

**Лабораторный­ практикум**

Курс: Запуск приложения при помощи голосовых команд

Сентябрь 2015 года

Содержание

**No table of contents entries found.**

Обзор

Приложения Windows 10 могут использовать интерфейс Кортана в тех областях, где доступно взаимодействие с пользователями через удобные и настраиваемые голосовые команды. В дополнение к доступу к особенностям системы Кортана может запустить ваше приложение на переднем плане или взаимодействовать с данными приложения в фоновом режиме.

Специфические для приложения голосовые команды начинаются с префикса, как правило, с имени приложения или ключевого слова, чтобы снимает необходимость устранения неоднозначности неограниченных комбинаций в отношении других приложений, которые могут иметь аналогичные команды. Вы можете определить опции произнесения имени приложения перед или после команды и выбрать схему поведения после запуска приложения.

В этой лаборатории вы создадите файл определения голосовых команд и будете добавлять команды, чтобы запустить своё приложение и настроить его вид.

# Цели

* 1. Эта лаборатория покажет вам как:
  + Создать файл определения голосовых команд
  + Запустить своё приложение, используя голосовую команду
  + Использовать переключение для выбора входящих голосовых команд
  + Передавать информацию из голосовой команды на вид
  + Переключать вид приложения по входящей голосовой команде
  + Запускать фоновую задачу из голосовой команды
  + Возвращать письменные и речевые ответы Кортаны из своего приложения

# Системные требования

* 1. Чтобы выполнить этот практикум, необходимо обладать следующим набором программных инструментов:
  + Microsoft Windows 10 настраивается на один из языков и регионов, в которых поддерживается Кортана
    - В сентябре 2015 года Кортана стала доступной в следующих странах/регионах: Китай, Франция, Германия, Италия, Испания, Соединённое Королевство и Соединённые Штаты. Кортана доступна на следующих языках: Китайский (упрощённый), английский (Великобритания), английский (США), французский, итальянский, немецкий и испанский языки
    - Для того, чтобы пользоваться Кортаной, необходимо произвести все настройки на один и тот же язык:
      * Языки (язык вашего устройства)
      * Разговорный язык (должен быть установлен языковой пакет)
      * Страна или регион
  + Microsoft Visual Studio 2015

# Настройка

* 1. Вы должны осуществить следующие шаги для подготовки своего компьютера для этого курса:
  2. Установите Microsoft Windows 10.
  3. Установите Microsoft Visual Studio 2015.

# Упражнения

* 1. Этот лабораторный практикум включает следующие упражнения:
  2. Запуск при помощи голосовых команд
  3. Используйте голосовую команду, чтобы изменить вид приложения
  4. Ответьте на голосовую команду при помощи фоновой задачи
  5. Расчётное время для завершения практикума: **От 30 до 45 минут**.

Упражнение1: Запуск при помощи голосовых команд

1. Голосовые команды представляют удобную альтернативу ручному запуску вашего приложения. Для реализации голосовых команд, создайте простую схему с помощью команды запуска вашего приложения и соответствующего ответа Кортаны. Зарегистрируйте команды с помощью Кортаны и используйте стартовый регистр OnActivated, чтобы начать активацию вашего приложения на переднем плане.

Задача 1 – Создать шаблон приложения Universal Windows

Начнём с создания проекта из шаблона пустого приложения.

1. В новой версии Visual Studio 2015 выберите **File (Файл) -> New (Новый) -> Project (Проект)**, чтобы открыть диалоговое окно New Project (Новый проект). Далее **Installed (Установленное) > Templates (Шаблоны) > Visual C# > Windows > Universal**, а затем выберите шаблон **Blank App** приложения (Universal Windows).
2. Назовите свой проект **SpeechRecognition** и выберите местоположение файловой системы, в которой вы сохраните результаты прохождения практикума. На диске **C** создана папка под именем **"HOL"**, информация о которой будет представлена в скриншотах во время прохождения всех практикумов.

Не изменяйте настройки, установленные для **Create new solution (Создания нового решения)** и **Create directory for solution (Создания папки для решения)**. Вы можете снять галочки как с **Add to source control (Добавить в систему контроля версий)**, так и с **Show telemetry in the Windows Dev Center (Отобразить телеметрию в Windows Dev Center)**, если не хотите обновлять версию своей работы или использовать инструмент Application Insights. Нажмите **OK** для создания проекта.

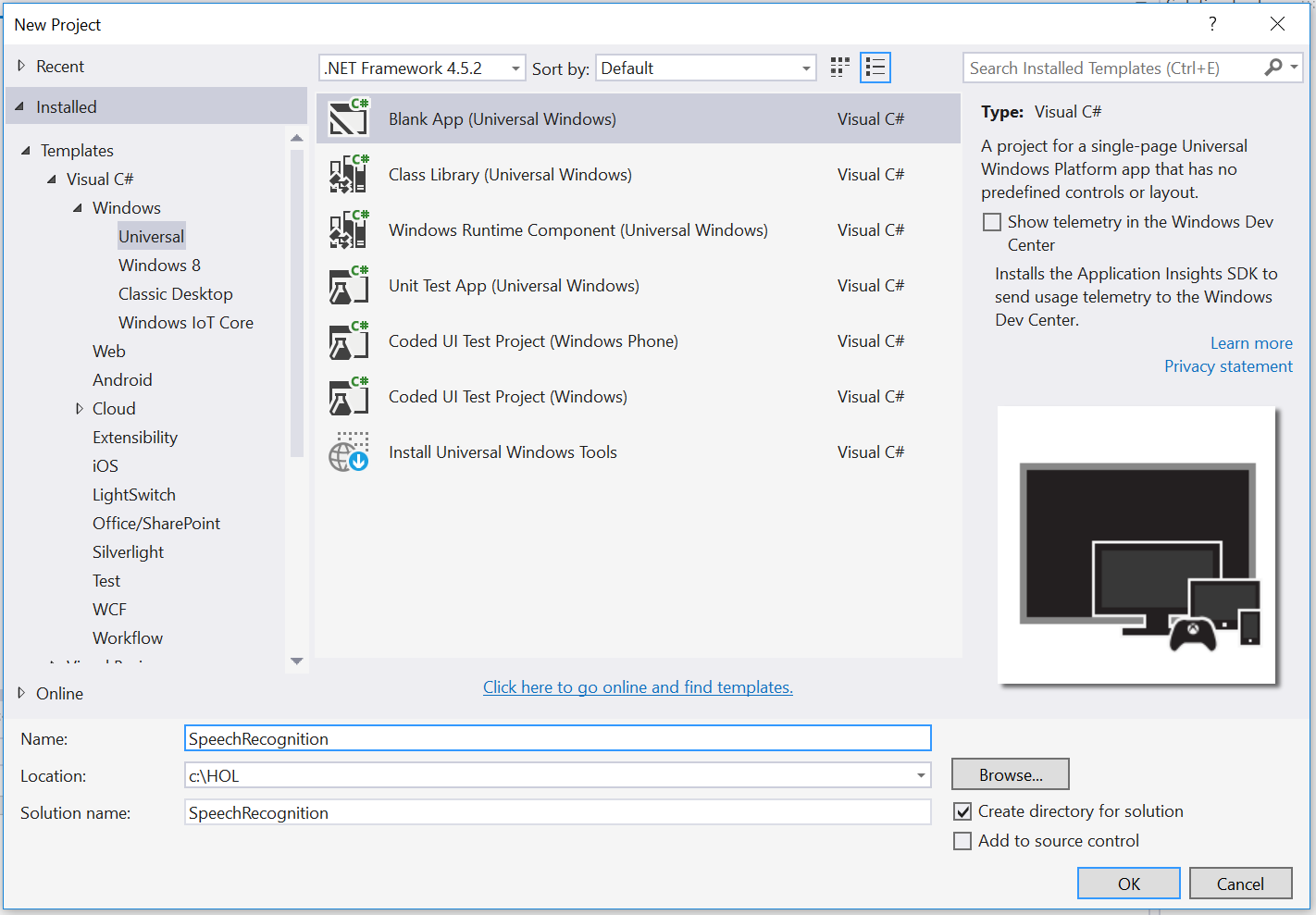


Рисунок 1

Создание нового проекта приложения в Visual Studio 2015.

1. Настройте Текущую конфигурацию решения на **Отладку** и Платформу решений в соответствии с **x86**. Выберите **Локальный компьютер** из выпадающего меню Цели отладки.



Рисунок 2

* + 1. Сконфигурируйте свое приложение таким образом, чтобы оно запускалось на Локальном компьютере.

1. Скомпилируйте и запустите своё приложение. Вы увидите окно шаблона приложения со счетчиком частоты кадров, активированном по умолчанию для отладки.



Рисунок 3

Пустое универсальное приложение, запущенное в режиме настольного компьютера.

* 1. **Примечание:** Счетчик частоты кадров является инструментом, используемым в процессе отладки, который помогает следить за производительностью вашего приложения. Он полезен для тех приложений, которые требуют интенсивной графической обработки, однако не подходит для простых приложений, которые будут создаваться вами в данном практическом курсе.
  2. В шаблоне пустого приложения директива препроцессора для активации или отключения счетчика частоты кадров находится на **App.xaml.cs**. Счетчик частоты кадров может перекрывать или скрывать контент вашего приложения, если не свернуть его. При выполнении данных работ вы можете отключить его, установив значение **DebugSettings.EnableFrameRateCounter** как **ложное**.

1. Вернитесь к Visual Studio и отключите отладку.

Задача 2 – Создайте файл определения голосовых команд

Схема голосовой команды определена в XML-файле. В этой задаче вы создадите простую схему с помощью команды ручного запуска приложения.

1. Щёлкните правой кнопкой мыши по наименованию проекта и выберите **Add (Добавить) > New Item** (Новый элемент).
2. Выберите тип XML-файлов и присвойте ему имя **VoiceCommands.xml**.

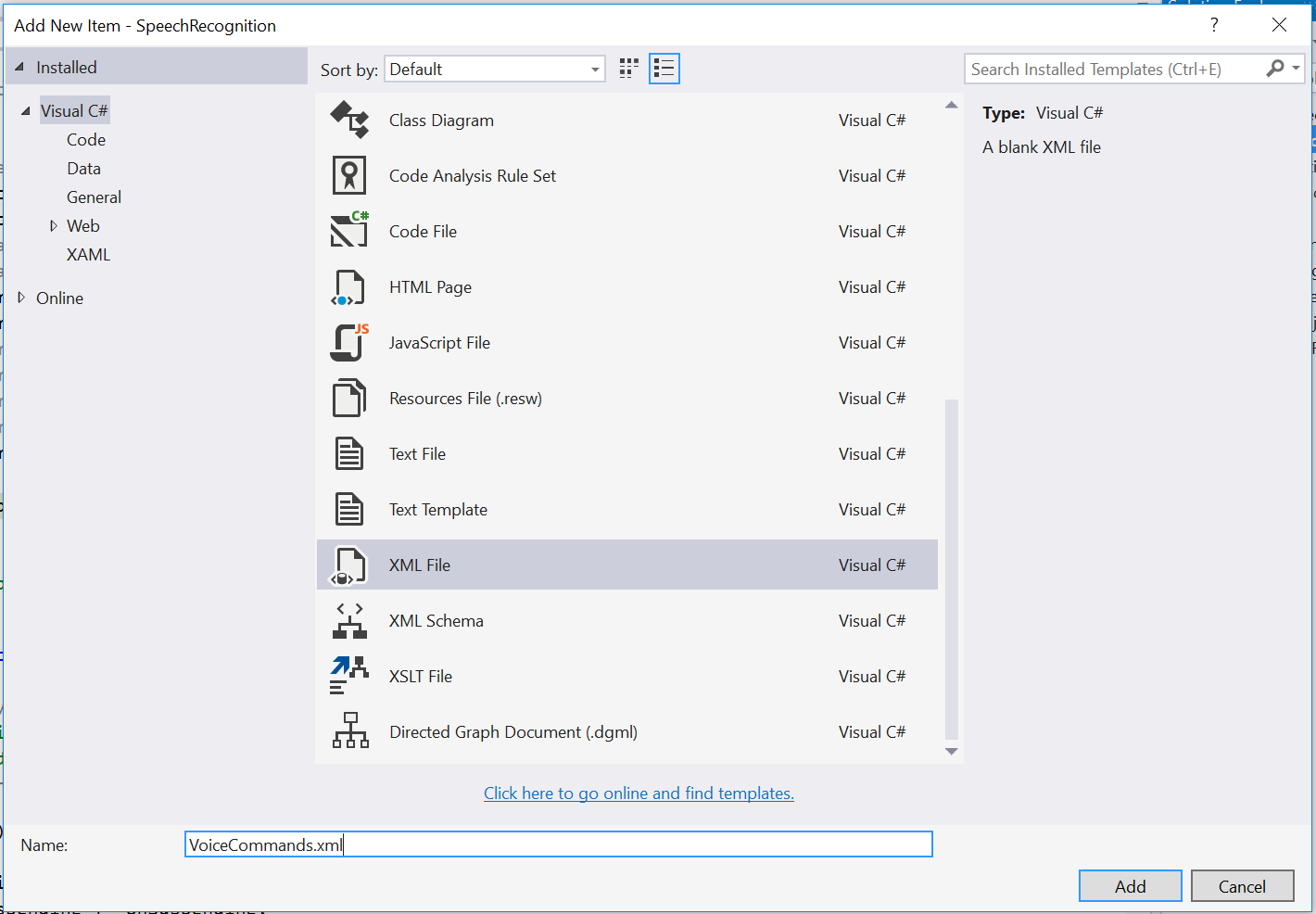


Рисунок 4

Создайте XML файл голосовых команд.

1. Откройте VoiceCommands.xml. После заголовка версии XML-файла добавьте элемент VoiceCommands и атрибут xmlns, ссылающийся на v1.2 схему голосовых команд . Добавьте элемент набора команд для языка EN-US и набор дополнительных команд для того, который будет содержать голосовые команды для вашего поддерживаемого языка (если он будет другим).
   * 1. XML
   1. <?версия XML-файла="1.0" кодирование="utf-8" ?>
   2. <VoiceCommands (голосовые команды) xmlns="http://schemas.microsoft.com/voicecommands/1.2">
   3. <CommandSet (набор команд) XML-файл: язык="en-us", Имя="HoLCommandSet\_"en-us">
   4. </CommandSet> (набор команд)
   6. <!-- Необязательный второй набор команд для поддерживаемых языков -->
   7. <!-- <CommandSet (набор команд) XML-файл: язык="de-de", Имя="HoLCommandSet\_"de-de" >

</CommandSet> -->(набор команд)

* 1. </VoiceCommands> (голосовые команды)
  2. **Примечание:** Мы добавили язык **EN-US** для этого примера, но вы можете добавить свой собственный набор команд. Например, языковый тег для Германии был бы xml:lang="de-de" (XML-**файл: язык= – "de-de"**. Список регионов и языков, которые поддерживает Кортана, находится на <http://windows.microsoft.com/en-us/windows-10/cortanas-regions-and-languages>
  3. Если вы решаете добавить другой набор команд на поддерживаемом языке, убедитесь в том, что каждый раз добавляя одну из EN-US команд во всей демо-версии, вы одновременно добавляете эквивалентную команду на этом языке.

1. Добавьте префикс команды в своё определение. Префикс команды – это слово или фраза, которую ваши пользователи могут проговаривать для того, чтобы приказать системе начать прослушивать команды, поступающие из вашего приложения.
   * 1. XML
   1. <?версия XML-файла="1.0" кодирование="utf-8" ?>
   2. <VoiceCommands (голосовые команды) xmlns="http://schemas.microsoft.com/voicecommands/1.2">
   3. <CommandSet (набор команд) XML-файл: язык="en-us", Имя="HoLCommandSet\_"en-us">
   4. <CommandPrefix (префикс команды)> -> Лабораторный практикум, </CommandPrefix>
   6. </CommandSet> (набор команд)
   7. **Примечание:** Запятая после префикса команды необязательна. Если вы решите добавить её, она будет означать небольшую паузу между префиксом команды и самой командой. При распознавании команды Кортана также кратко отобразит префикс и команду именно так, как они записаны здесь.
   8. Если вы решите добавить дополнительный набор голосовых команд, добавьте префикс команды в том языке в набор подходящих команд.
2. Наша цель - запуск приложения с помощью голосовой команды. Создайте команду запуска с помощью элемента **ListenFor (Прослушать)**, который определяет слово или слова, которые будут распознаны для этой команды, с помощью элемента **Feedback (Обратная связь)**, который определяет текст подтверждения, проговариваемый Кортаной после пользователя при распознавании команды, и элемента **Navigate (Переход)**. Добавьте элемент **Example (Пример)** для новой команды к содержащему её **CommandSet (Набору команд)**.
   * 1. XML
   1. <?версия XML-файла="1.0" кодирование="utf-8" ?>
   2. <VoiceCommands (голосовые команды) xmlns="http://schemas.microsoft.com/voicecommands/1.2">
   3. <CommandSet (набор команд) XML-файл: язык="en-us", Имя="HoLCommandSet\_"en-us">
   4. <CommandPrefix (префикс команды)> -> Лабораторный практикум, </CommandPrefix>
   5. <Пример> Запуск </Пример>
   6. <Имя команды= "LaunchApp" (Запуск приложения)>
   7. <Пример> Запуск </Пример>
   8. <Прослушать> -> запуск/</Прослушать>
   9. <Обратная связь>, Открытие вашего приложения< распознавания речи/Обратная связь>
   10. <Переход />
   11. </Команда>
   12. </CommandSet> (набор команд)
   13. </VoiceCommands> (голосовые команды)
   14. **Примечание:** Элемент "Переход" означает, что приложение запустится на переднем плане. Альтернативой запуска на переднем плане является определение компонента WinRT, который обрабатывает скрытое взаимодействие с данными приложения посредством Кортаны. Вы можете узнать больше об Определениях голосовых команд по ссылке<https://msdn.microsoft.com/en-us/library/windows/apps/dn722331.aspx>

Задача 3 – Установите определения голосовых команд

Мы установим определения голосовых команд (VCD) в переопределении **OnLaunched (Запущено)**. Не существует простого способа проверки импорта VCD, а если это самая последняя версия, то её удобно устанавливать каждый раз при запуске приложения.

1. Откройте **App.xaml.cs**, добавьте **асинхронное** ключевое слово в переопределение **OnLaunched** и добавьте следующие строки:
   * 1. C#
   1. protected override async void OnLaunched(LaunchActivatedEventArgs e)
   2. {
   3. //#если Отладка
   4. // если (System.Diagnostics.Debugger.IsAttached)
   5. // {
   6. // this.DebugSettings.EnableFrameRateCounter = true;
   7. // }
   8. //#endif
   9. var storageFile = await Windows.Storage.StorageFile.GetFileFromApplicationUriAsync(new Uri("ms-appx:///VoiceCommands.xml"));
   10. await Windows.ApplicationModel.VoiceCommands.VoiceCommandDefinitionManager.InstallCommandDefinitionsFromStorageFileAsync(storageFile);
   11. Frame rootFrame = Window.Current.Content as Frame;
   12. **Примечание:** Определение голосовых команд, (VCD) будет установлено при первом же запуске приложения способом запуска **Launched (запущено)**. Вы можете открыть своё приложение из меню Запуск для того, чтобы убедиться, что голосовые команды зарегистрированы.

Задача 4 – Активация голосовой команды

Если ваше приложение запускается через голосовую команду, то его тип запуска будет **Activated (активировано)**. Соответственно, мы будем обрабатывать входящие голосовые команды в переопределении **OnActivated**. В этой задаче вы создадите переключение для оценки команды **LaunchApp (запуск приложения)**, по которой вы перейдете к изображению MainPage (главная страница), если она использовалась для запуска приложения.

1. Создайте переопределение **OnActivated** на **App.xaml.cs**. Запуск голосовой команды имеет тип запуска **OnActivated**, так что вы уже здесь сможете обработать входящую голосовую команду.
   * 1. C#
   1. protected override void OnActivated(IActivatedEventArgs args)

{

* 1. base.OnActivated(args);
  2. }

1. Добавьте переключение для обработки случая **ActivationKind.VoiceCommand (Тип активации. Голосовая команда)**, и примените метод **HandleVoiceCommand (управление голосовой командой)**. Метод **HandleVoiceCommand(управление голосовой командой)** вы создадите на следующем этапе.
   * 1. C#
   1. protected override void OnActivated(IActivatedEventArgs args)
   2. {
   3. switch (args.Kind)
   4. {
   5. регистр ActivationKind.VoiceCommand (тип активации. Голосовая команда):
   6. HandleVoiceCommand (управление голосовой командой)(аргументы);
   7. разрыв;
   9. по умолчанию:
   10. разрыв;
   11. }
   12. base.OnActivated(args);
   13. }
2. Добавьте пространство имен**System.Diagnostics к App.xaml.cs**.
   * 1. C#
   1. using System.Diagnostics;
3. Создайте метод **HandleVoiceCommand(управление голосовой командой)**. Этот метод определяет входящую голосовую команду и применяет переключение по имени голосовой команды. Вы задали команду **LaunchApp (запуск приложения)** в своём файле определений голосовых команд в Задаче 2.
   * 1. C#
   1. private void HandleVoiceCommand(IActivatedEventArgs args)

{

* 1. var commandArgs = args as VoiceCommandActivatedEventArgs;
  2. var speechRecognitionResult = commandArgs.Result;
  3. var command = speechRecognitionResult.Text;
  4. var voiceCommandName = speechRecognitionResult.RulePath[0];
  5. var textSpoken = speechRecognitionResult.Text;
  6. Debug.WriteLine("Command: " + command);
  7. Debug.WriteLine("Text spoken: " + textSpoken);
  9. переключение (voiceCommandName) (имя голосовой команды)
  10. {
  11. регистр "LaunchApp" (Запуск приложения):
  12. разрыв;
  13. по умолчанию:
  14. разрыв;
  15. }
  16. }
  17. **Примечание:** Опции "Отладка. Строка записываемых переменных" будут применены позднее при отладке кода. Вы можете добавить дополнительные сообщения о выходных данных, если захотите увидеть результаты команд commandArgs и speechRecognitionResult. Далее в этой задаче мы будем исследовать отлаживаемые приложения, запущенные речевыми командами.

1. Для успешного запуска приложения и перехода на страницу, нам нужно будет воспроизвести некоторые модели поведения, существующие в настоящий момент в переопределении OnLaunched. Во время запуска приложения шаблон Пустого приложения создаёт корневой кадр и активирует окно, но не гарантирует выполнение тех же действий для регистров OnActivated. Добавьте код для обработки задач увеличения запуска.
   * 1. C#
   1. protected override void OnActivated(IActivatedEventArgs args)
   2. {
   3. Frame rootFrame = Window.Current.Content as Frame;
   4. if (rootFrame == null)
   5. {
   6. // Создайте фрейм для того, чтобы действовать, как навигационный контекст, и для перехода на первую страницу
   7. rootFrame = new Frame();
   8. rootFrame.NavigationFailed += OnNavigationFailed;
   9. // Поместите кадр в текущее Окно
   10. Window.Current.Content = rootFrame;
   11. }
   12. switch (args.Kind)
   13. {
   14. регистр ActivationKind.VoiceCommand (тип активации. Голосовая команда):
   15. HandleVoiceCommand (управление голосовой командой)(аргументы);
   16. разрыв;
   18. по умолчанию:
   19. разрыв;
   20. }
   21. Окно.Текущее.Активировать ();
   22. base.OnActivated(args);
   23. }
   24. **Примечание:** Чтобы избежать дублирования в реальном приложении, вы можете захотеть создать общий код запуска, который выполнится для запущенных и активированных приложений. Template10 демонстрирует более унифицированный способ обработки этих важных действий при запуске приложения. Для ознакомления с более подробной информацией о Template10 посетите страницу<https://github.com/Windows-XAML/Template10>
2. Перейдите по корневому кадру к методу **HandleVoiceCommand** в дополнение к **аргументам**. Вам понадобится контекст корневого кадра для перехода на страницу.
   * 1. C#
   1. switch (args.Kind)
   2. {
   3. регистр ActivationKind.VoiceCommand (тип активации. Голосовая команда):
   4. HandleVoiceCommand (аргументы, корневой кадр);
   5. разрыв;
   7. по умолчанию:
   8. разрыв;
   9. }
3. В методе **HandleVoiceCommand (управление голосовой командой)** добавьте параметр входящего **кадра** и используйте его для перехода на изображение **MainPage (Главная страница)**, когда команда **LaunchApp (Запуск приложения)** будет обнаружена.
   * 1. C#
   1. private void HandleVoiceCommand(IActivatedEventArgs args, Frame frame)

{

* 1. var commandArgs = args as VoiceCommandActivatedEventArgs;
  2. var speechRecognitionResult = commandArgs.Result;
  3. var command = speechRecognitionResult.Text;
  4. var voiceCommandName = speechRecognitionResult.RulePath[0];
  5. var textSpoken = speechRecognitionResult.Text;
  6. переключение (voiceCommandName) (имя голосовой команды)
  7. {
  8. регистр "LaunchApp" (Запуск приложения):
  9. кадр.Переход (тип (MainPage (Главная страница));
  10. разрыв;
  11. по умолчанию:
  12. разрыв;
  13. }
  14. }

1. Откройте MainPage.xaml и добавьте заголовок страницы. Текст поможет определить, что переход был совершен в момент запуска приложения.
   * 1. XAML
   1. <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
   2. <TextBlock Text="Speech Recognition and Voice Commands" FontWeight="Light" FontSize="20" Margin="12" />
   3. </Сетка>
2. Создайте и запустите своё приложение на Локальном компьютере.



Рисунок 5

Приложение, выполняющееся на Локальном компьютере.

1. Закройте своё приложение. Откройте проект **Свойства** из Обозревателя решений и перейдите во вкладку Отладка. Выберите действие активации.  **Не запускать, произвести отладку моего кода при активации**, и сохраните файл свойств. Вы можете использовать эту опцию, чтобы отладить приложение, которое вы не запускаете прямо из Visual Studio.
2. Для активации отладки без запуска приложения используйте кнопку **Начало отладки**.
3. Установите точку останова на **App.xaml.cs** после сообщений Отладка. Строка записываемых переменных.
4. Щёлкните по кнопке микрофона в своей панели задач для подготовки к запуску через голосовую команду.
5. Произнесите слова: "Лабораторный практикум, запуск". Кортана словами подтвердит, что открывает ваше приложение Распознавания речи.
   1. **Примечание:** Если ваши настройки региона, языка и речи будут заданы по другому поддерживаемому языку, для которого вы создали набор голосовых команд, вы можете использовать те команды вместо этих. Если вы измените свои настройки региона, языка и речи, вам может понадобиться выйти из системы и снова зайти, чтобы изменение полностью вступило в силу.
6. Когда вы достигнете своей точки останова, откройте окно **Вывода**, чтобы просмотреть свои сообщения об отладке. Вы увидите команду, которую распознала Кортана, а также фактический произнесенный текст. Они могут различаться в зависимости от того, насколько хорошо Кортана истолковала произнесенный вами текст. Используйте кнопку **Continue (Продолжить)** для возобновления запуска приложения при завершении.
7. Ваше приложение запустится и перейдет к изображению MainPage (Главной страницы).



Рисунок 6

Приложение запускается через голосовую команду.

1. Закройте своё приложение и вернитесь в Visual Studio. Удалите точку останова из **App.xaml.cs**, и снимите галочку с опции **Do not launch, but debug my code when it starts (Не запускать, произвести отладку моего кода при активации)**  в свойствах проекта. Сохраните файл.

Упражнение 2: Используйте голосовую команду, чтобы изменить вид приложения

* 1. В дополнение к запуску вашего приложения, голосовые команды могут взаимодействовать с содержанием в приложении. В этом упражнении вы будете использовать голосовую команду для изменения фонового цвета приложения при его запуске.

Задача 1 – Установите цвет фона

* 1. В этой задаче вы зададите исходный цвет фона для своего приложения и зададите атрибут **x:Name**, чтобы облегчить ориентирование на свойства сетки.

1. Укажите **AliceBlue** как цвет фона для вашей сетки в **MainPage.xaml** и присвойте сетке x: Имя **Container (Контейнер)** .
   * 1. XAML
   1. <Grid Background="AliceBlue" x:Name="Container" >
   2. <TextBlock Text="Speech Recognition and Voice Commands" FontWeight="Light" FontSize="20" Margin="12" />

</Сетка>

1. Создайте и запустите своё приложение на Локальном компьютере. Вы увидите заголовок страницы на светло-синем фоне.

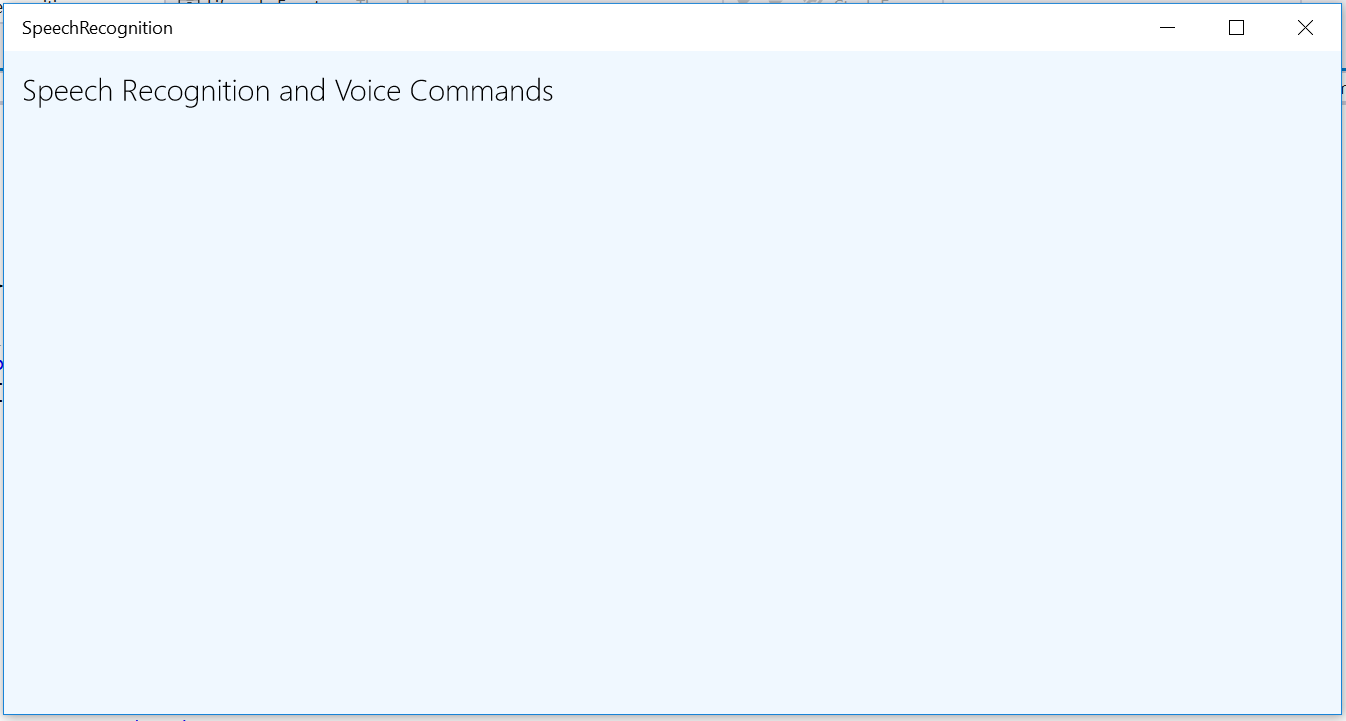


Рисунок 7

Работа приложения с цветным фоном.

1. Отключите отладку и вернитесь в Visual Studio.

Задача 2 – Создайте голосовую команду, чтобы вызвать изменение цвета

* 1. В этой задаче вы создадите голосовую команду для изменения фонового цвета сетки на красный.

1. Откройте свой файл определения VoiceCommands.xml и добавьте новую команду. Присвойте команде имя **TurnRed.**
   * 1. XML
   1. <Имя команды="TurnRed" (замменить на красный)>
   2. <Пример -> turn red (заменить на красный)</Пример>
   3. <Прослушать> заменить на красный </Прослушать>
   4. <Обратная связь> мой любимый цвет – красный</Обратная связь>
   5. <Переход />
   6. </Команда>
   7. **Примечание:** Если вы поддерживаете дополнительные языки, добавьте эквивалентную команду в её набор команд.
2. Добавьте случай для команды **TurnRed (Заменить на красный)** в переключение **voiceCommandName (имя голосовой команды)** на **App.xaml.cs**. В этот раз при переходе передайте параметр на MainPage (Главную страницу).
   * 1. C#
   1. переключение (voiceCommandName) (имя голосовой команды)
   2. {
   3. регистр "LaunchApp" (Запуск приложения):
   4. кадр.Переход (тип (MainPage (Главная страница));
   5. разрыв;
   6. случай "TurnRed" (Заменить на красный):
   7. кадр.Переход (тип (MainPage (Главная страница), "Красный");
   8. разрыв;
   9. по умолчанию:
   10. разрыв;
   11. }
3. Откройте отделённый код MainPage (Главной страницы) и создайте переопределение **OnNavigatedTo** для обработки входящего параметра. Задайте фон сетки в цвете, который соответствует входящему параметру.
   * 1. C#
   1. protected override void OnNavigatedTo(NavigationEventArgs e)
   2. {
   3. if (e.Parameter.ToString() == "Red")
   4. Container.Background = new SolidColorBrush(Windows.UI.Colors.DarkRed);
   5. base.OnNavigatedTo(e);
   6. }
4. Щёлкнитесь правой кнопкой мыши на своём проекте в Обозревателе решений и выберите **Развёртывание**.
5. Создайте и запустите своё приложение на Локальном компьютере для регистрации новой голосовой команды. Вы должны всё еще видеть синий фон. Закройте своё приложение.
6. Щёлкните по кнопке микрофона Кортаны и проговорите команду: **"Лабораторные практикумы, заменить на красный".** Кортана ответит, что красный – её любимый цвет. При следующем запуске приложения вы увидите, что цвет фона заменен на красный.

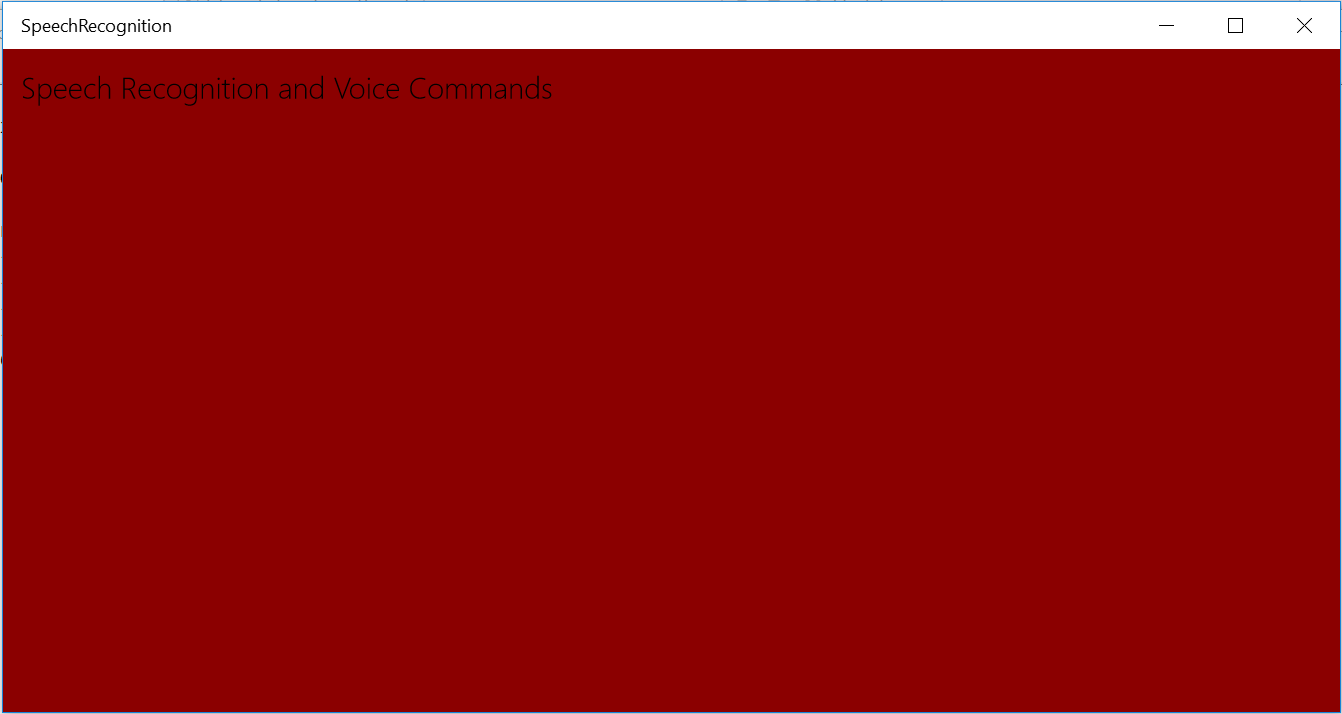


Рисунок 8

Приложение запускается через голосовую команду с красным фоном.

* 1. **Примечание:** Если вам нужно отладить код, который вы записали в этой задаче, обратитесь к инструкции для метода **Do not launch, but debug my code when it starts (Не запускать, произвести отладку моего кода при активации)** с Упражнения 1: Задача 4.

1. Отключите отладку и вернитесь в Visual Studio.

Упражнение 3: Ответьте на голосовую команду при помощи фоновой задачи

* 1. Голосовые команды могут активировать работу фоновых задач без запуска вашего приложения. Такая схема может быть полезной, когда вы хотите разрешить своим пользователям выполнять простые задачи, связанные с вашим приложением, непосредственно через Кортану без необходимости запуска приложения. В этом упражнении вы создадите Компонент Времени выполнения Windows для ответа на вопрос пользователя через фоновую задачу.

Задача 1 – Создайте компонент времени выполнения Windows

* 1. Мы начнём с создания Компонента Времени выполнения Windows для форматирования фоновой задачи, которую вы создадите в более поздней задаче.

1. Щёлкните правой кнопкой мыши на названии решения в Обозревателе решений. Выберите **Добавить -> Новый Проект** и выберите тип проекта **Компонент** Времени выполнения Windows (Windows Универсальная).
2. Назовите проект **VoiceCommandService (Служба голосовой команды)**.

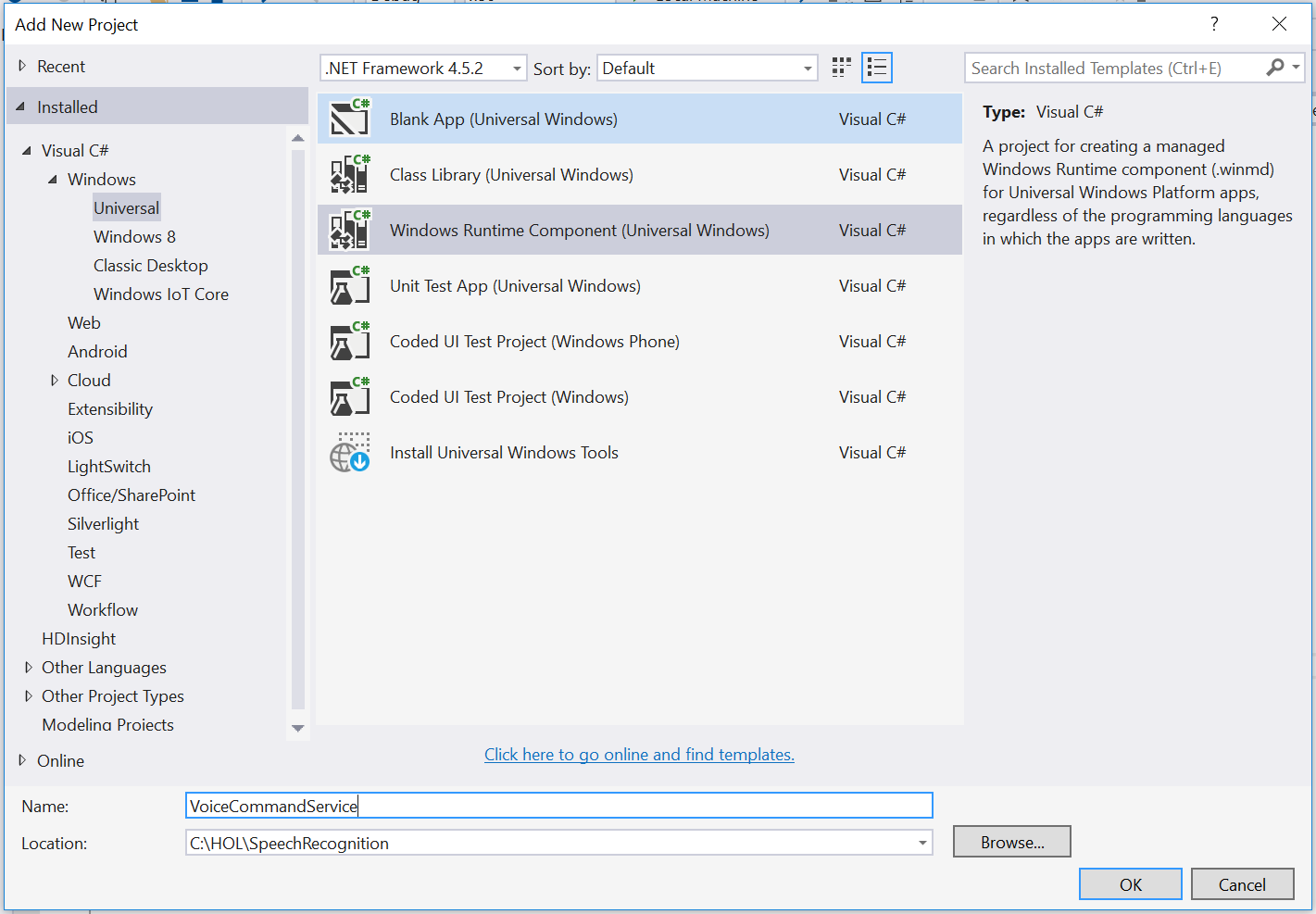


Рисунок 9

Добавьте Компонент Времени выполнения Windows.

1. Щёлкните правой кнопкой мыши по Class1.cs в Обозревателе решений и воспользуйтесь опцией Rename**(Переименование)**, чтобы переименовать его в **HolVoiceCommandService**. Если появится необходимость сменить имя проекта по всем ссылкам на Class1, выберите **"Да"**.

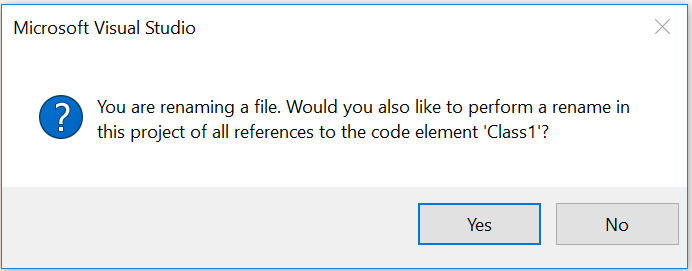


Рисунок 10

Переименуйте Class1.cs в HolVoiceCommandService.cs.

1. Вернитесь к проекту Распознавание речи. Щёлкните правой кнопкой мыши на папке Ссылки и добавьте VoiceCommandService (Служба голосовой команды) в виде ссылки.

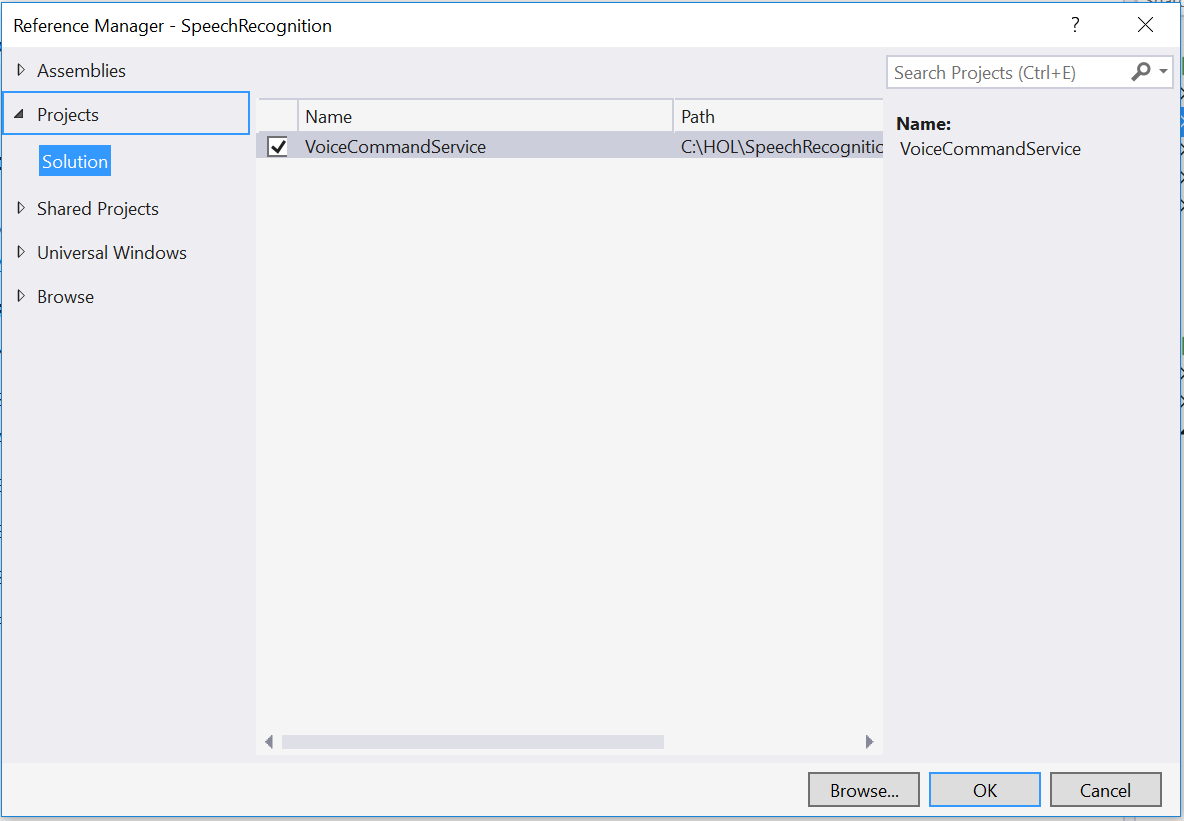


Рисунок 11

Ссылка на VoiceCommandService (Служба голосовой команды) из из проекта SpeechRecognition (Распознавание речи).

Задача 2 – Добавьте голосовую команду для ссылки на VoiceCommandService

* 1. Голосовые команды, которые запускают фоновые задачи, отличаются от голосовых команд, которые вы создавали в этом практическом курсе раньше. Вместо того, чтобы она использовать элемент Navigate (Переход), команда определит пользовательский элемент VoiceCommandService , имеющий целевой атрибут с указанием на класс HolVoiceCommandService. В этой задаче вы создадите голосовую команду, а также предоставите ей дополнительные опции словесного выражения. Вы можете добавить опции словесного выражения в любые голосовые команды, независимо от того, используются ли они для запуска приложения на переднем плане или фоновой задачи.

1. В своём проекте SpeechRecognition добавьте к**VoiceCommands.xml** команду **SayHello**. Вы можете определять в команде многочисленные <**ListenFor>** элементы. В этом случае определите для Кортаны два вопроса, оба из которых будут служить для запуска команды.
   1. **Примечание:** Если вы поддерживаете дополнительные языки, добавьте эквивалентную команду в её набор команд.
      1. XML
   2. <Имя команды="SayHello">
   3. <Пример>, SayHello<,/Пример>
   4. <ListenFor RequireAppName="BeforeOrAfterPhrase">How's it going</ListenFor>
   5. <ListenFor RequireAppName="BeforeOrAfterPhrase">Say hello</ListenFor>
   6. <Обратная связь -> Постойте, позвольте спросить</Обратная связь>
   7. <VoiceCommandService Target="HolVoiceCommandService" />
   8. </Команда>
   9. **Примечание:** Атрибут **RequireAppName="BeforeOrAfterPhrase"** придает гибкость и естественность словесному выражению ваших голосовых команд. Обе фразы "Лабораторный практикум, как дела?" И "Как дела, лабораторный практикум?" допустимы, если этот атрибут установлен в BeforeOrAfterPhrase. Для получения более подробной информации об опциях словесного выражения зайдите в ListenFor документацию по ссылке <https://msdn.microsoft.com/en-us/library/windows/apps/dn706593.aspx>

Задача 3 – Зарегистрируйте сервис в манифесте приложения

* 1. Для фонового запуска HolVoiceCommandService должен быть зарегистрирован в манифесте приложений SpeechRecognition.

1. Откройте Package.appxmanifest в редакторе манифестов и перейдите на вкладку Declarations (Манифесты).
2. Используя выпадающее меню **Available Declarations (Доступные манифесты)**, выберите **Службу приложений** и кликните **Добавить**, чтобы добавить его в список Поддерживаемых манифестов.

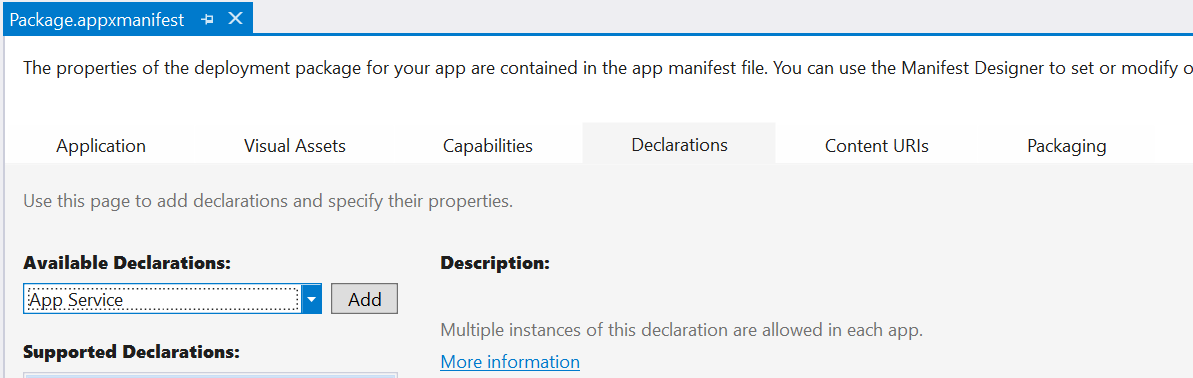


Рисунок 12

Добавьте Службу приложений в список Поддерживаемых манифестов.

1. В Свойствах манифеста Сервисов приложений, установите свойство **Имени** в **HolVoiceCommandService**.
2. Установите свойство точки **входа в VoiceCommandService.HolVoiceCommandService**.

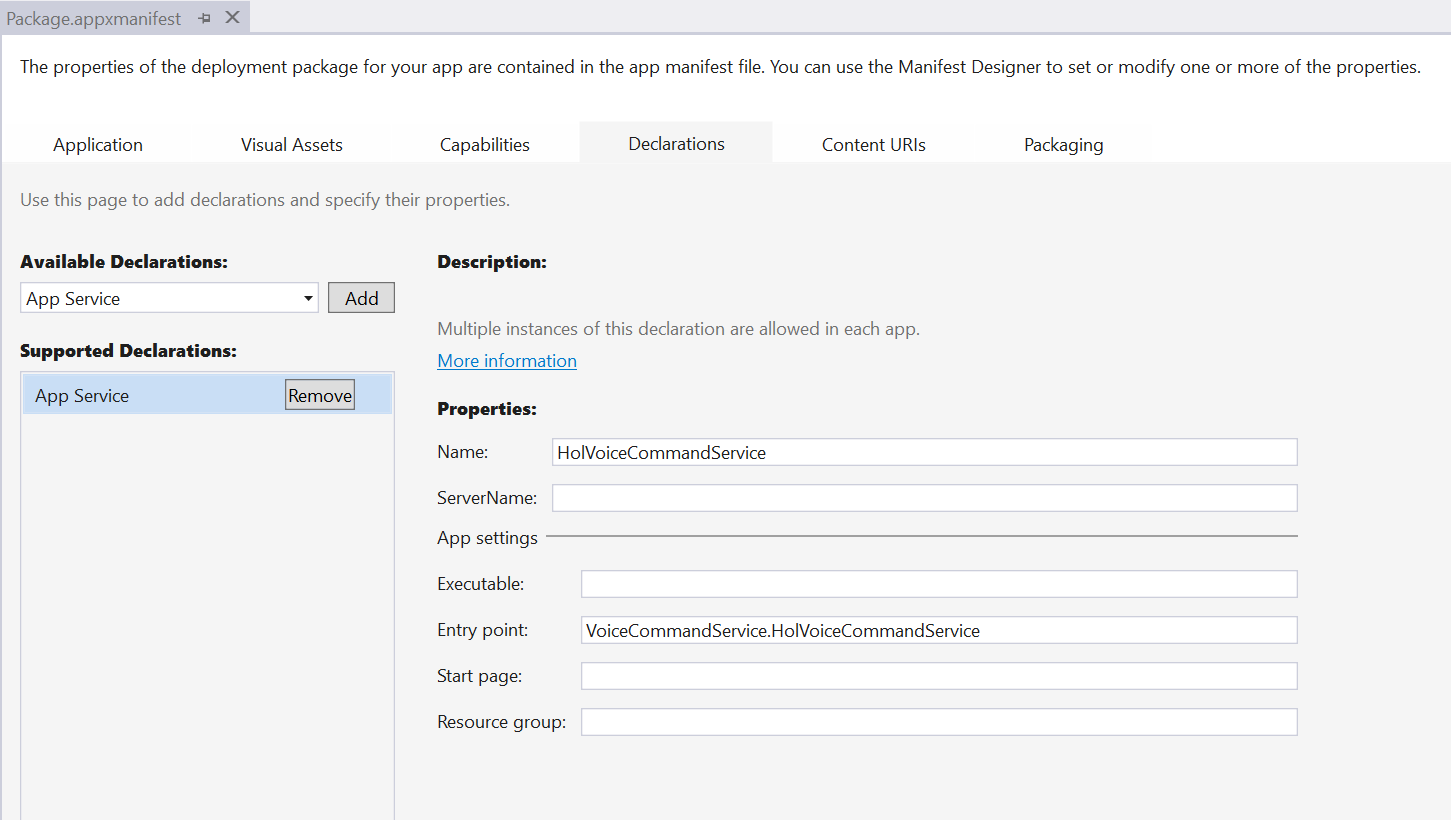


Рисунок 13

Зарегистрируйте Службу приложений в манифесте приложений.

* 1. **Примечание:** При регистрации службы приложений, свойство Имени должно соответствовать имени класса в рамках вашего компонента, а не имени самого компонента. Мы дали классу HolVoiceCommandService другое имя из компонента WinRT, чтобы их было легче различать.
  2. Это различие также важно при установке Целевого атрибута в определении голосовых команд, которое вы делали в Задаче 2.

1. Закройте манифест, сохранив изменения.

Задача 4 – Обработайте входящую команду в сервисе

* 1. В этой задаче вы будете применять точку входа для всех лишенных руководства голосовых команд, запущенных через Кортану. Фоновая задача, которую вы создадите, должна отвечать Кортане в течение 0.5 секунды и каждые 5 секунд должна сообщать о прогрессе.

1. Откройте **HolVoiceCommandService.cs и** добавьте **пространство имени** Windows.ApplicationModel.Background.
   * 1. C#
   1. используя Windows.ApplicationModel.Background;
2. Модифицируйте манифест класса так, чтобы он реализовывал интерфейс **IBackgroundTask**. Добавьте затребованный интерфейсом метод **Run**
   * 1. C#
   1. public sealed class HolVoiceCommandService : IBackgroundTask
   2. {
   3. public void Run(IBackgroundTaskInstance taskInstance)
   4. {
   5. throw new NotImplementedException();
   6. }
   7. }
3. Объявите элемент класса, именуемый **serviceDeferral**, типа **BackgroundTaskDeferral**. В своём методе **Run** получите отсрочку для своего экземпляра задач и сохраните его в serviceDeferral. Добавьте методы **OnVoiceCommandCompleted()** и **OnTaskCanceled()** для обработки завершения отсрочки. На более позднем этапе вы оформите подписку на события **VoiceCommandCompleted**.
   * 1. C#
   1. public sealed class HolVoiceCommandService : IBackgroundTask
   2. {
   3. BackgroundTaskDeferral serviceDeferral;
   4. public void Run(IBackgroundTaskInstance taskInstance)
   5. {
   6. serviceDeferral = taskInstance.GetDeferral();
   8. taskInstance.Canceled += OnTaskCanceled;
   9. }
   10. }
5. private void OnVoiceCommandCompleted(VoiceCommandServiceConnection sender, VoiceCommandCompletedEventArgs args)
6. {
7. if (this.serviceDeferral != null)
8. {
9. this.serviceDeferral.Complete();
10. }
11. }
12. private void OnTaskCanceled(IBackgroundTaskInstance sender, BackgroundTaskCancellationReason reason)
13. {
14. System.Diagnostics.Debug.WriteLine("Задача отменена, очистить");
15. if (this.serviceDeferral != null)
16. {
17. //Завершить отсрочку службы
18. this.serviceDeferral.Complete();
19. }
20. }
21. Добавить **пространства имен Windows.ApplicationModel.AppService** и **Windows.ApplicationModel.VoiceCommands.**
    * 1. C#
    1. используя Windows.ApplicationModel.AppService
    2. используя Windows.ApplicationModel.VoiceCommands;
22. Добавьте подключение службы голосовых команд.
    * 1. C#
    1. public sealed class HolVoiceCommandService : IBackgroundTask
    2. {
    3. VoiceCommandServiceConnection voiceServiceConnection;
    4. BackgroundTaskDeferral serviceDeferral;
    5. **Примечание:** Подключение службы поддерживается в течение всего периода работы Кортаны.
23. Проверьте детали триггера, чтобы посмотреть, соответствует ли имя имени регистрации Службы приложений из манифеста приложений. Если так, выполните блок попытка-перехват.
    * 1. C#
    1. taskInstance.Canceled += OnTaskCanceled;

var triggerDetails = taskInstance.TriggerDetails as AppServiceTriggerDetails;

* 1. if (triggerDetails != null && triggerDetails.Name == "HolVoiceCommandService")
  2. {
  3. try
  4. {
  5. voiceServiceConnection =
  6. VoiceCommandServiceConnection.FromAppServiceTriggerDetails(
  7. triggerDetails);
  8. voiceServiceConnection.VoiceCommandCompleted += OnVoiceCommandCompleted;
  9. }
  10. catch (Exception ex)
  11. {
  12. System.Diagnostics.Debug.WriteLine ("Обработка голосовой команды потерпела неудачу" + примеры ToString());
  13. }

}

* 1. **Примечание:** Подписка на событие VoiceCommandCompleted находится в этом кодовом пакете, так как оно должно произойти после установки voiceServiceConnection.
  2. Если вам нужно отладить этот код, используйте метод **Do not launch, but debug my code when it starts (Не запускать, произвести отладку моего кода при активации)**, описанный выше, и установите в вашем методе run точку останова.

1. Дождитесь входящей голосовой команды и создайте переключение для обработки регистра **SayHello**. Добавьте **асинхронное** ключевое слово в метод Run.
   * 1. C#
   1. try
   2. {
   3. voiceServiceConnection =
   4. VoiceCommandServiceConnection.FromAppServiceTriggerDetails(
   5. triggerDetails);
   6. voiceServiceConnection.VoiceCommandCompleted += OnVoiceCommandCompleted;
   7. VoiceCommand voiceCommand = ожидание voiceServiceConnection.GetVoiceCommandAsync();
   8. переключение (voiceCommand.CommandName)
   9. {
   10. "SayHello" регистр:
   11. разрыв;
   12. по умолчанию:
   13. разрыв;
   14. }
   15. }
2. Если входящая голосовая команда будет **SayHello**, создайте пользовательское сообщение и задайте дисплейное и речевое сообщения для передачи Кортаны обратно пользователю.
   * 1. C#
   1. переключение (voiceCommand.CommandName)
   2. {
   3. "SayHello" регистр:
   4. var userMessage = new VoiceCommandUserMessage();
   5. userMessage.DisplayMessage = "Hello!";
   6. userMessage.SpokenMessage = "Your app says hi. It is having a great time.";
   7. var response = VoiceCommandResponse.CreateResponse(пользовательское сообщение);
   8. ожидание voiceServiceConnection.ReportSuccessAsync (ответ);
   9. разрыв;
   10. по умолчанию:
   11. разрыв;
   12. }
3. Сохраните HolVoiceCommandService.

Задача 5 – Начните работу со своим приложением через фоновую задачу

Теперь, когда вы настроили голосовую команду, фоновую задачу и зарегистрировали службу приложений, самое время попробовать, как это работает.

1. Запустите проект SpeechRecognition на Локальном компьютере, чтобы зарегистрировать самую последнюю версию файла определения голосовых команд.
2. Закройте приложение.
3. Используйте подсказку Кортаны для того, чтобы проговорить один из вариантов голосовой команды SayHello:

* "Лабораторный практикум, как дела?"
* И "Как дела, лабораторный практикум?"
* Лабораторный практикум, скажите привет.”
* "Скажите привет – Лабораторный практикум.

1. Кортана покажет изображение вашего приложения и дисплейное сообщение. Одновременно она проговорит речевое сообщение, которое вы выбрали в фоновой задаче.

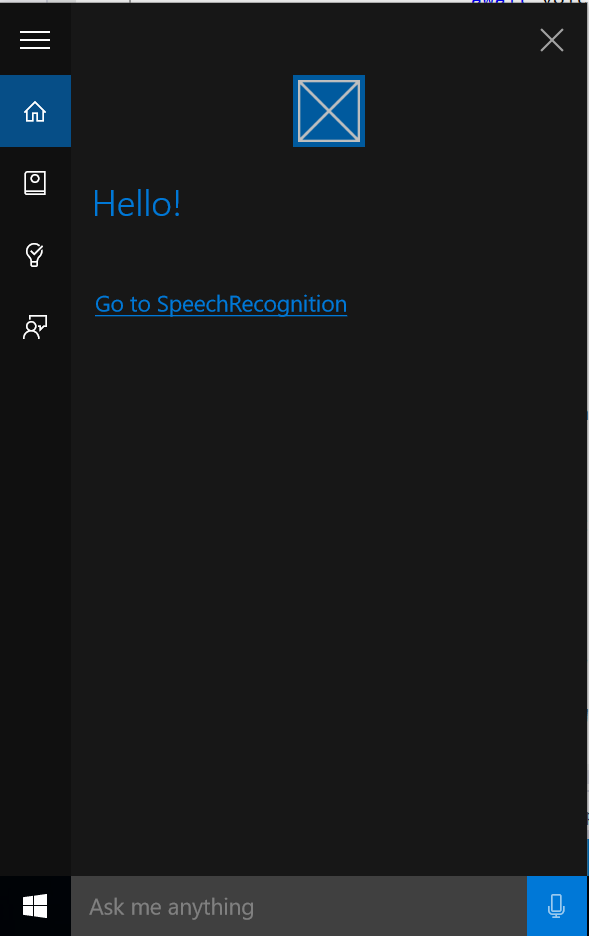


Рисунок 14

Кортана возвращает сообщение из фоновой задачи приложения.

* 1. **Примечание:** Чтобы отладить свою фоновую задачу, используйте метод аналогичный тому, который вы использовали для отладки своих голосовых команд. Щёлкните правой кнопкой мыши на проекте SpeechRecognition в Обозревателе решений и откройте редактор **Свойств**. На подокне **Отладка** отметьте опцией **Не запускать, произвести отладку моего кода при активации**. Сохраните файл свойств и используйте кнопку Пуска отладки для активации отладки. Установите точку останова в HolVoiceCommandService и запустите своё приложение, используя команду SayHello. Если ваши точки входа будут настроены правильно, вы достигнете точки останова.

1. Отключите отладку и вернитесь в Visual Studio.

Summary

* 1. В голосовых командах обрабатываются важные взаимодействия в Windows 10. В этом курсе вы создали файл определения голосовых команд и исследовали VCD схему. Вы добавили команды запуска приложения, взаимодействия с его видом и получения ответа из фоновой задачи. Вы также научились регистрировать и различать голосовые команды.