

## Link-OS SDK for Xamarin README

This readme is specific to the LinkOS Xamarin SDK. This SDK is a Xamarin PCL in the plugin format. Also included in the files is a sample app showing use of specific APIs.

**Note:** The Developer Demo application is provided AS-IS, for example purposes only.

### Installation Instructions

This SDK is installed as a NuGet package into MS Visual Studio. Currently this is the only correct way to install the SDK.

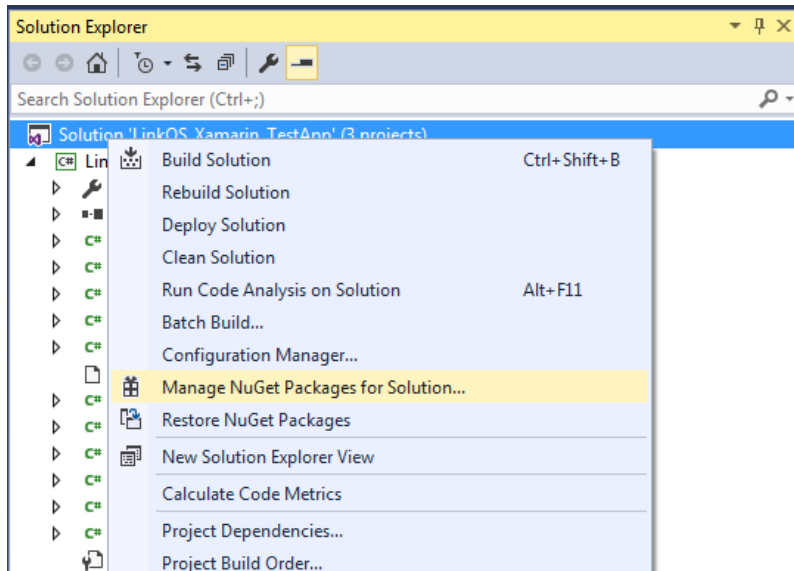
### Prerequisites

- Visual Studio 2012 – 2015 ( <https://www.visualstudio.com/post-download-vs?sku=community&clid=0x409> ) installed.
- Xamarin for Visual Studio (<https://xamarin.com/download> ) installed.
- A Business or Enterprise license for Xamarin and logged in to Visual Studio.
- A current version of NuGet plugin for Visual Studio (<https://visualstudiogallery.msdn.microsoft.com/5d345edc-2e2d-4a9c-b73b-d53956dc458d> ) installed.

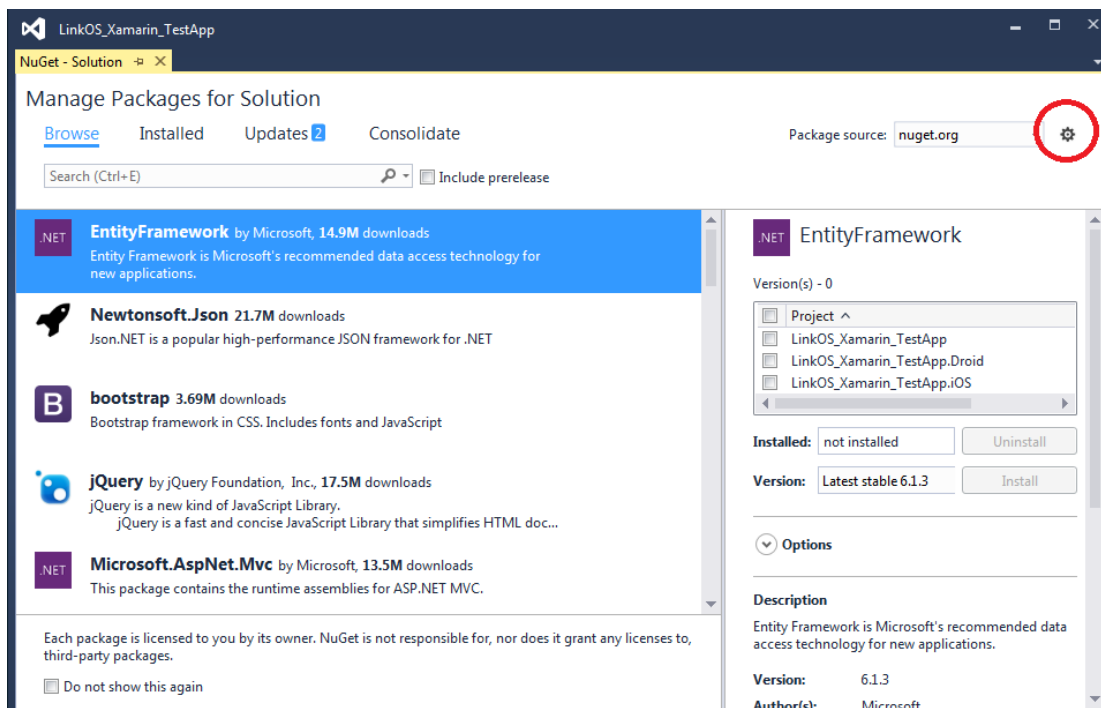
## Installation Steps

From a local nupkg package file:

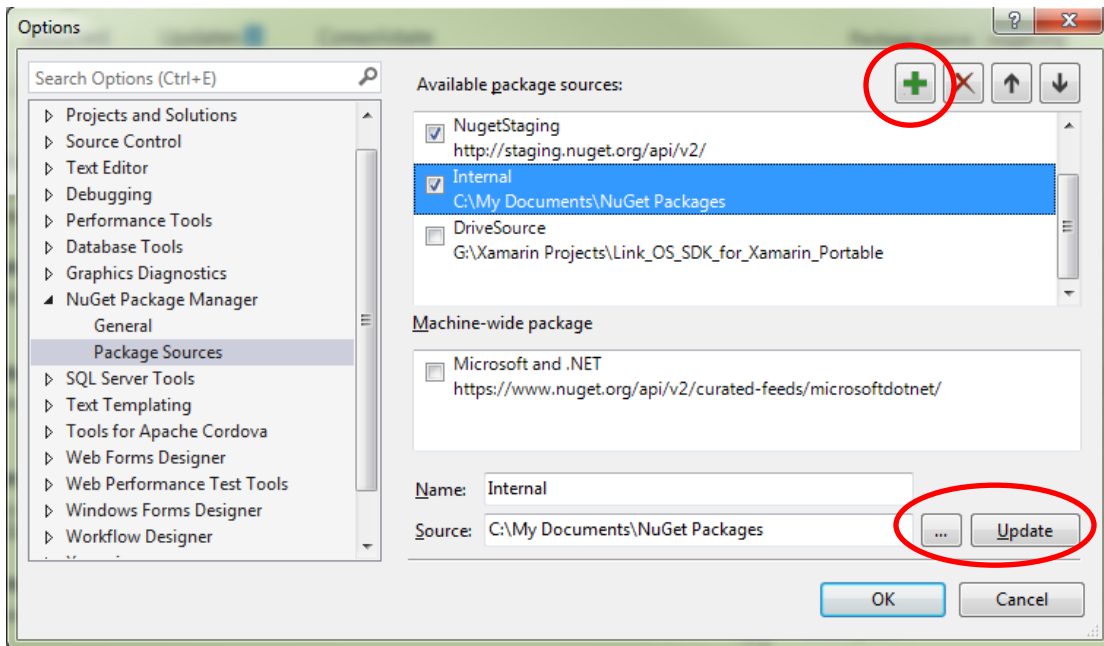
1. Open Visual Studio and create a new solution, open your own Xamarin solution, or open the Zebra developer demo included.
  - This solution can be a Xamarin Forms (Portable) app, Android app, or iOS app.
2. From the Solution Explorer, right click the top level solution. Do not manage from one of the projects. It will not work properly.
3. Click “Manage NuGet Packages for Solution...”



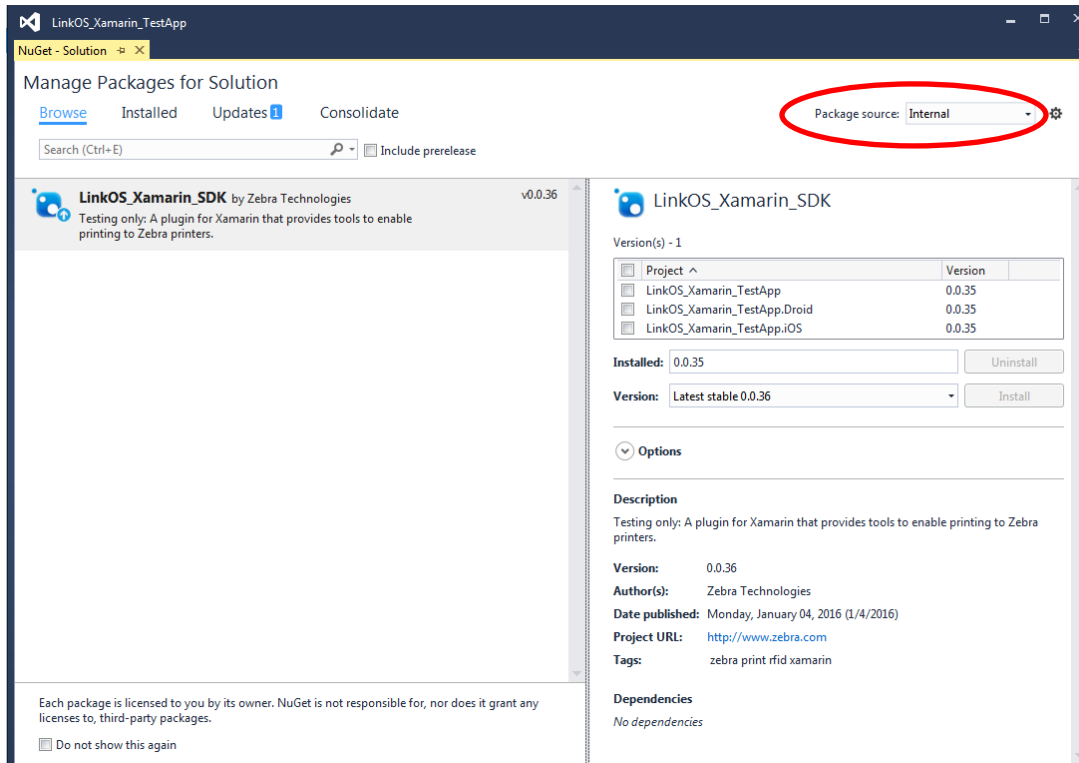
4. Click the source manager for the Package Sources ( button in red circle ).



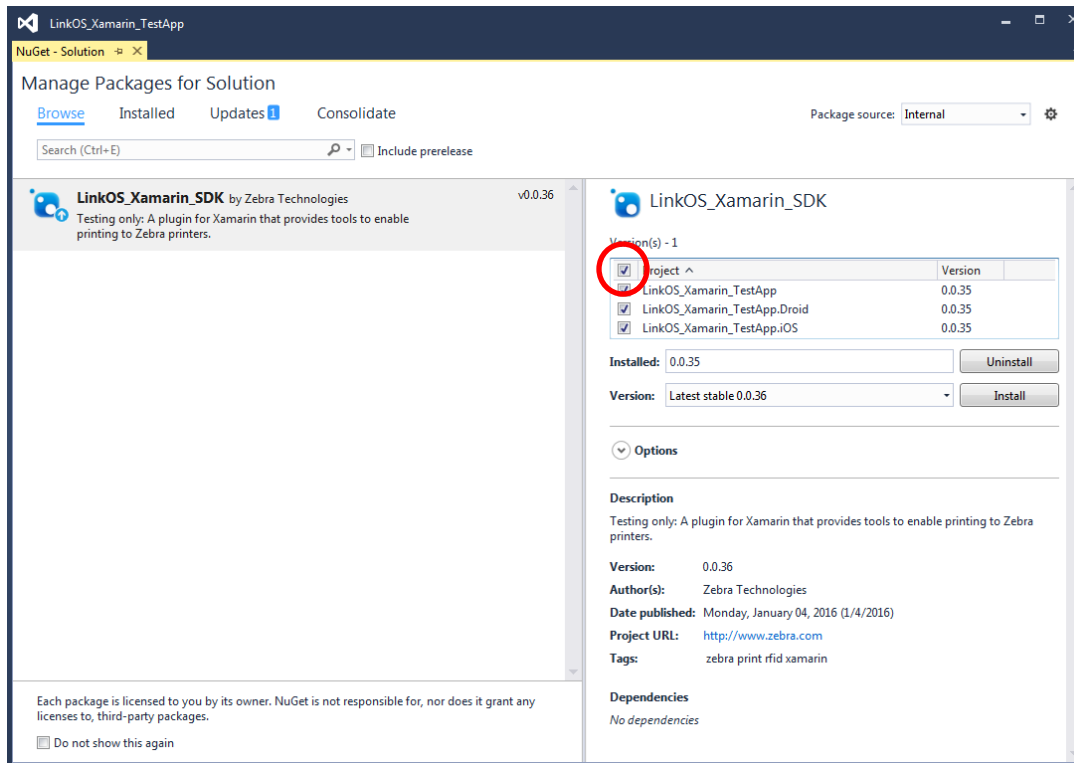
5. Click the Add package sources button
6. Type in or browse to the folder you installed the LinkOS\_Xamarin.nupkg file to. Give the location a name.



7. Click Update
8. Click OK
9. Change the Package source to your new package location. You should now see the LinkOS\_Xamarin\_SDK package in the packages list.

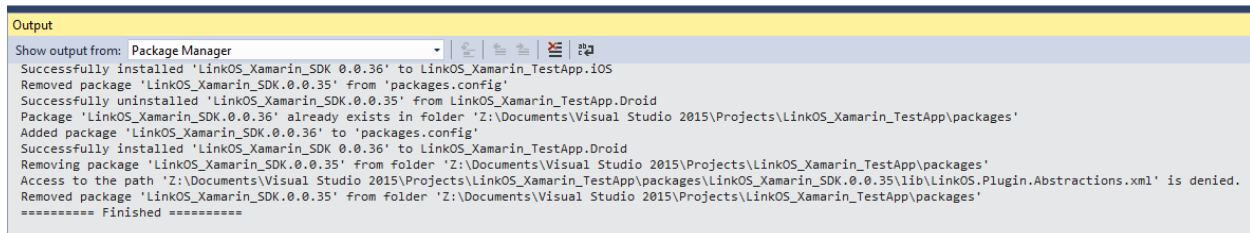


10. Click the Project checkbox to install the entire package.



11. Click the Install button next to the version.

12. The Package Manager will install all the appropriate references to the libraries. Wait till the console says it's finished. You are now capable of using the Link-OS API's in a Xamarin project.



## Updating the SDK package

If Zebra updates the SDK (and we will!), follow these steps to update your installation:

Follow installation steps 1-3, then 9-12. Notes

## Notes

The API's are documented in the documentation folder.

**Note:** There are several API's that may not work as expected with non-Link-OS printers. These API's are documented in the API docs. For a full list of Link-OS printers see the [Link-OS website](#).

**Note:** There are several differences between usage on Android and iOS for this API, please see the documentation for full details. Notes below are general to Android and iOS Bluetooth development.

**ZEBRA** Zebra LinkOS Multiplatform SDK for Xamarin 1.0.57  
Link-OS Portable Class Library plugin for Xamarin

Main Page Classes

▼ Zebra LinkOS Multiplatform SDK for Xamarin

- ▶ Namespace Members
- ▼ Classes
  - ▼ Class List
    - ▶ LinkOS
    - Class Index
    - ▶ Class Hierarchy
    - ▶ Class Members

### Zebra LinkOS Multiplatform SDK for Xamarin Documentation

Provides classes for interfacing with Zebra printers using Xamarin as a development tool.

I want to...

- Add LinkOS\_Xamarin\_SDK.nupkg to my development environment project
- [Print over TCP/IP](#)
- [Print over Bluetooth](#)
- [Create and print formats](#)
- [Send files to the printer](#)
- [Query printer status](#)
- [Print graphics](#)
- [Find more information](#)

## For Android development

There are a few extra things to keep in mind when developing for Android.

1. This is necessary for Bluetooth printing. In the Droid project, update the Android Manifest. Make sure to include all BLUETOOTH permissions and with Android v23- ACCESS\_FINE\_LOCATION.
2. For USB printing you will need to include in the AndroidManifest: `<uses-feature android:name="android.hardware.usb.host" />` You also may want to include an intent filter on: `android.hardware.usb.action.USB_DEVICE_ATTACHED`. The Zebra device filter vendor id is 2655.
3. Also may want to include READ\_EXTERNAL\_STORAGE permission as well for retrieving formats and graphics from Android.

## For iOS development

There are a few extra things to keep in mind when developing for iOS. These are necessary for Bluetooth printing.

1. Before debugging, update the Info.plist file by opening it in t XML editor or text editor. Add the following to the <dict> section. This is to allow your app to access the Bluetooth port.

```
<key>UIBackgroundModes</key>
-<array><string>external-accessory</string></array>
<key>UISupportedExternalAccessoryProtocols</key>
-<array><string>com.zebra.rawport</string>
<string>com.zebra.protocols</string></array>
```

2. Due to Apple API restrictions, you must pair the Bluetooth printer in the iOS Settings before you can use it in your app.

## Document Control

Version	Date	Description
1	January, 2016	Initial Release
2	January, 2016	Added Android and iOS notes
3	March, 2016	Added notes and updated screenshot for docs.

All links and information correct at time of writing

Created for the Zebra Global ISV Program by Zebra Development Services