# How to use Speech Recognizer in search textbox in Win 10 UWP app

## Introduction

This sample demonstrates how to use Speech Recognizer in search textbox in Win 10 UWP app. In this sample we use continuous dictation.

## Building the sample

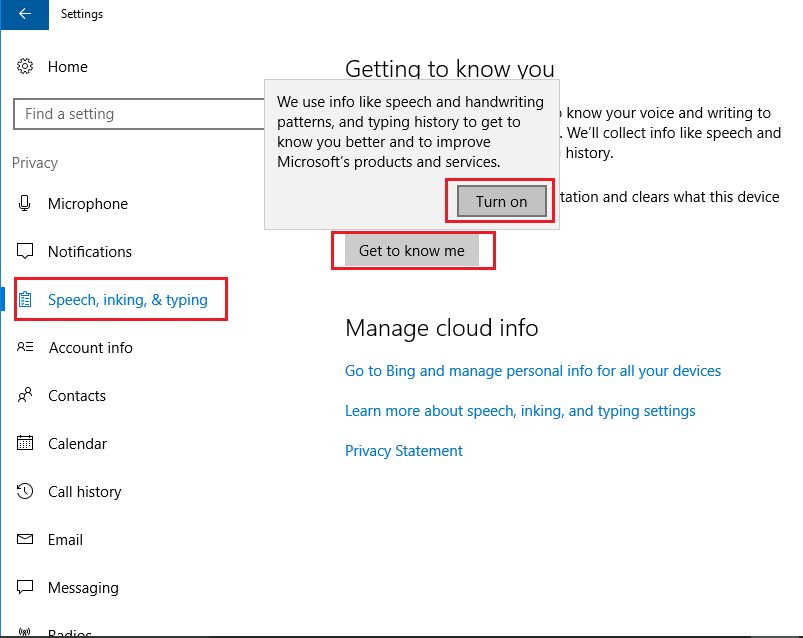
This sample should be run in Microsoft Visual Studio 2015 version and Windows 10.

Before you build the project, make sure you have installed Microsoft.NETCore.UniversalWindowsPlatform package in the project. The following steps can help you to install it:

* Open the solution CSUWPAccessibility.sln.
* Right click the project and select [Manage NuGet Packages...].
* Search Microsoft.NETCore.UniversalWindowsPlatform from the Browse tab page. Find the right package and then install it.

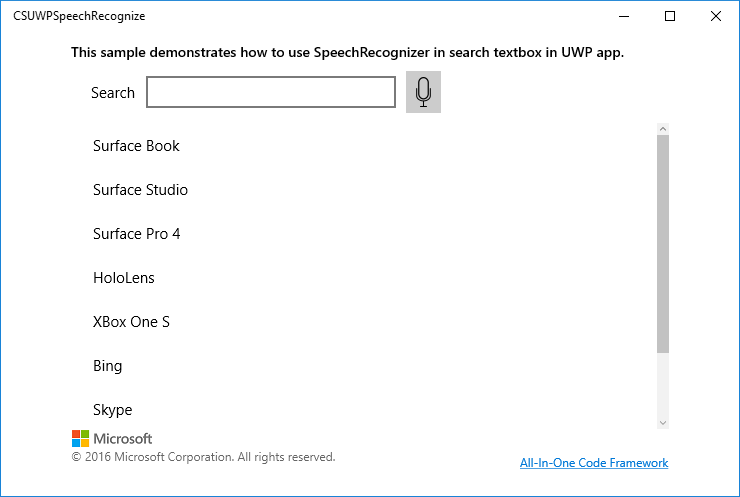
## Running the sample

Make sure you have turn on the “Get to know me” option on your PC.

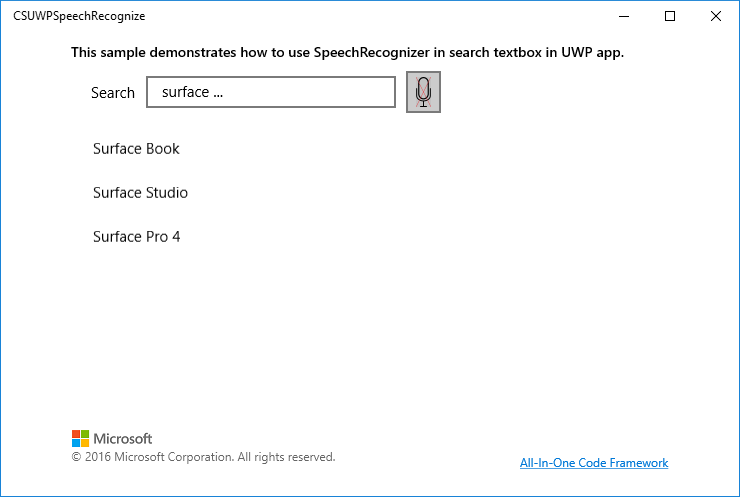


Please make sure that Local Machine which is in the target device menu from the Standard toolbar has been selected. Then do one of the following:

* Click the Start Debugging button on the toolbar.
* Click Start Debugging in the Debug menu.
* Press F5.



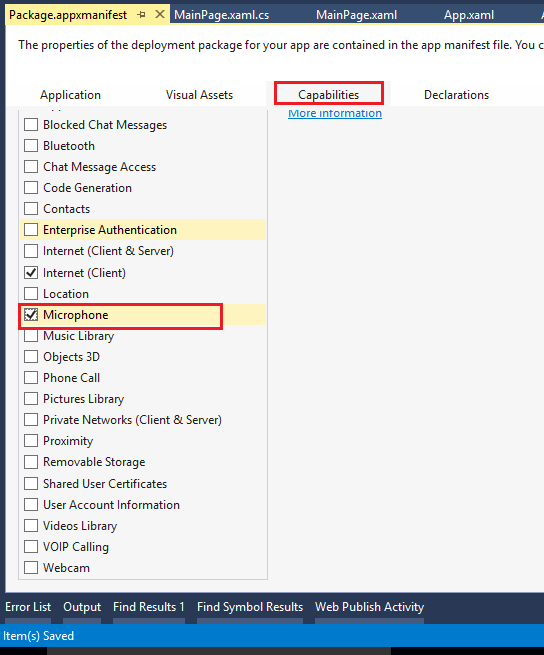
Click the button to start recognition, and say something, then click again to stop recognition, or you can wait it to time out and cancel the recognition itself.



You will see that the result in the listview has been filtered.

**Using the code**

Before writing the code, you should make some change with the manifest file.



protected async override void OnNavigatedTo(NavigationEventArgs e)

{

dictatedTextBuilder = new StringBuilder();

this.dispatcher = CoreWindow.GetForCurrentThread().Dispatcher;

speechRecognizer = new SpeechRecognizer(SpeechRecognizer.SystemSpeechLanguage);

SpeechRecognitionCompilationResult result = await speechRecognizer.CompileConstraintsAsync();

speechRecognizer.ContinuousRecognitionSession.ResultGenerated += ContinuousRecognitionSession\_ResultGenerated;

speechRecognizer.ContinuousRecognitionSession.Completed += ContinuousRecognitionSession\_Completed;

speechRecognizer.HypothesisGenerated += SpeechRecognizer\_HypothesisGenerated;

base.OnNavigatedTo(e);

}

SpeechRecognizer\_HypothesisGenerated

private async void SpeechRecognizer\_HypothesisGenerated(SpeechRecognizer sender, SpeechRecognitionHypothesisGeneratedEventArgs args)

{

string hypothesis = args.Hypothesis.Text;

string textboxContent = dictatedTextBuilder.ToString() + " " + hypothesis + " ...";

await dispatcher.RunAsync(CoreDispatcherPriority.Normal, () =>

{

tbSearch.Text = textboxContent;

});

}

private async void ContinuousRecognitionSession\_Completed(SpeechContinuousRecognitionSession sender, SpeechContinuousRecognitionCompletedEventArgs args)

{

if (speechRecognizer.State == SpeechRecognizerState.Idle)

{

await dispatcher.RunAsync(CoreDispatcherPriority.Normal, () =>

{

this.imgMode.Source = new BitmapImage(new Uri("ms-appx:///Assets/mic.png"));

});

}

}

private async void ContinuousRecognitionSession\_ResultGenerated(SpeechContinuousRecognitionSession sender, SpeechContinuousRecognitionResultGeneratedEventArgs args)

{

if (args.Result.Confidence == SpeechRecognitionConfidence.Medium ||

args.Result.Confidence == SpeechRecognitionConfidence.High)

{

dictatedTextBuilder.Append(args.Result.Text + " ");

}

await dispatcher.RunAsync(CoreDispatcherPriority.Normal, () =>

{

tbSearch.Text = dictatedTextBuilder.ToString();

});

}

## More information

[Speech recognition](https://msdn.microsoft.com/en-us/windows/uwp/input-and-devices/speech-recognition)

[Continuous dictation](https://msdn.microsoft.com/en-us/windows/uwp/input-and-devices/enable-continuous-dictation)