

# Хатмарин и кросс-платформенная разработка

**Xamarin.Forms** представляет платформу, которая нацелена на создание кроссплатформенных приложений под **Android, iOS** и **Windows 10**.

**Xamarin** позволяет создавать одну единственную логику приложения с применением **C#** и **.NET** сразу для всех трех платформ - **Android, iOS, UWP**.

Преимущества использования **Xamarin.Forms**:

- В процессе разработки создается единый код для всех платформ
- **Xamarin** предоставляет прямой доступ к нативным **API** каждой платформы
- При создании приложений мы можем использовать платформу **.NET** и язык программирования **C#** (а также **F#**), который является достаточно производительным, и в тоже время ясным и простым для освоения и применения
- **Xamarin Forms** поддерживает несколько платформ. Основные платформы: **Android, iOS, UWP, Tizen**. Дополнительные платформы, которые развиваются сообществом: **MacOS, WPF, GTK#**.

5 января 2021 года вышла последняя версия данной платформы - **Xamarin Forms 5.0**

# Как работает Xamarin




# Установка Xamarin


При установке Visual Studio 2019 в программе для установщика обязательно надо выбрать пункт "Разработка мобильных приложений на .NET":

Изменение — Visual Studio Community 2019 — 16.8.3

Рабочие нагрузки    Отдельные компоненты    Языковые пакеты    Расположения установки


 **Разработка мобильных приложений на .NET**

Создавайте кросс-платформенные приложения для iOS, Android или Windows на языке Xamarin.

 **Разработка мобильных приложений на языке C++**

Создавайте кросс-платформенные приложения для iOS, Android или Windows на языке C++.

Игры (2)


 **Разработка игр с помощью Unity**

В Unity, мощной кросс-платформенной среде разработки, можно создавать игры в формате 2D и 3D.

Расположение  
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

Продолжая, вы принимаете условия [лицензии](#) для выбранного выпуска Visual Studio. Мы также предлагаем возможность скачать Visual Studio другое программное обеспечение. Оно лицензируется отдельно, как указано в [уведомлениях третьих сторон](#) или сопутствующей лицензии. Продолжая, вы также принимаете условия этих лицензий.

Общее необходимое пространство 4,21 ГБ

Установка при скачивании  [Изменить](#)

# После инсталляции можно в меню **Help** ->**About Microsoft Visual Studio** увидеть отметку о

About Microsoft Visual Studio

## Visual Studio

[License status](#)  
[License terms](#)

Microsoft Visual Studio Community 2019  
Version 16.1.0  
© 2019 Microsoft Corporation.  
All rights reserved.


Microsoft .NET Framework  
Version 4.7.03056  
© 2019 Microsoft Corporation.  
All rights reserved.

Installed products:

- Visual Studio Tools for Unity – 4.1.1.0
- VisualStudio.Mac – 1.0
- Xamarin – 16.1.0.542 (d16-1@68b985244)**
- Xamarin Designer – 16.1.0.418 (remotes/origin/d16-1@5b958bb10)
- Xamarin Templates – 16.2.112 (4db4af4)
- Xamarin.Android SDK – 9.3.0.22 (HEAD/8e7764fdf)
- Xamarin.iOS and Xamarin.Mac SDK – 12.10.0.153 (750a879)

[Copy Info](#)  
[System Info](#)  
[DxDiag](#)

Product details:

 Visual Studio extension to enable development for Xamarin.iOS and Xamarin.Android.

Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

[OK](#)

Если целевой операционной системой является Mac OS X, то в этом случае нам надо установить Visual Studio for Mac.

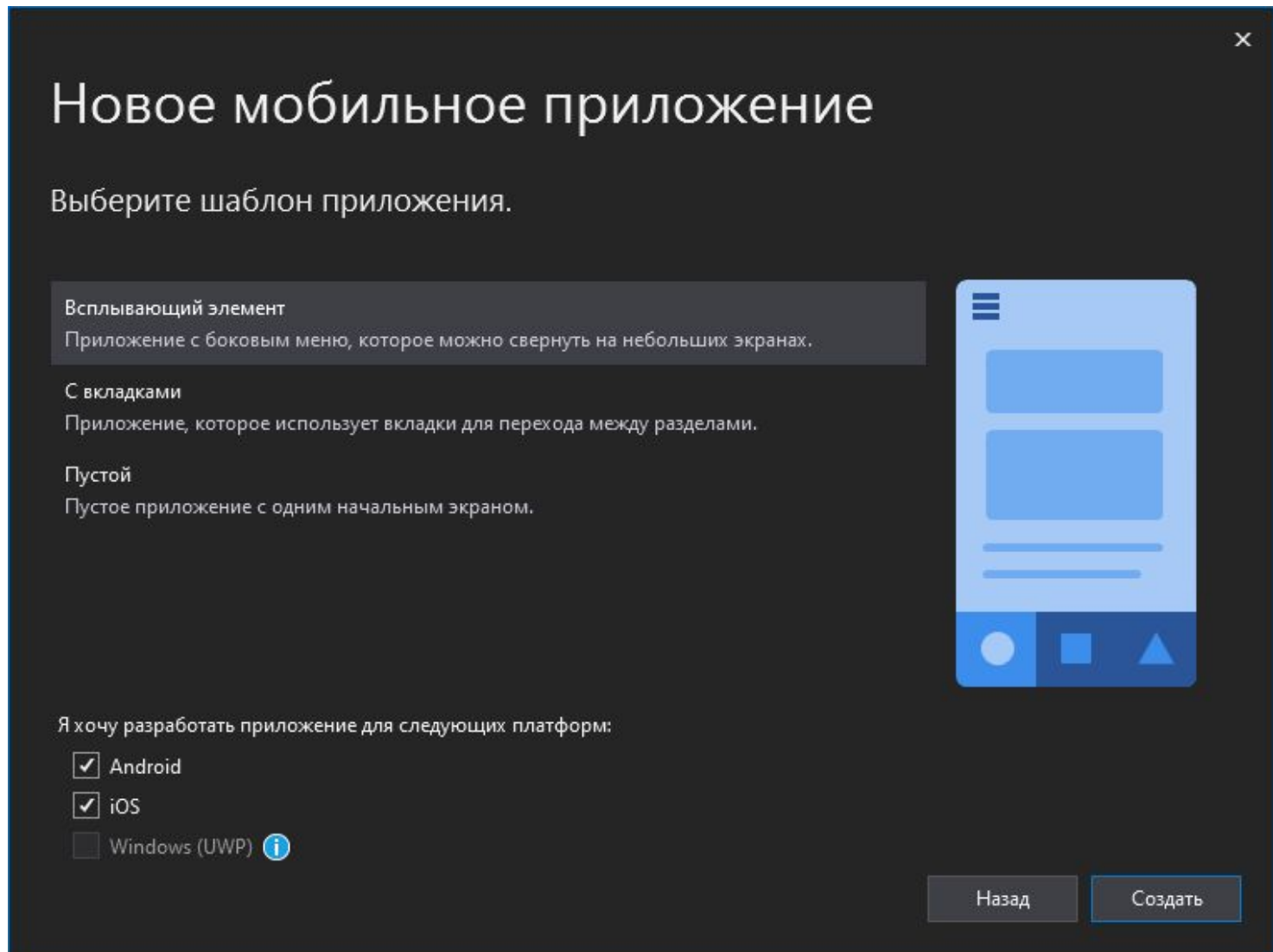
Кроме того, для разработки на Mac OS X требуется установить XCode, который доступен в AppStore.

# Создание и настройка проекта в Visual Studio

Для создания кроссплатформенных приложений для Xamarin Forms в Visual Studio 2019 предназначен шаблон проекта, который называется **Mobile App (Xamarin.Forms)**.



После создания проекта будет предложено выбрать шаблон проекта и установить ряд настроек:



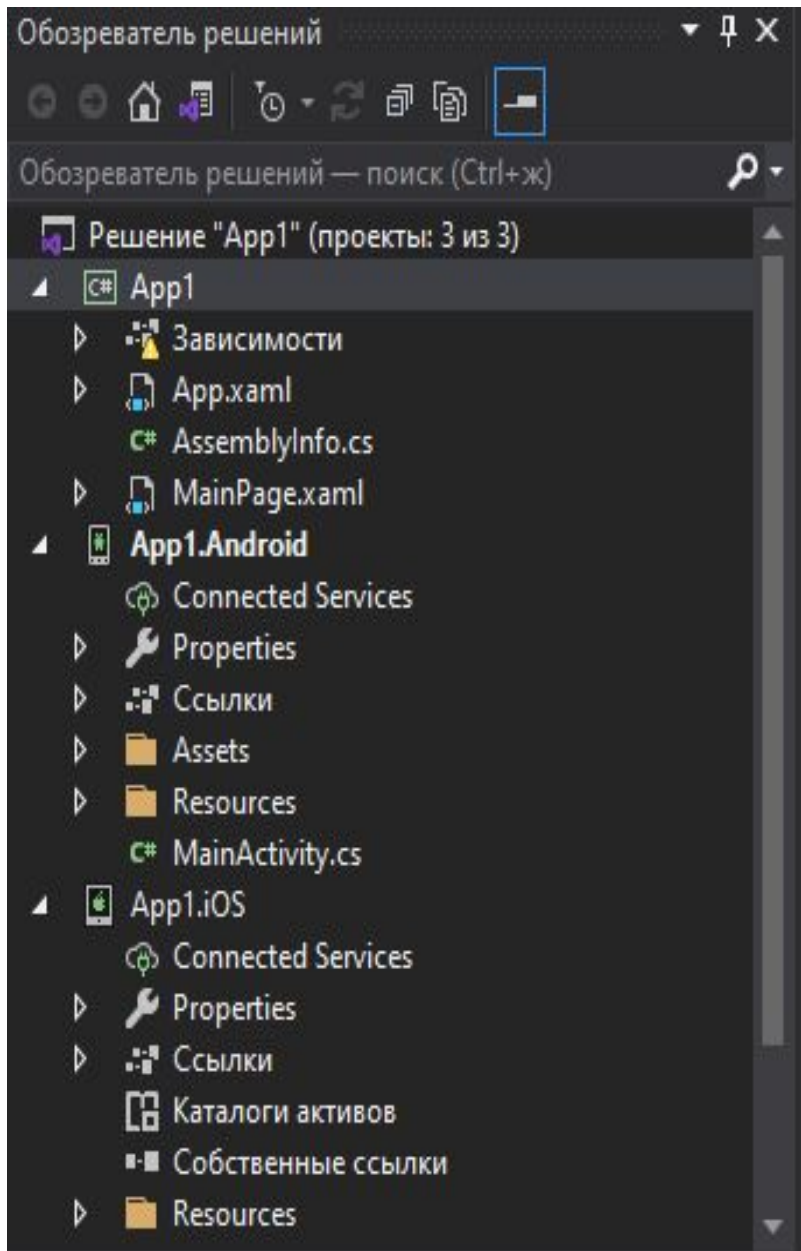


Здесь доступно три шаблона:

- **Floyout**: проект с боковым меню
- **Tabbed**: проект приложения, которое использует вкладки для навигации между страницами
- **Blank**: пустой шаблон, создающий проект с минимальной функциональностью

Кроме того, при создании проекта в опции **Platform** мы можем отметить, под какие ОС будет создаваться проект. В данном случае мы можем выбрать все платформы: **Android, iOS, Windows (UWP)**.

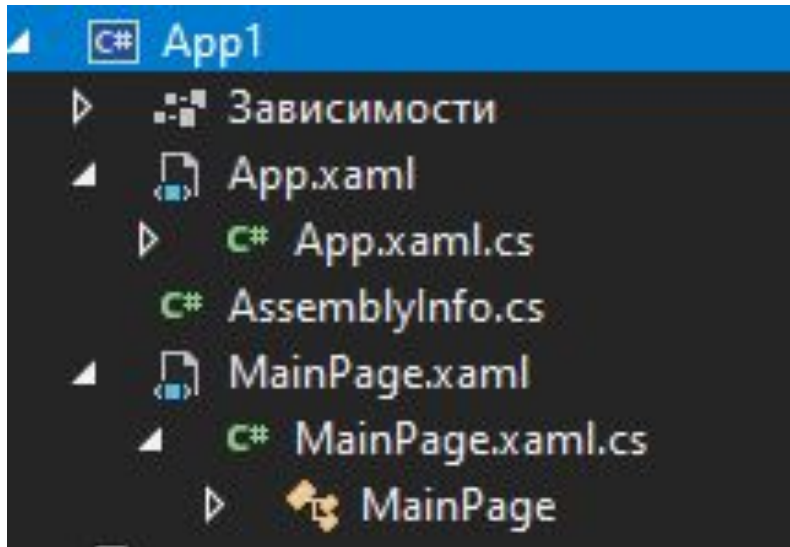
**! Windows (UWP)** доступна только под **Windows**, если для **Visual Studio** установлен инструментарий для разработки под **UWP**.



И Visual Studio сгенерирует новое решение. Если были выбраны выше все три платформы, то созданное решение будет содержать четыре проекта.

- **App1**: главный проект библиотеки, которая и будет содержать всю основную логику приложения
- **App1.Android**: проект для Android
- **App1.iOS**: проект для iOS
- **App1.UWP**: проект для Windows 10

Главным будет самый верхний проект (в данном случае **App1**). Он будет содержать весь код и определение интерфейса, которые затем будут использовать все остальные проекты.



**App.xaml:** файл, который определяет ресурсы, общие для всего приложения

**App.xaml.cs:** файл с кодом C#, с которого начинается выполнение приложения

**MainPage.xaml:** файл с визуальным интерфейсом для единственной страницы MainPage в виде xaml

**MainPage.xaml.cs:** файл, который содержит логику MainPage на языке C#

**AssemblyInfo.cs:** файл с кодом на языке C#, который используется для установки настроек приложения

Файл **App.xaml.cs** содержит следующий класс:

```
namespace App1
{
    public partial class App : Application
    {
        public App()
        {
            InitializeComponent();
            MainPage = new MainPage();
        }
        protected override void OnStart()
        {
        }
        protected override void OnSleep()
        {
        }
        protected override void OnResume()
        {
        }
    }
}
```

Файл **MainPage.xaml** представляет визуальный интерфейс страницы в виде кода XAML. В качестве корневого элемента в этом файле определен элемент **ContentPage**.

Все его содержимое состоит из элемента **StackLayout**, который позволяет расположить вложенные элементы в виде вертикального стека. Внутри **StackLayout** определен ряд элементов, которые собственно определяют видимые на экране элементы. В данном случае в основном это элементы **Label**, которые служат для вывода простого или отформатированного текста на экран. Текст задается с помощью атрибута **Text**.

Также в проекте есть и файл с кодом логики страницы - файл **MainPage.xaml.cs**:

```
namespace HelloApp
{
    public partial class MainPage : ContentPage
    {
        public MainPage()
        {
            InitializeComponent();
        }
    }
}
```

Через ЭТОТ ВЫЗОВ в конструкторе **InitializeComponent()** на странице формируется интерфейс, который определен в файле **MainPage.xaml**.

Главный проект со всеми рассмотренными выше файлами компилируется в библиотеку `dll`, а остальные три проекта содержат ссылки на него.

Так, если открыть узел `References` у каждого проекта, то можно увидеть там ссылку на библиотеку `HelloApp` - то есть ссылку на главный проект.

Кроме этой библиотеки все три проекта содержат еще ряд важных ссылок, в частности каждый проект содержит ссылку на свою специфическую библиотеку:

Xamarin.Forms.Platform.Android

Xamarin.Forms.Platform.iOS

Universal Windows

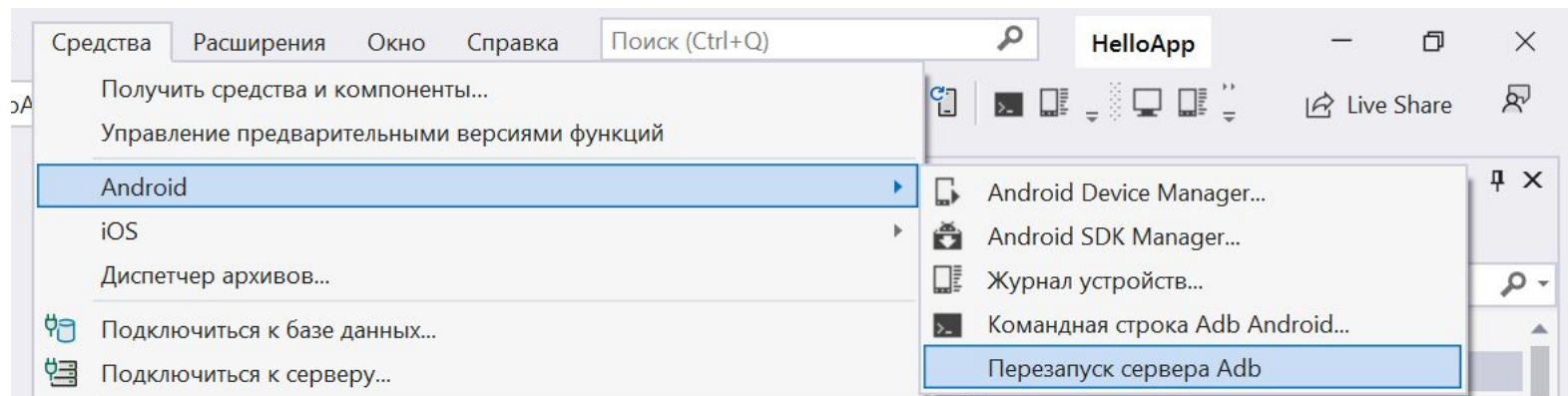
Данные библиотеки определяют для своей мобильной платформы статический метод `Xamarin.Forms.Init()`, который выполняет инициализацию системы `Xamarin.Forms`.



# Запуск на Android

1. Если рабочая машина, на которой ведется разработка, поддерживает виртуализацию, то можно для отладки приложений использовать эмуляторы: эмулятор Android от Microsoft или Android Player.
2. Если есть устройство с ОС Android, то можно его использовать для тестирования. Для этого надо подключить это устройство к компьютеру с помощью USB-кабеля. А на самом мобильном устройстве установить режим разработчика в параметрах.

Если мобильное устройство подключено и настроено должным образом, но Visual Studio его все равно не видит, то в этом случае можно перезагрузить сервер Adb. Для этого надо в Visual Studio перейти в меню к пункту **Tools -> Android -> Restart Adb Server**:



Перед самым запуском можно настроить некоторые параметры, которые связаны с **Android**.

Для этого необходимо перейти в **Visual Studio** в меню **Tools -> Options** и в появившемся окне выбрать пункт **Xamarin -> Android Settings**.

Здесь можно настроить расположение **Java SE SDK** и **Android SDK**, а также настройки эмулятора и прочее.

# Параметры

? X

Параметры поиска (CTRL+E)



- Производительность
- Расположения
- Сборка и запуск
  - Управление веб-пакетами
- Рабочие элементы
- Система управления версиями
- Текстовый редактор
- Отладка
- Средства производительности
- CMake
- F# Tools
- IntelliCode
- Live Share
- Xamarin
  - Apple Accounts
  - Конструктор пользовательского интерфейса
  - Параметры Android**
  - Параметры iOS
- Адаптер теста для Google Test
- Диагностика графики
- Диспетчер пакетов NuGet
- Инструменты базы данных
- Инструменты веб-тестов производи...

Расположение пакета Java Development Kit



C:\Program Files\Android\Jdk\microsoft\_dist\_openjdk\_1.8.0.25

Расположение пакета Android SDK



C:\Program Files (x86)\Android\android-sdk

Отладка на устройстве или в эмуляторе



Сохранять кэш данных приложения на устройстве между развертывани



Предоставить отладочные символы для общей среды выполнения и би



Предупреждать, если ускорение AVD не поддерживается

Дополнительные аргументы запуска эмулятора:



Автоматически установить пакеты SDK для Android



Включить средство миграции на AndroidX (экспериментальная функци

Детализация выходных данных Xamarin Diagnostics:

Обычный

Расположение архивов

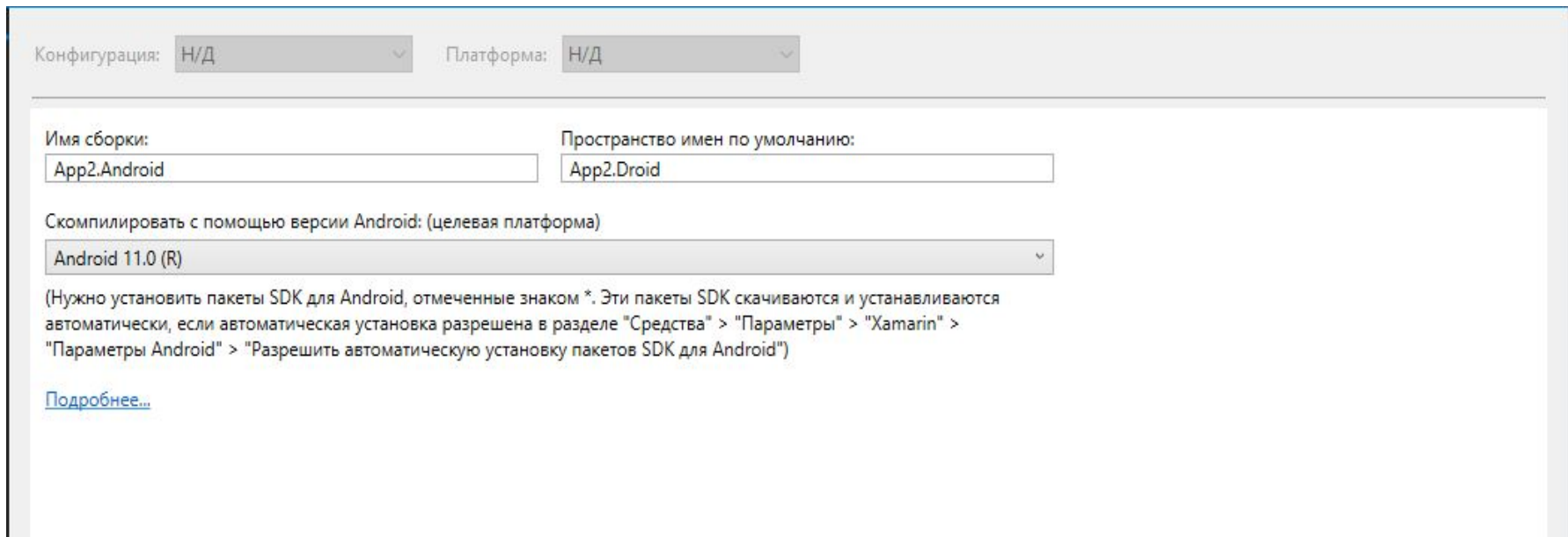


C:\Users\iRIP\AppData\Local\Xamarin\Mono for Android\Archives

OK

Отмена

Можно перейти в свойства проекта для **Android**, и в настройках установить версию среды **Android**, которая будет использоваться для компиляции приложения:



Конфигурация: Н/Д ▾ Платформа: Н/Д ▾

Имя сборки:  Пространство имен по умолчанию:

Скомпилировать с помощью версии Android: (целевая платформа)

(Нужно установить пакеты SDK для Android, отмеченные знаком \*. Эти пакеты SDK скачиваются и устанавливаются автоматически, если автоматическая установка разрешена в разделе "Средства" > "Параметры" > "Xamarin" > "Параметры Android" > "Разрешить автоматическую установку пакетов SDK для Android")

[Подробнее...](#)

Важную роль играют настройки манифеста, где можно установить различные разрешения, иконку и название приложения, целевую и минимальную версию **Android**, которые будут поддерживаться приложением, и прочие настройки:

Конфигурация: Н/Д

Платформа: Н/Д

Имя приложения:

App2.Android

Имя пакета:

com.companyname.app2

Значок приложения:

Тема приложения:

@style/MainTheme

Номер версии:

1

Имя версии:

1.0

Расположение установки:

Предпочтительно внутренняя

Минимальная версия Android:

Android 5.0 (уровень API 21 — Lollipop)

Целевая версия Android:

Android 11.0 (уровень API 30 — R)

Требуемые разрешения:

Версию среды стоит учитывать, если есть определенное мобильное устройство, которое нужно использовать для тестирования. Если его версия **Android** ниже минимальной, то **Visual Studio** может просто не увидеть это устройство, даже если оно подключено через **USB** к компьютеру.

Если же все настроено правильно, то в панели инструментов **Visual Studio** отобразит подключенное устройство:





Если необходимо установить какую-то новую версию Android API или какие-то компоненты из Android SDK, можно перейти к меню **Tools -> Android -> Android SDK Manager** и там отметить нужные компоненты для установки. После запуска проекта, в соответствии с имеющимся кодом он будет отображать метку с текстом в центре экрана:

