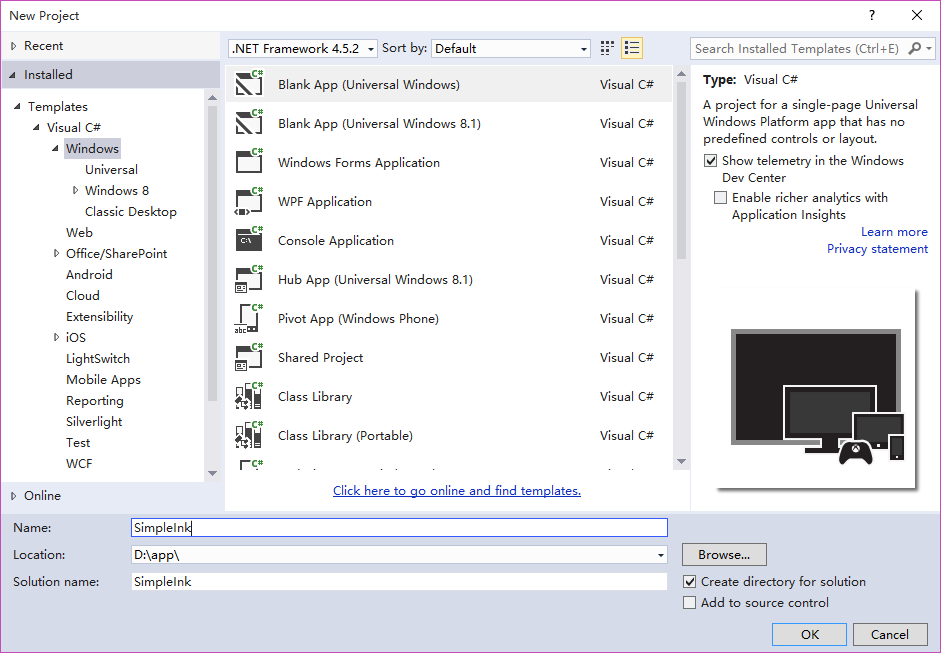
**Create a new project**



Select blank application (general Windows) type project from the project template provided.

This template is located in **Installed > Template > Visual C# > Windows > Blank application (general Windows)**.

Blank application (general Windows) is one of the most simple window applications, it can run on a number of Windows 10 platforms. For the purpose of this experiment, we proposed the project name as SimpleInk.



**Create the first scenario**

**Scenario Description**

In this scenario, the main demonstration is how to draw by using the drawing board, users can use the stylus or press the left mouse button to create on the drawing board. At the same time, the users can set the brush type, brush thickness, brush color, load the local picture, open / close the touching and other operations.

**Knowledge keys**

**InkCanvas Control:** In simple terms, InkCanvas is the layout control to achieve ink, it is mainly used to capture handwriting through the mouse or stylus. It is the main control for this scene.

**InkDrawingAttributes Class:** Brush properties, including the brush thickness, color and text recognition, etc.

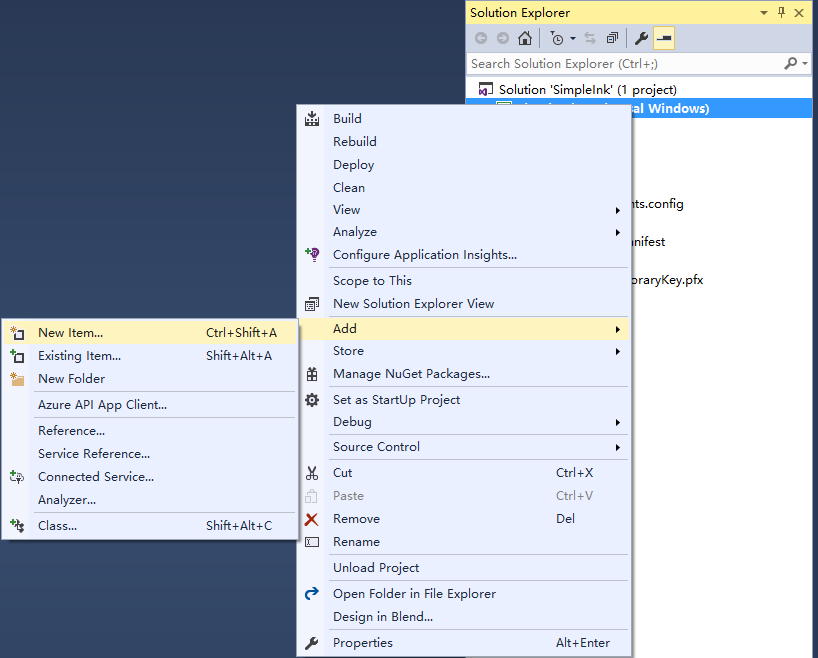
**FileSavePicker Class:** File selector, which is mainly used to save the file in the user's machine.

**FileOpenPicker Class:** File selector, which is used to select a file in one or more user machines.

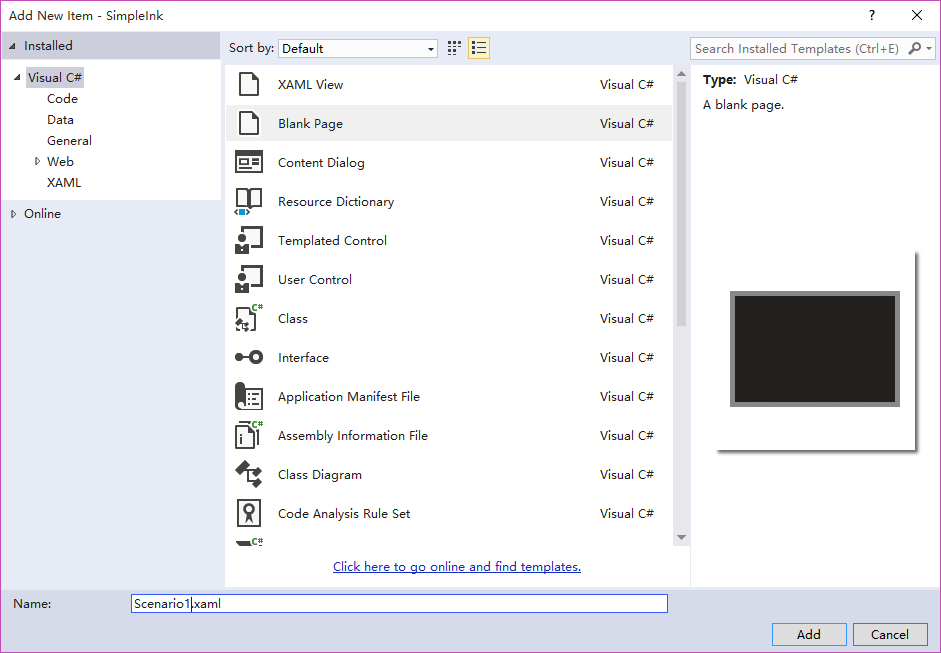
**StorageFile Class:** Represents a file. It provides information about the documents and their contents, as well as the methods to handle them.

**Create XAML view file**

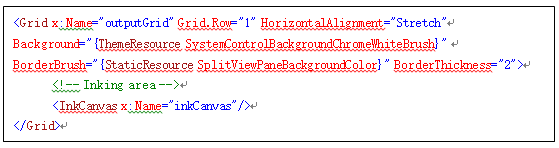
Right-click SimpleInk(unless you have other names) in solution explorer. Select “new project” from “Add” menu.



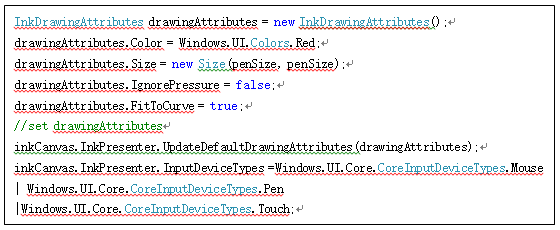
In the left Visual C# template, you can select the file type. We add a "blank page" to display our first scenario. We proposed the name as Scenario1.



Open the xaml file of Scenario1, and add the InkCanvas control, the code sample is as follows：



Open the Scenario1.xaml.cs file, and add the following code in the page constructor function：



The rendering of the drawing board is mainly achieved by the InkPresenter property of InkCanvas control.

**InkPresenter:** It is a class in Windows.UI.Input.Inking namespace, which is a canvas mainly used to display ink strokes. The ink capturing of InkCanvas is actually completed by it.

**InkDrawingAttributes:** Ink drawing properties. InkPresenter is mainly based on the set in InkDrawingAttributes to gather and capture handwriting.

The main properties of InkDrawingAttributes are as follows：

**InkDrawingAttributes.Color:** brush color;

**InkDrawingAttributes.Size:** brush thickness;

**InkDrawingAttributes.DrawAsHighlighter:** Whether use fluorescent pen drawing or not;

**InkDrawingAttributes.FitToCurve:** Whether use Bessel curve or not;

**InkDrawingAttributes.IgnorePressure:** Whether ignore the pressure or not;

**InkDrawingAttributes.PenTip:** The shape of pen point;

**InkDrawingAttributes.Matrix3x2:** 3\*2 matrix;

**Create the second scenario**

**Scenario Description**

In this scenario, it mainly demonstrated the recognition of ink.

**Knowledge Keys**

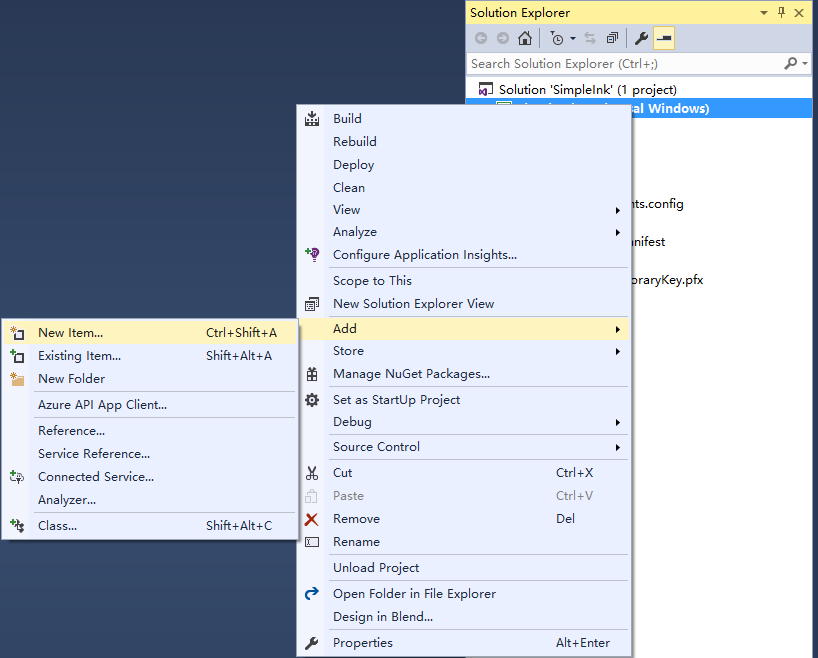
**InkCanvas Control:** For details, see the description in the first scenario.

**InkDrawingAttributes:** For details, see the description in the first scenario.

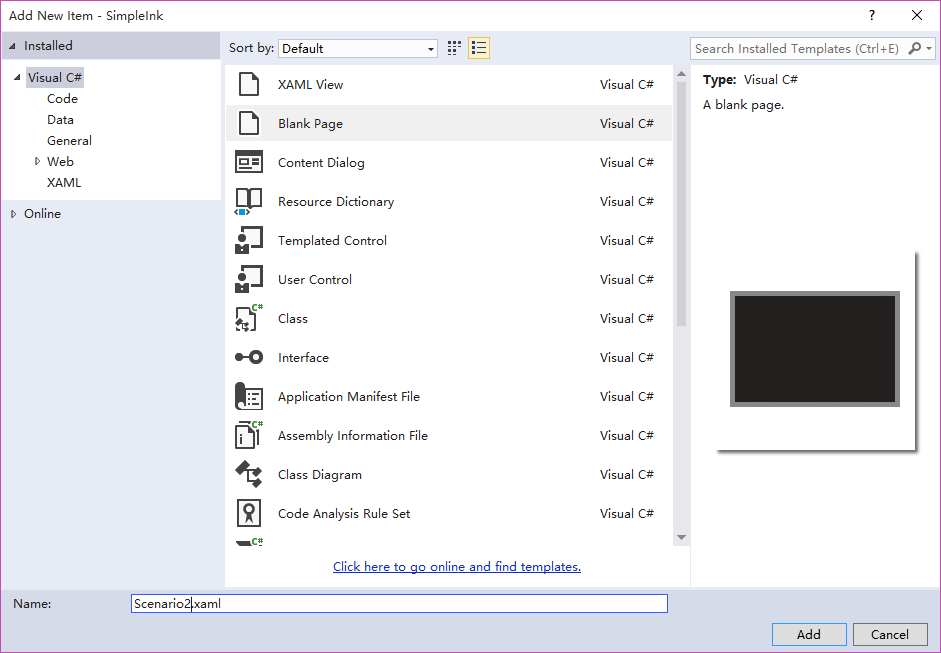
**InkRecognizerContainer:** This class is a recognition engine provided by Windows for developers, it can identify text through a group of brush ink.

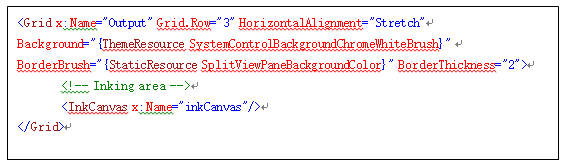
**Create XAML view file**

Right-click SimpleInk(unless you have other names) in solution explorer. Select “new project” from “Add” menu.



In the left Visual C# template, you can select the file type. We add a "blank page" to display our first scenario. We proposed the name as Scenario2.



****

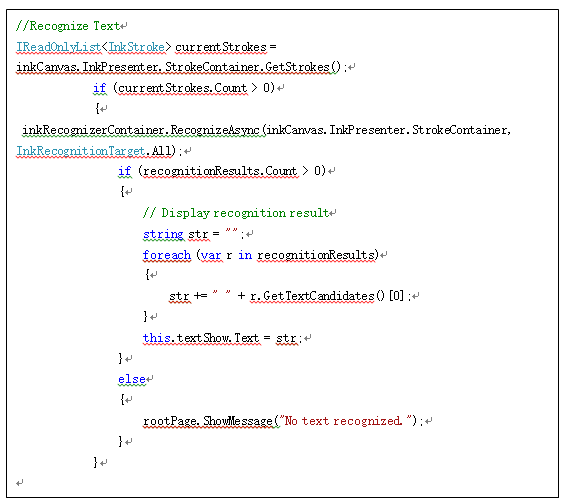
Open the xaml file of Scenario2, and add the InkCanvas control, the code sample is as follows：

Text identification code is as follows:

The main functions of InkRecognizerContainer are described as follows:

**GetRecognizers():** The identifier used for gaining the current application settings. (Bcp47 support is required for the identifier, such as "zh-CN"," Microsoft Chinese (Simplified) handwriting recognition");

**SetDefaultRecognizer(InkRecognizer recognizer):** Set the default identifier, after setting up successfully, the recognition engine will identify the text based on the language of the identifier.



**RecognizeAsync(InkStrokeContainer strokeCollection, InkRecognitionTarget recognitionTarget).** Text recognition, the example is as follows:

Var recognitionResults = await

inkRecognizerContainer.RecognizeAsync(inkCanvas.InkPresenter.StrokeContainer, InkRecognitionTarget.All)

The function returns a set of objects to identify the identification results. Each item in the collection represents a written language.

**Create the third scenario**

**Scenario Description**

In this scenario, sketchpad picture editor is demonstrated. Users can select ink, and make cut, copy, paste and other operations to ink through long pressing the right mouse button. (We only provide one page to select ink for Phone users).

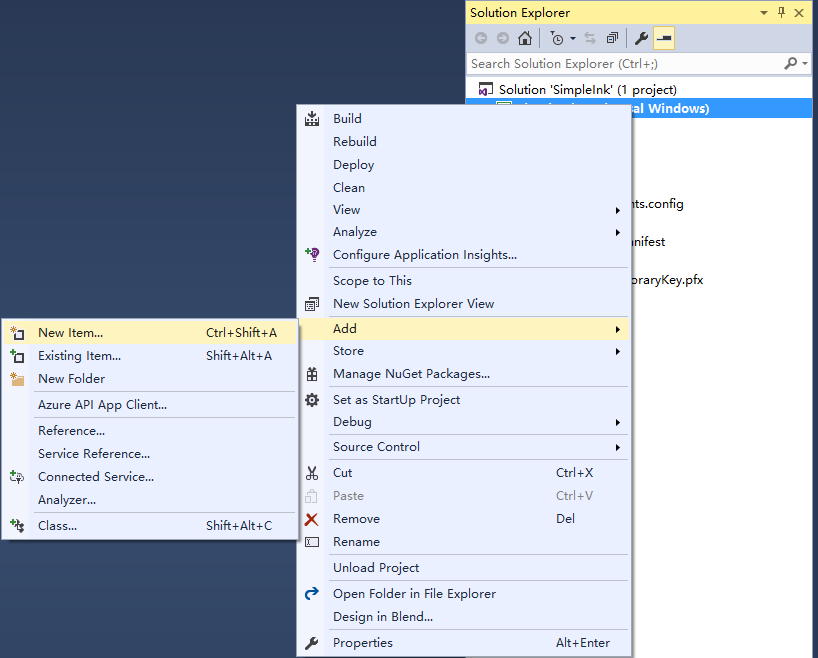
**Knowledge Keys**

**InkCanvas:** For details, see the description in the first scenario.

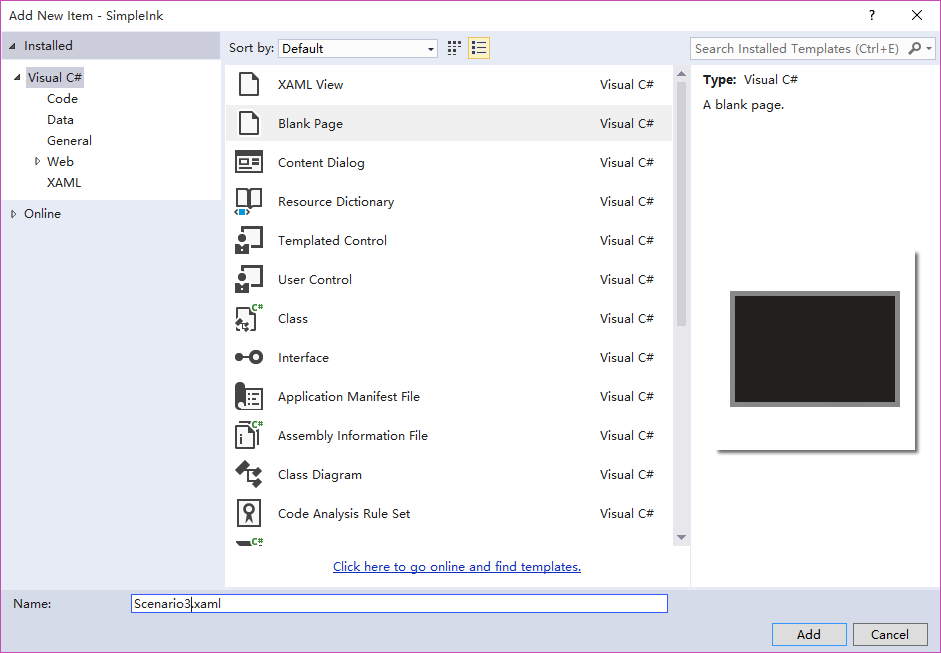
**Polyline:** PolyLine class is mainly used to draw a series of connected straight line, simply to say, is a rectangular or square.

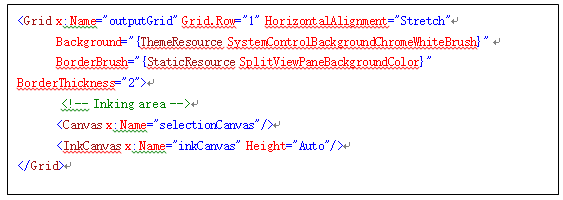
**Create XAML view file**

Right-click SimpleInk(unless you have other names) in solution explorer. Select “new project” from “Add” menu.



In the left Visual C# template, you can select the file type. We add a "blank page" to display our first scenario. We proposed the name as Scenario3.





Open the xaml file of Scenario3, and add the InkCanvas control, the code sample is as follows：

Notice: In this scenario, a Canvas is more than the first and second scenes. It is a transparent float, mainly used to display the selected picture frame. <Canvas x:Name="selectionCanvas"/>

In this scenario, the StrokeContainer attribute in InkCanvas.InkPresenter is mainly used, which belongs to InkStrokeContainer type, the main functions used are as follows:

**Clear():** Delete all information from InkStrokeContainer. Namely, clear ink;

**DeleteSelected():** Cancel the InkStroke content in InkStrokeContainer;

**SelectWithPolyLine(IEnumerable<Point> polyline):** Pitch on the contents of InkStroke in InkStrokerContainer. That is, pitch on all of the ink.

**CopySelectedToClipboard():** Select the InkStroke object and copy it to the clipboard;

**PasteFromClipboard(Point position):** Paste the InkStroke object in the clipboard to the InkStrokeContainer object to render the brush.

**AddStroke(InkStroke stroke):** Add the ink onto the clipboard.

**MoveSelected(Point translation):** According to the coordinate parameters, move the selected Stroke.

**Create the forth scenario**

**Scenario Description**

In this scenario, it mainly demonstrated the synchronization of the multi device drawing board. After filling in the IP address of the remote device, you can establish a Socket connection with the remote device to share the drawing board.

**Knowledge Keys**

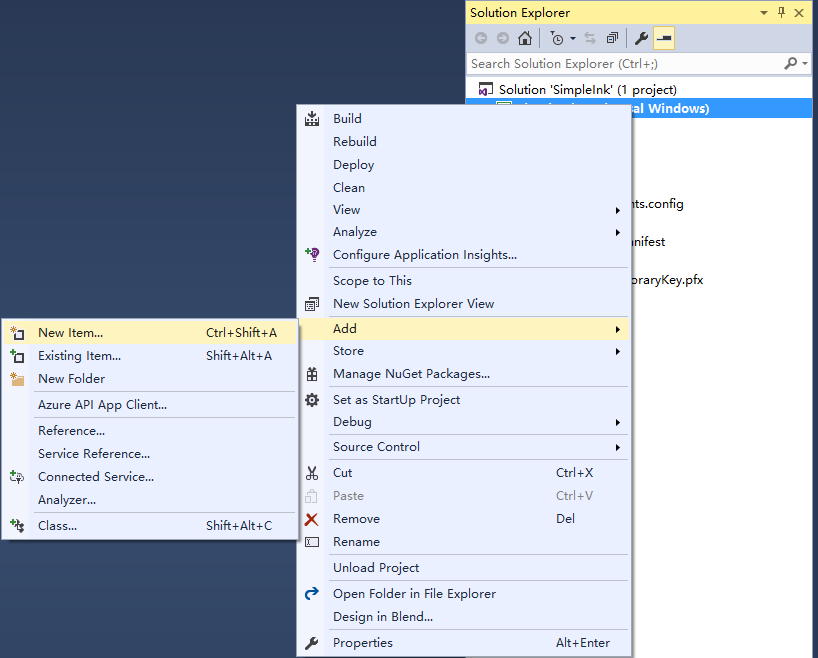
**StreamSocketListener:** To achieve the socket monitor of TCP communication through StreamSocketListener;

**StreamSocket:** To achieve the socket of TCP communication through StreamSocket;

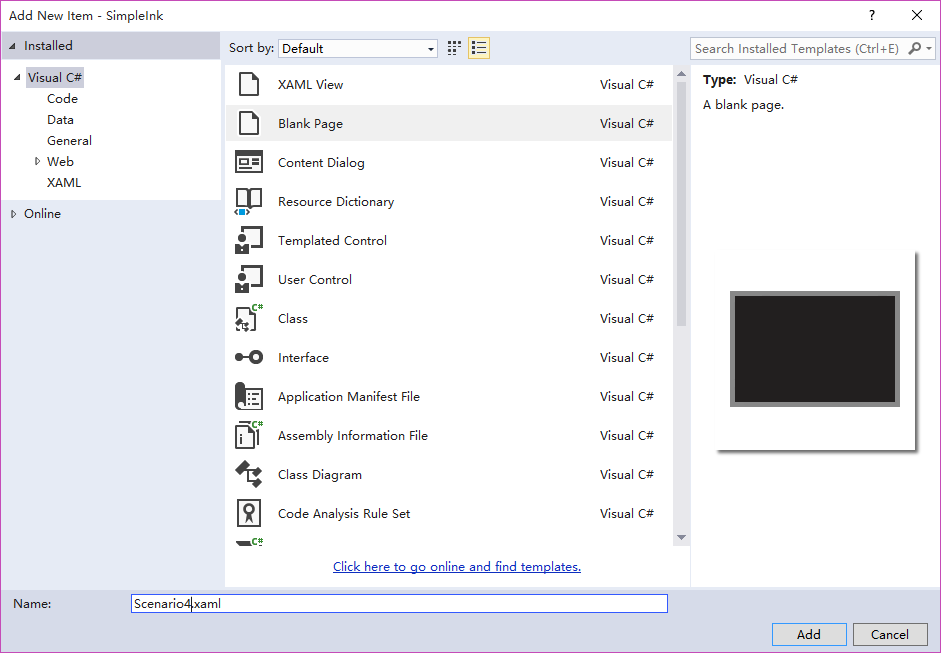
**InkStrokeContainer：**LoadAsync(IInputStream inputStream)、SaveAsync(IOutputStream outputStream);

**Create XAML view file**

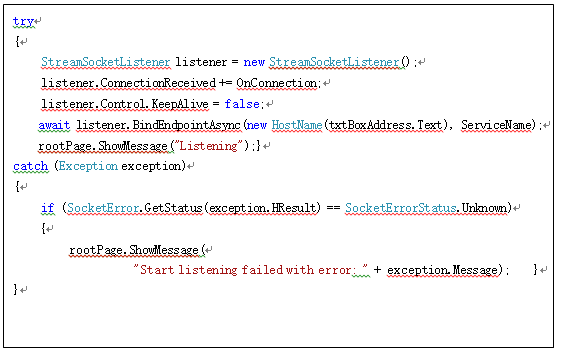
Right-click SimpleInk(unless you have other names) in solution explorer. Select “new project” from “Add” menu.



In the left Visual C# template, you can select the file type. We add a "blank page" to display our first scenario. We proposed the name as Scenario4.



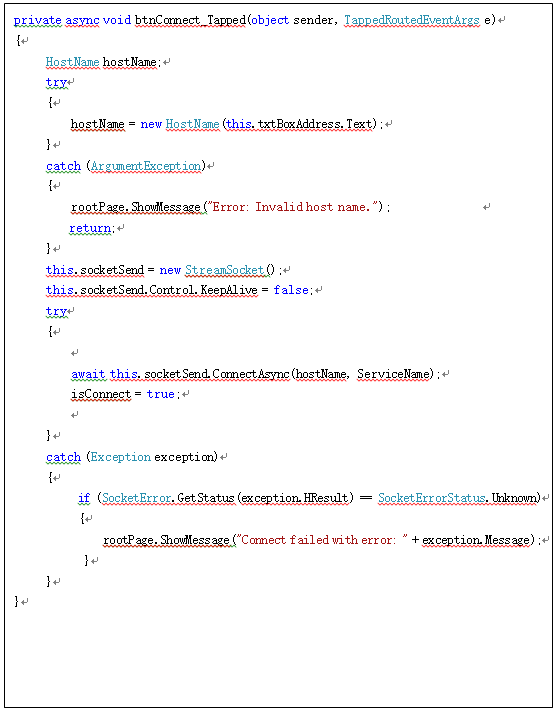
Socket monitor code samples are as follows:

****

Socket ConnectionReceived event code is as follows:



Socket connection code is as follows:

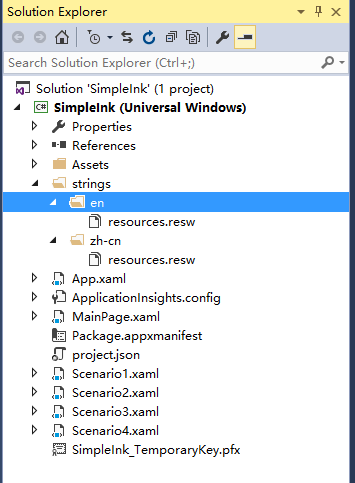


In this scenario, the display of the remote device drawing board is accomplished by the StrokeContainer class of InkCanvas. InkPresenter property, LoadAsync (IInputStream inputStream) function, and SaveAsync(IOutputStream outputStream) function.

LoadAsync: Load the InkStroke object from the memory stream to the InkStrokeContainer to display the brush.

SaveAsync: Writes the InkStroke object of InkStrokeContainer to the memory stream.

**Multi language support**



Add the strings folder, which were created in both English and Chinese resource file in solution.

en/resources.resw、zh-cn/resources.resw. English display and Chinese display for storing UI elements.

The current language setting can use the following code in Windows 10:

Windows.Globalization.ApplicationLanguages.PrimaryLanguageOverride = "zh-cn"; or

Windows.Globalization.ApplicationLanguages.PrimaryLanguageOverride = "en-US";

**PrimaryLanguageOverride：**The preferred language for the current application.