

Adding the MobileFirst Platform Foundation SDK to Windows 8.1 Universal or Windows 10 UWP Applications

Overview

The MobileFirst Platform Foundation SDK consists of a collection of dependencies, available through Nuget (<https://www.nuget.org/>), that you can add to your Visual Studio project. The dependencies correspond to core functions and other functions:

- **IBMMobileFirstPlatformFoundation** - Implements client/server connectivity, handles authentication and security aspects, resource requests, and other required core functions.

In this tutorial you will learn how to add the MobileFirst Native SDK using Nuget to either a new or existing Windows 8.1 Universal application or Windows 10 UWP (Universal Windows Platform) application. You will also learn how to configure the MobileFirst Server to recognize the application, as well as find information about the MobileFirst configuration files that are added to the project.

For instruction to manually add the SDK files to a project, visit the user documentation (http://www-01.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/wl_welcome.html).

Prerequisites:

- Microsoft Visual Studio 2013 or 2015 and MobileFirst Developer CLI installed on the developer workstation. Developing Windows 10 UWP solution requires at least Visual Studio 2015.
- MobileFirst Server to run locally, or a remotely running MobileFirst Server.
- Make sure you have read the Setting up your MobileFirst development environment ([../setting-up-your-development-environment/mobilefirst-development-environment](#)) and Setting up your Windows 8 Universal and Windows 10 UWP development environment ([../setting-up-your-development-environment/windows-8-10-development-environment](#)) tutorials.

Jump to:

- Adding the MobileFirst Native SDK
- Updating the MobileFirst Native SDK
- Generated MobileFirst Native SDK artifacts
- Tutorials to follow next

Adding the MobileFirst Native SDK

Follow the below instructions to manually add the MobileFirst Native SDK to either a new or existing Visual Studio project, and registering the application in the MobileFirst Server.

Before starting, make sure the MobileFirst Server is running.

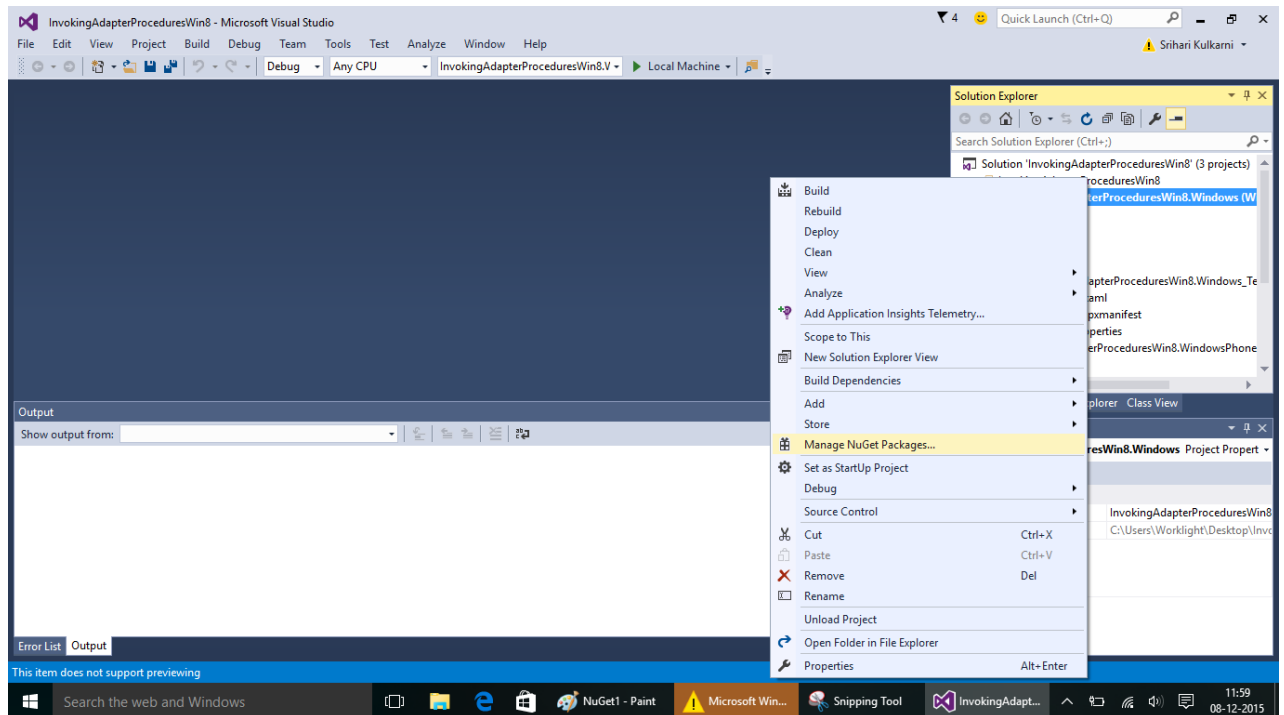
If using a locally installed server: From a **Command-line** window, navigate to the server's **scripts** folder and run the command: `./start.sh` in Mac and Linux or `start.cmd` in Windows.

Creating an application

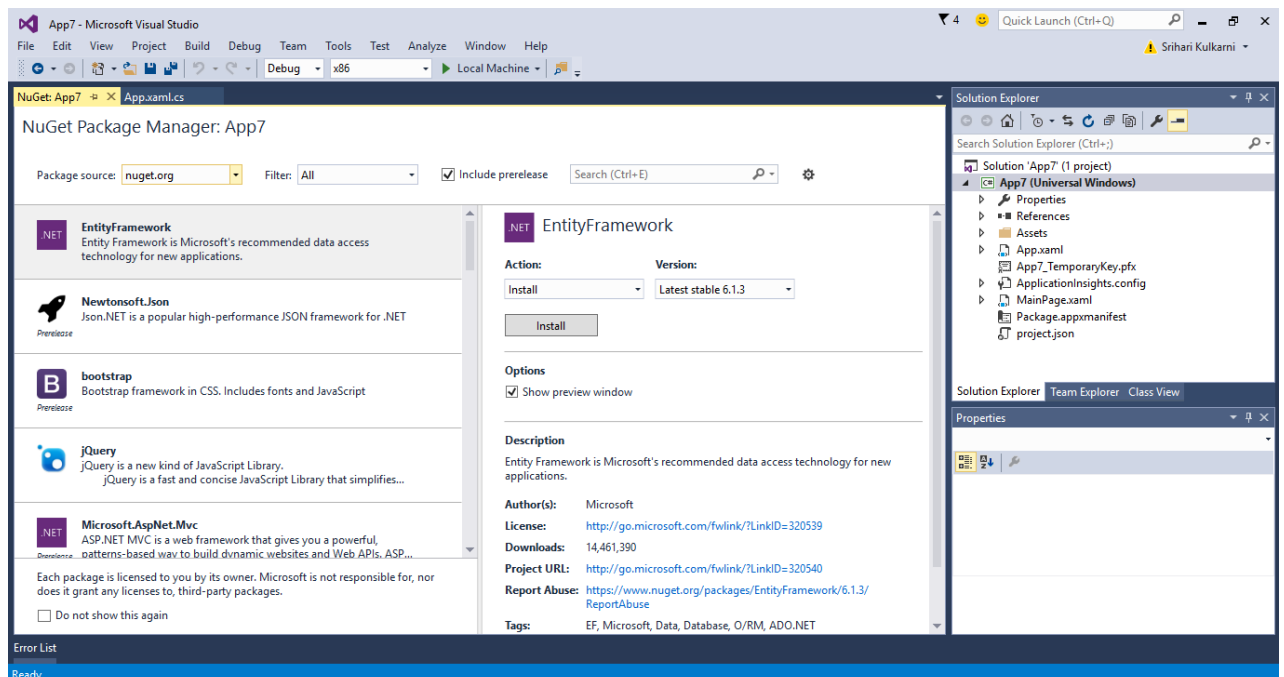
Create a Windows 8.1 Universal or Windows 10 UWP project using Visual Studio 2013/2015 or use an existing project.

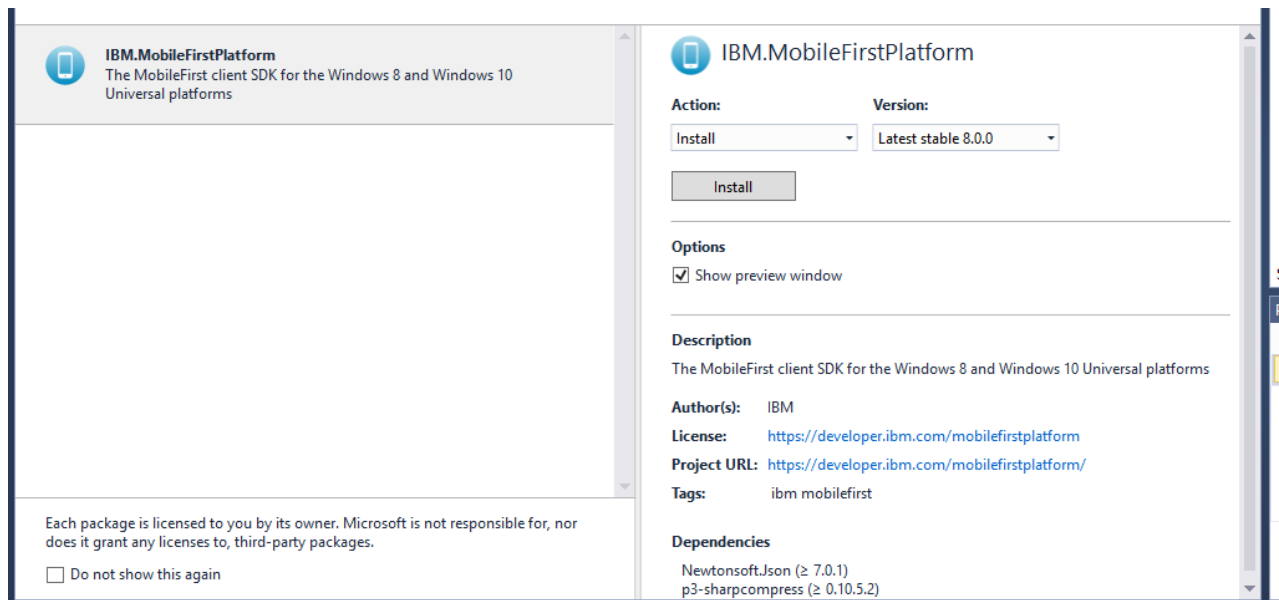
Adding the SDK

1. To import worklight studio packages, NuGet package manager is used. NuGet is the package manager for the Microsoft development platform including .NET. The NuGet client tools provide the ability to produce and consume packages. The NuGet Gallery is the central package repository used by all package authors and consumers.
2. Open the Windows 8.1 Universal or Windows 10 UWP project in Visual studio 2013/2015. Right-click the project solution and select **Manage Nuget packages**.



3. In the search option, search for "IBM MobileFirst Platform". Choose **IBM.MobileFirstPlatform.8.0.0.0**.





4. Click **Install**. This installs the IBM MobileFirst Platform Native SDK and its dependencies. This step also generates an empty `mfpclient.resw` file in `strings` folder in the Visual Studio project.
5. Ensure that, at a minimum, the following capabilities are enabled in `Package.appxmanifest`:

- Internet (Client)

Registering the application

1. Open the **Command-line** and navigate to the root of the Visual Studio project.
2. Run the command:

```
mfpdev app register
```

The `mfpdev app register` CLI command first connects to the MobileFirst Server to register the application, followed by updating the `mfpclient.resw` file in the **strings** folder in the Visual Studio project, and adding to it the metadata that identifies the MobileFirst Server.

Tip: The application registration can also be performed from the MobileFirst Operations Console:

1. Open your browser of choice and load the MobileFirst Operations Console using the address `http://localhost:9080/mfpconsole/`. You can also open the console from the **Command-line** using the CLI command `mfpdev server console`.
2. Click the "New" button next to "Applications" to create a new application and follow the on-screen instructions.
3. Once the application is registered, navigate to the application's **Configuration Files** tab and copy or download the `mfpclient.resw` file. Follow the onscreen instructions to add the file to your project.

Updating the MobileFirst Native SDK

To update the MobileFirst Native SDK with the latest release, run the following command from the root folder of the Visual Studio project in a **Command-line** window:

```
Nuget update
```

Generated MobileFirst Native SDK artifacts

mfpclient.resw

Located within the project in the `strings` folder, this file contains server connectivity properties and is user-editable:

- `protocol` – The communication protocol to MobileFirst Server. Either `HTTP` or `HTTPS`.
- `WlAppId` - The identifier of the application. This should be same as the Application identifier in the server.
- `host` – The hostname of the MobileFirst Server instance.
- `port` – The port of the MobileFirst Server instance.
- `wlServerContext` – The context root path of the application on the MobileFirst Server instance.
- `languagePreference` - Sets the default language for client sdk system messages

Tutorials to follow next

With the MobileFirst Native SDK now integrated, you can now:

- Review the Adapters development tutorials ([../adapters/](#))
- Review the Authentication and security tutorials ([../authentication-and-security/](#))
- Review the Notifications tutorials ([../notifications/](#))
- Review All Tutorials ([../all-tutorials](#))