Contents

[Introduction 2](#_Toc101429056)

[Pre 2](#_Toc101429057)

[Create the Hello World project 2](#_Toc101429058)

[Build the project 3](#_Toc101429059)

[Create the application file structure 4](#_Toc101429060)

[Sign the application 4](#_Toc101429061)

[Install the application 5](#_Toc101429062)

# Introduction

This document describes how to create the FR22 Hello World application in C#. The source code for the sample can be found at [https://github.com/NordicID/fr22\_samples/1. HelloWorld](https://github.com/NordicID/fr22_samples/1.%20HelloWorld).

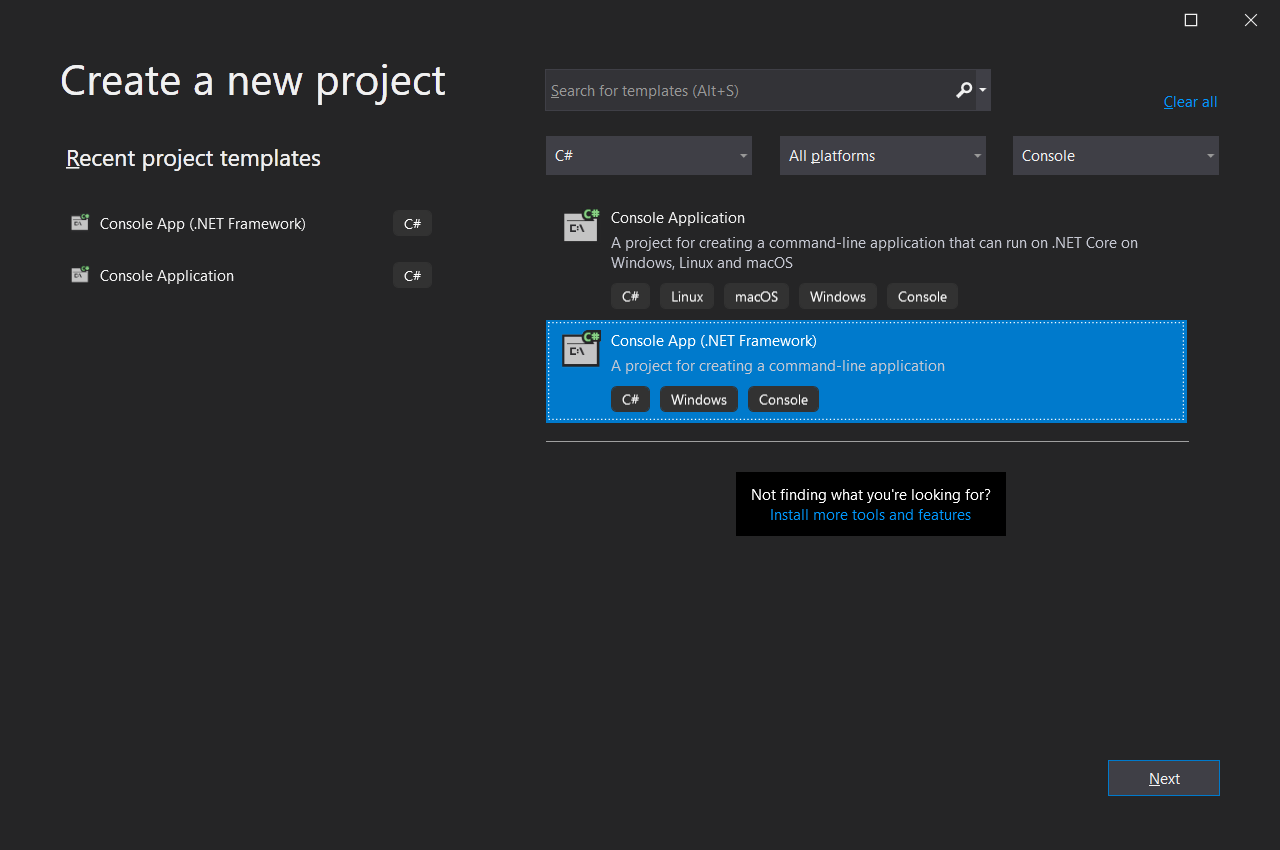
## Prerequisites

* FR22 device with OS version 0.8.0 or later
* fr\_appsigntool files from fr22\_appsigntool directory in <https://github.com/NordicID/fr22_samples/> (requires .NET Core 3.1)
* Visual Studio 2019/2022 with C# support installed
  + It’s possible to use VS Code or the dotnet tool, but this guide assumes Visual Studio is used.

# Create the Hello World project

In Visual Studio 2019, choose

* Create a new project
* Apply template filter: C#/All Platforms/Console
* Choose template: Console App (.NET Framework)



* Press Next
* Configure the new project:
  + Project name: HelloWorld
  + Framework: .NET Framework 4.8

A screenshot of a computer

Description automatically generated

* Press Create

**Note**: it’s also possible to use older .NET Framework versions; for available versions see the FR22’s built-in App Center in the WebUI by navigating to *Software->App Center* and search for *“mono”*. (Devices not connected to the internet will not be able to use the App Center; the mono .NET Framework plugins can also be installed manually by first downloading them from the Nordic ID support site).

# Build the project

Modify the project and compile it

* Update the Main function to write “Hello World” to the console
* Change to Release build (optional)
* Build solution

# Create the application file structure

A FR22 application is a zip file with a fixed directory structure documented separately. The simples possible format is the executable file and a manifest file describing the application.

To create the application zip file structure:

* Create the ZipContents directory, and under it create bin and meta directories
* From the C# build output, copy bin\Release\HelloWorld.exe and bin\Release\HelloWorld.pdb to the ZipContent\bin\ directory
  + The pdb file is optional, it’s only needed in order to get usable stack traces
* The manifest for a FR22 application needs to specify the application name and version. A mono application also needs to specify how to start the application and what mono plugin it depends on. To do this, create a ZipContent\meta\manifest.json file with the content:

{

"name": "HelloWorld",

"version": "1.0.0",

"startup": {

"exec\_start": "mono bin/HelloWorld.exe",

"depends": {

"plugins": [

"mono-api-4.8"

]

}

}

}

# Sign the application

Use the supplied fr\_appsigntool to create and sign the FR22 application zip file by opening a command prompt in the project directory:

* ..\fr22\_appsigntool\fr\_appsigntool.exe ZipContents

Text

Description automatically generated

* The file HelloWorld\_1.0.0-app.zip will be generated

# Install the application

To install the application on the FR22 device, login to the WebUI and navigate to *Software->Applications*.

* Under *Install new application*, press *Browse* and then select the HelloWorld\_1.0.0-app.zip file
* Press *Install*
* If mono has not been previously installed on the device, and the device is connected to the internet, the WebUI will now ask if it should automatically install the mono dependencies via the built-in App Center; allow this
  + If the App Center is not able to reach an App Center server, the dependencies need to be installed manually by navigating to Software->System Plugins and installing:
    - mono-4.5\_6.12.0.107.1-plugin.zip
    - mono-api-4.8\_6.12.0.107.1-plugin.zip

After the application (and the mono dependencies have been installed), navigate to Software/Applications and on the HelloWorld row click on the Start button.

Graphical user interface, application

Description automatically generated

The application should now start and print Hello World! in the application log. To view the application log click on the Log button at the end of the HelloWorld row.

Text

Description automatically generated