# User Guide

# Datamax-O’Neil SDK

# Version 2.x.x.x

# TABLE OF CONTENTS

## Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

## Getting Started . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

## Datamax-O’Neil SDK API Guide . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

## Establishing Connection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

## Creating a Print Job for Printing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

## Printing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

## Querying . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

## Terminating Connection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

# OVERVIEW

This SDK is for developers who want to print or query to Datamax-O’Neil printer without knowledge of the printer’s language.

## Requirements

* Development Environment
  + Windows XP,7, and 8
  + OS X 10.8+
* IDE
  + Visual Studios 2008 (C#,C++)
  + Visual Studios 2013 (for Windows Store)
  + Eclipse (Java)
  + Android Studio (Android)
  + Xcode 5.1 (iOS)
* .NET Framework 3.5, .NET Framework 4.5, and .NET Compact Framework 2.0 or newer
* Android SDK for Android 2.1+ and above. See link for instructions: <http://developer.android.com/sdk/index.html?hl=sk>
* Java JDK 6+

## Features

* Targeted Platforms – iOS, Android, Windows Desktop, Windows 8 PC, Windows 8 Devices (x86 processor)
* Supports serial, Bluetooth Microsoft Stack, and TCP/IP connections (**NOTE: For the iOS SDK, BLUETOOTH CONNECTION IS ONLY SUPPORTED FOR APEX PRINTERS)**
* Printing support for Legacy Portable Printers (OC2, OC3, MF4Te, etc…), Apex Series (ExPCL Language), and DPL printers (RL3, RL4, etc…).
  + Supports printing Text, Barcodes, Images, and Graphics. PDF printing is supported on Android, Java, .NET C# Windows Desktop, .NET C# for Windows Store Apps, and iOS
* Query support for Legacy Printers (OC2, OC3, MF4Te, etc...), Apex Series (ExPCL Language), and DPL printers (RL3, RL4, etc...).

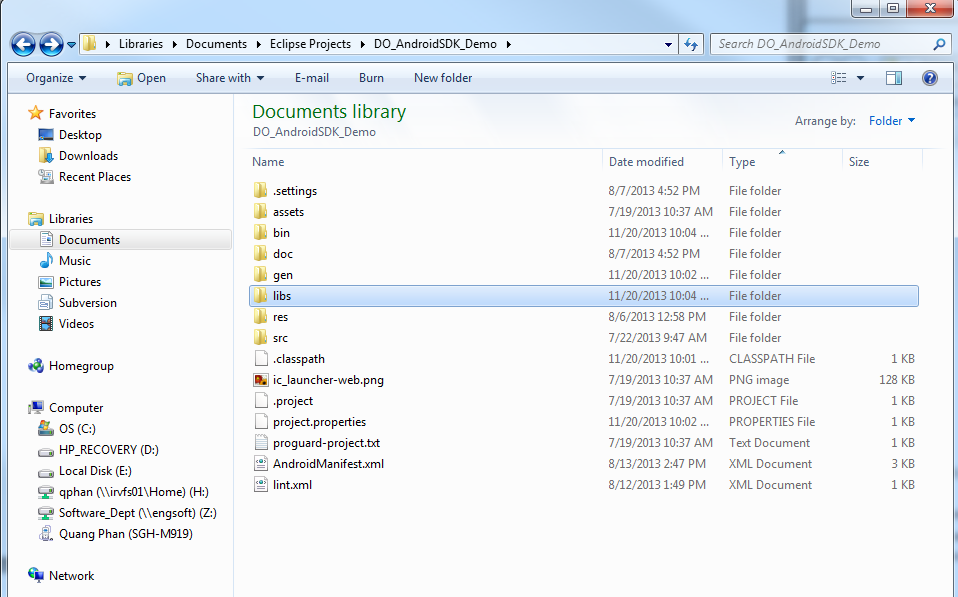
# GETTING STARTED

## Setup new/existing application to use Datamax-O’Neil SDK

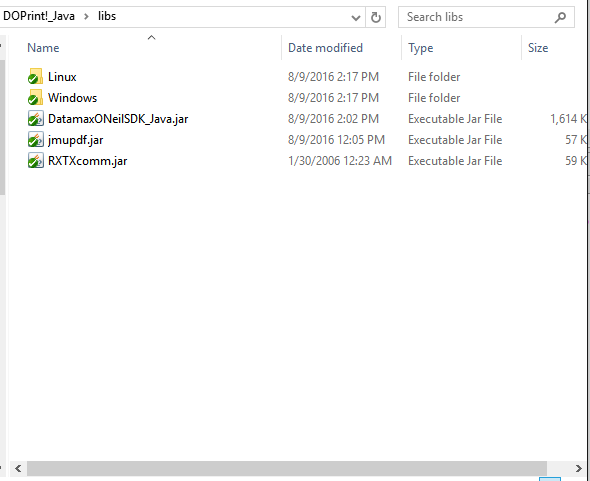
To use the Datamax-O’Neil SDK API in your application, you must reference the Datamax-O’Neil SDK library in your application. Below are steps to set up your application for each SDK available.

### Datamax-O’Neil Java SDK

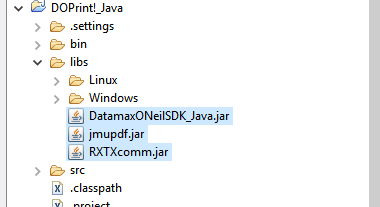
1. Create a new Java projector open an existing project using **Eclipse.**
2. Navigate to the project’s directory and open the **libs** folder.



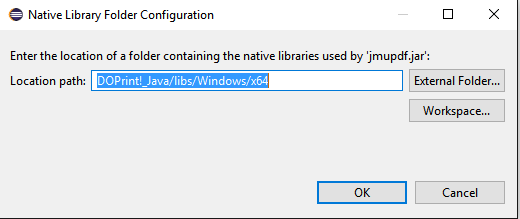
1. Copy and replace your project’s **libs folder** with the provided **libs folder**.Copy **DatamaxONeilSDK\_Java.jar** to your project’s **libs folder**. Your project’s **libs folder** should look like the following:

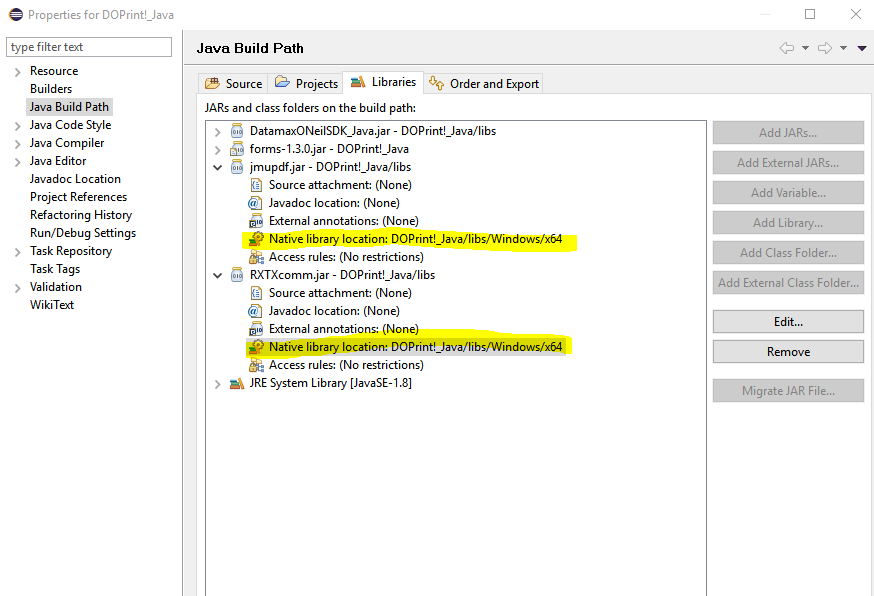


1. Right-click project in **Package Explorer** and click on **Properties**. Click on **Java Build Path** and click **Add JARs…** Navigate to your project’s **libs folder** and add the following **jar** libraries:



1. (Optional) For serial RS232 connection and PDF printing support, you need to link the **RXTXcomm.jar** and **jmupdf.jar** with its native library (platform dependent). For each library, double-click on **Native library location** and specify the correct library location for the targeted platform. (Eg. For application running on Windows 64-bit, select **{PROJECT\_ROOT}/libs/Windows/x64** where **{PROJECT\_ROOT}** is root directory of your Java Project).





1. Import the following to use all of the SDK functionalities:

**import** honeywell.connection.ConnectionBase;

**import** honeywell.connection.Connection\_Bluetooth;

**import** honeywell.connection.Connection\_TCP;

**import** honeywell.printer.DocumentDPL;

**import** honeywell.printer.DocumentDPL.\*;

**import** honeywell.printer.DocumentEZ;

**import** honeywell.printer.DocumentLP;

**import** honeywell.printer.DocumentExPCL\_LP;

**import** honeywell.printer.DocumentExPCL\_PP;

**import** honeywell.printer.DocumentExPCL\_PP.\*;

**import** honeywell.printer.ParametersDPL;

**import** honeywell.printer.ParametersDPL.\*;

**import** honeywell.printer.ParametersEZ;

**import** honeywell.printer.ParametersExPCL\_LP;

**import** honeywell.printer.ParametersExPCL\_LP.\*;

**import** honeywell.printer.ParametersExPCL\_PP;

**import** honeywell.printer.ParametersExPCL\_PP.\*;

**import** honeywell.printer.UPSMessage;

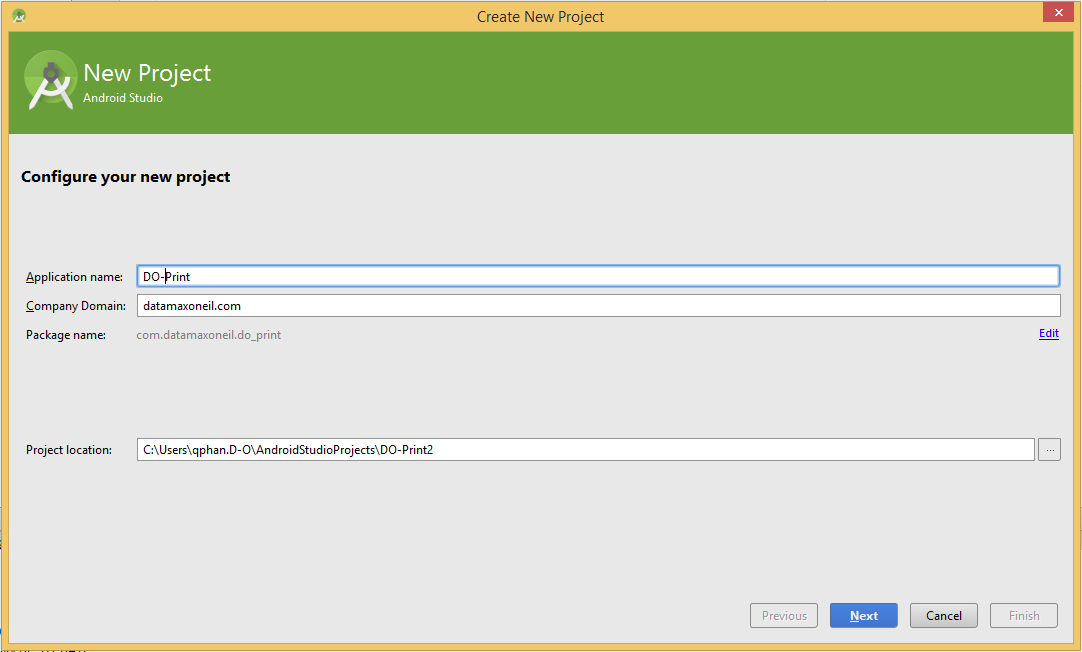
**import** honeywell.printer.configuration.dpl.\*;

**import** honeywell.printer.configuration.ez.\*;

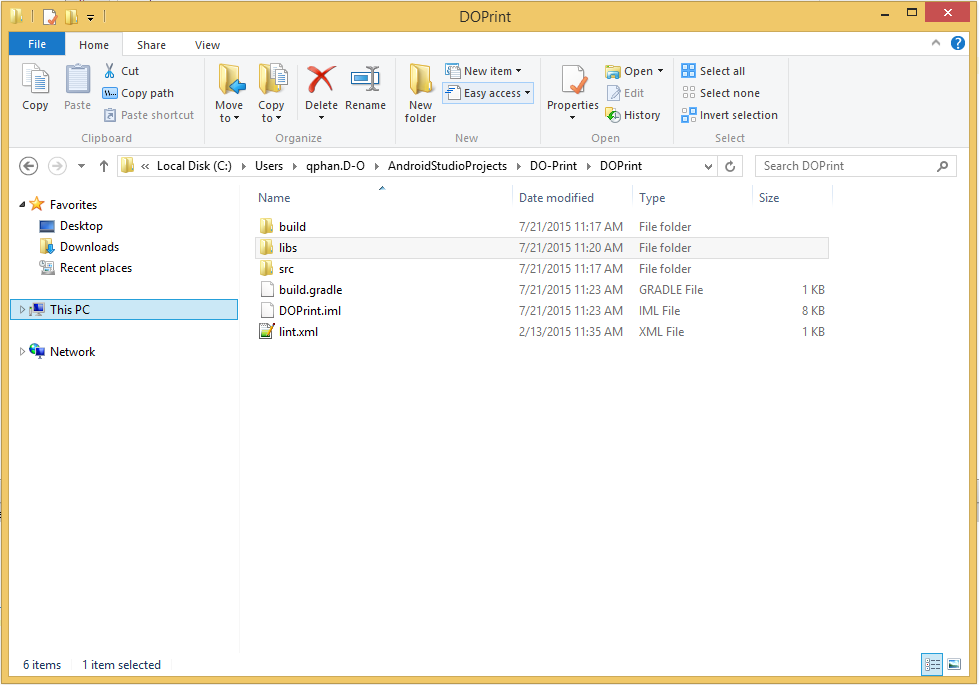
**import** honeywell.printer.configuration.expcl.\*;

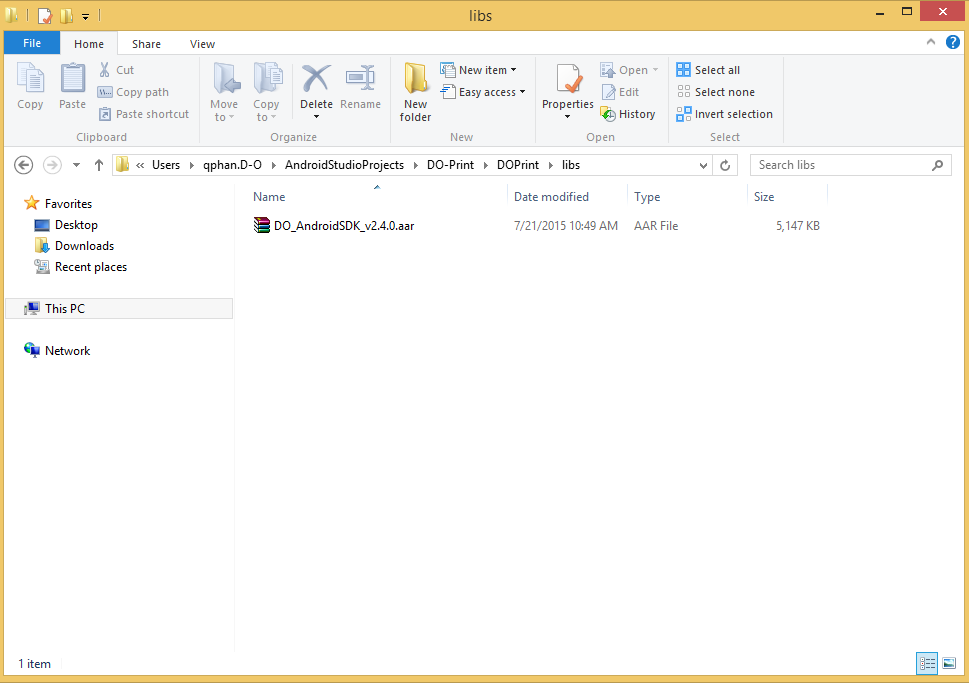
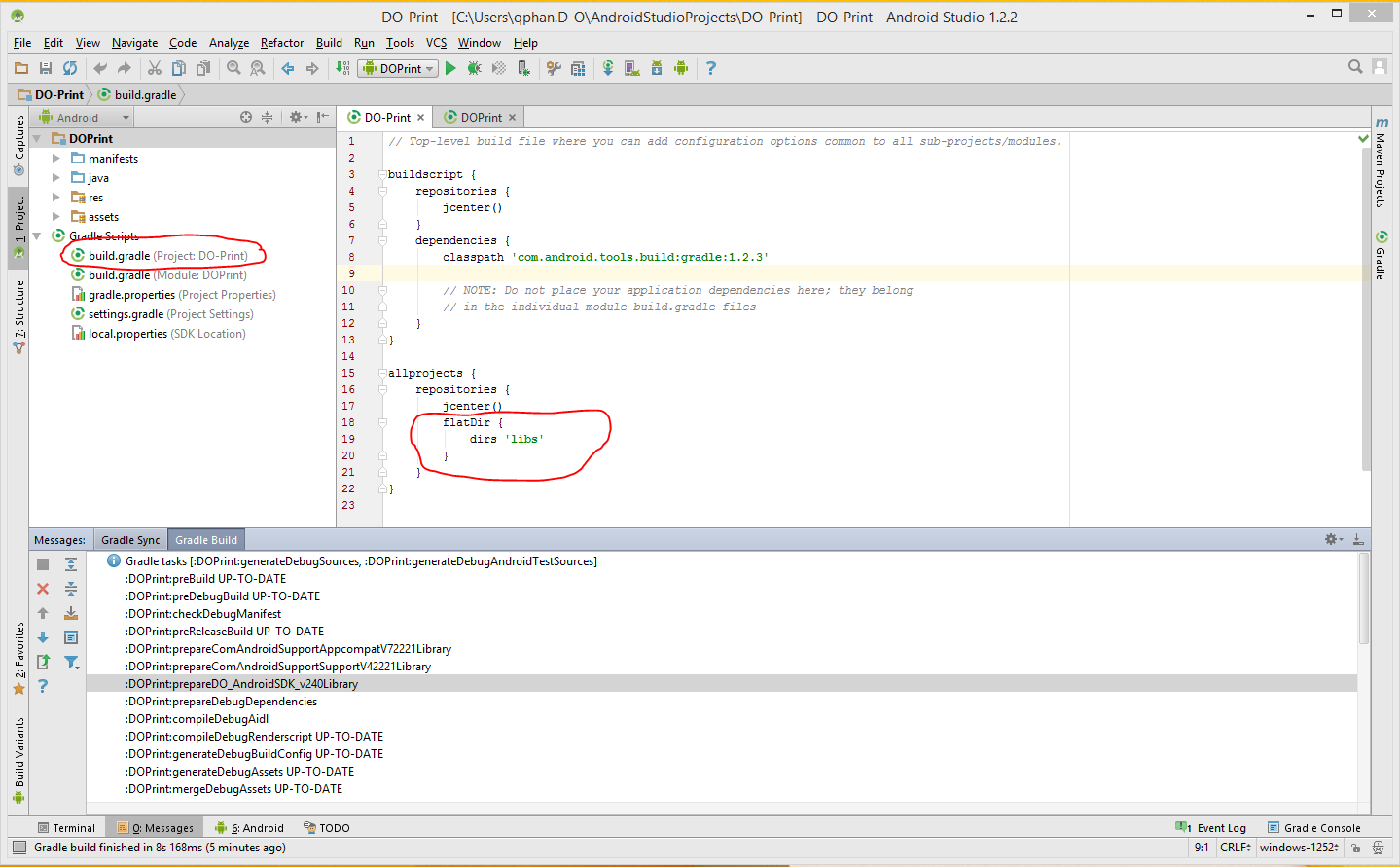
### Datamax-O’Neil Android SDK

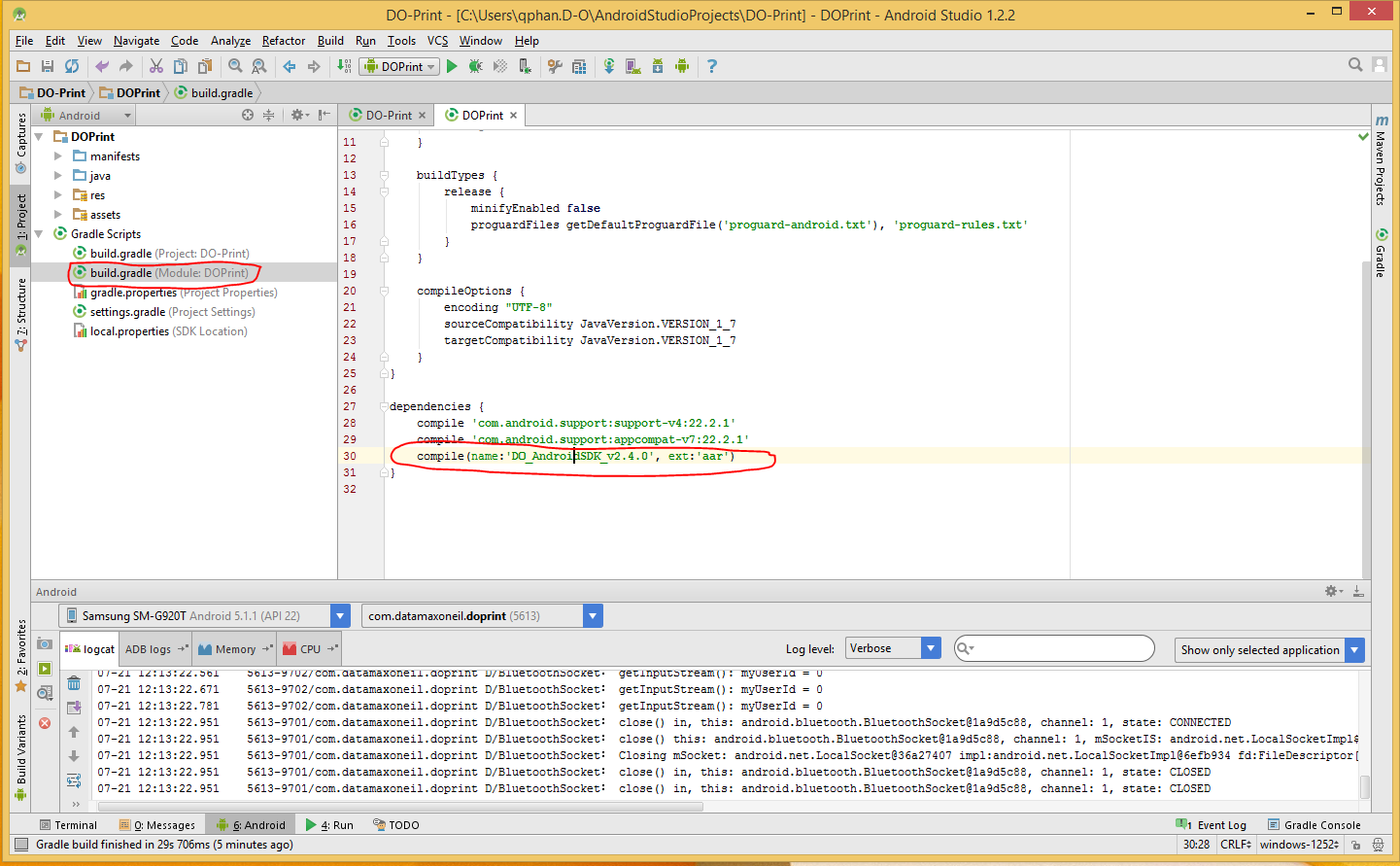
1. Click **File-> New -> New Project** or open an existing project**.**



1. Navigate to the project’s directory and open/create the **libs** folder inside your application module folder.



1. Copy **DO\_AndroidSDK\_v2.4.x.aar** file into the libs folder that you created. 
2. On the **Project Panel** in Android Studio, open the **build.gradle** of the project. Add the below to the **repositories** section
3. On the **Project Panel** in Android Studio, open the **build.gradle** of the application module. Add the following to the **dependencies** section: compile(name:**'DO\_AndroidSDK\_v2.4.0'**, ext:**'aar'**). **Note: ‘DO\_AndroidSDK\_v2.4.0’ will change based on the version.**



1. Import the following to use all of the SDK functionalities:

**import honeywell.**connection.ConnectionBase;

**import honeywell.**connection.Connection\_Bluetooth;

**import honeywell.**connection.Connection\_TCP;

**import honeywell.**printer.DocumentDPL;

**import honeywell.**printer.DocumentDPL.\*;

**import honeywell.**printer.DocumentEZ;

**import honeywell.**printer.DocumentLP;

**import honeywell.**printer.DocumentExPCL\_LP;

**import honeywell.**printer.DocumentExPCL\_PP;

**import honeywell.**printer.DocumentExPCL\_PP.\*;

**import honeywell.**printer.ParametersDPL;

**import honeywell.**printer.ParametersDPL.\*;

**import honeywell.**printer.ParametersEZ;

**import honeywell.**printer.ParametersExPCL\_LP;

**import honeywell.**printer.ParametersExPCL\_LP.\*;

**import honeywell.**printer.ParametersExPCL\_PP;

**import honeywell.**printer.ParametersExPCL\_PP.\*;

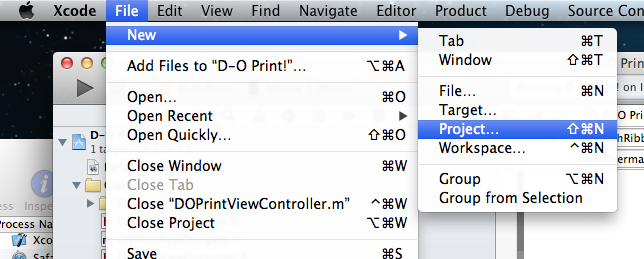
**import honeywell.**printer.UPSMessage;

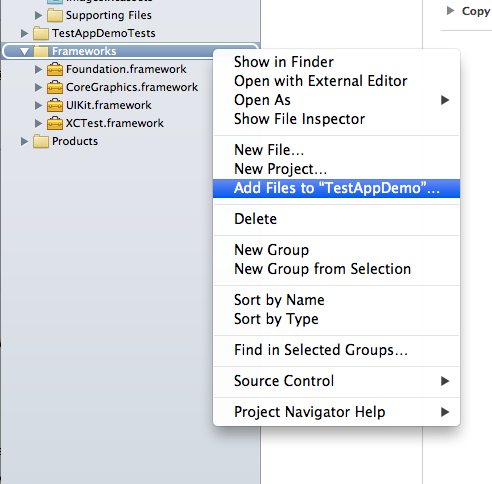
**import honeywell.**printer.configuration.dpl.\*;

**import honeywell.**printer.configuration.ez.\*;

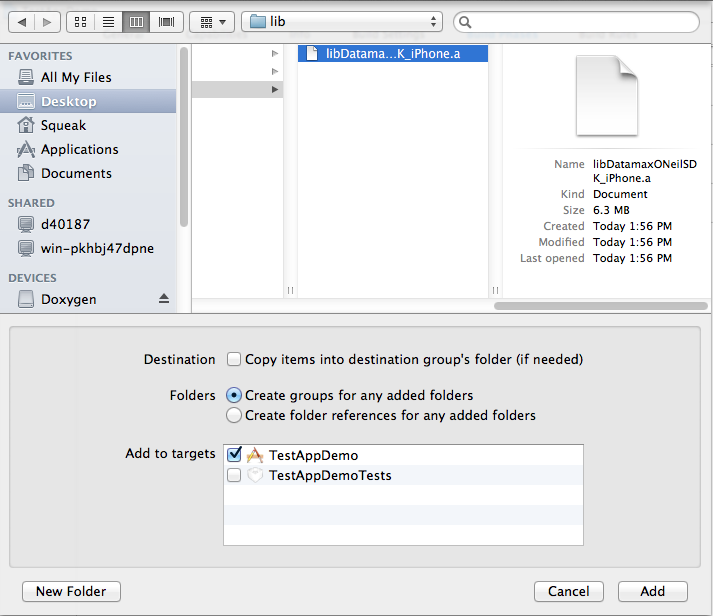
**import honeywell.**printer.configuration.expcl.\*;

### Datamax-O’Neil iOS SDK

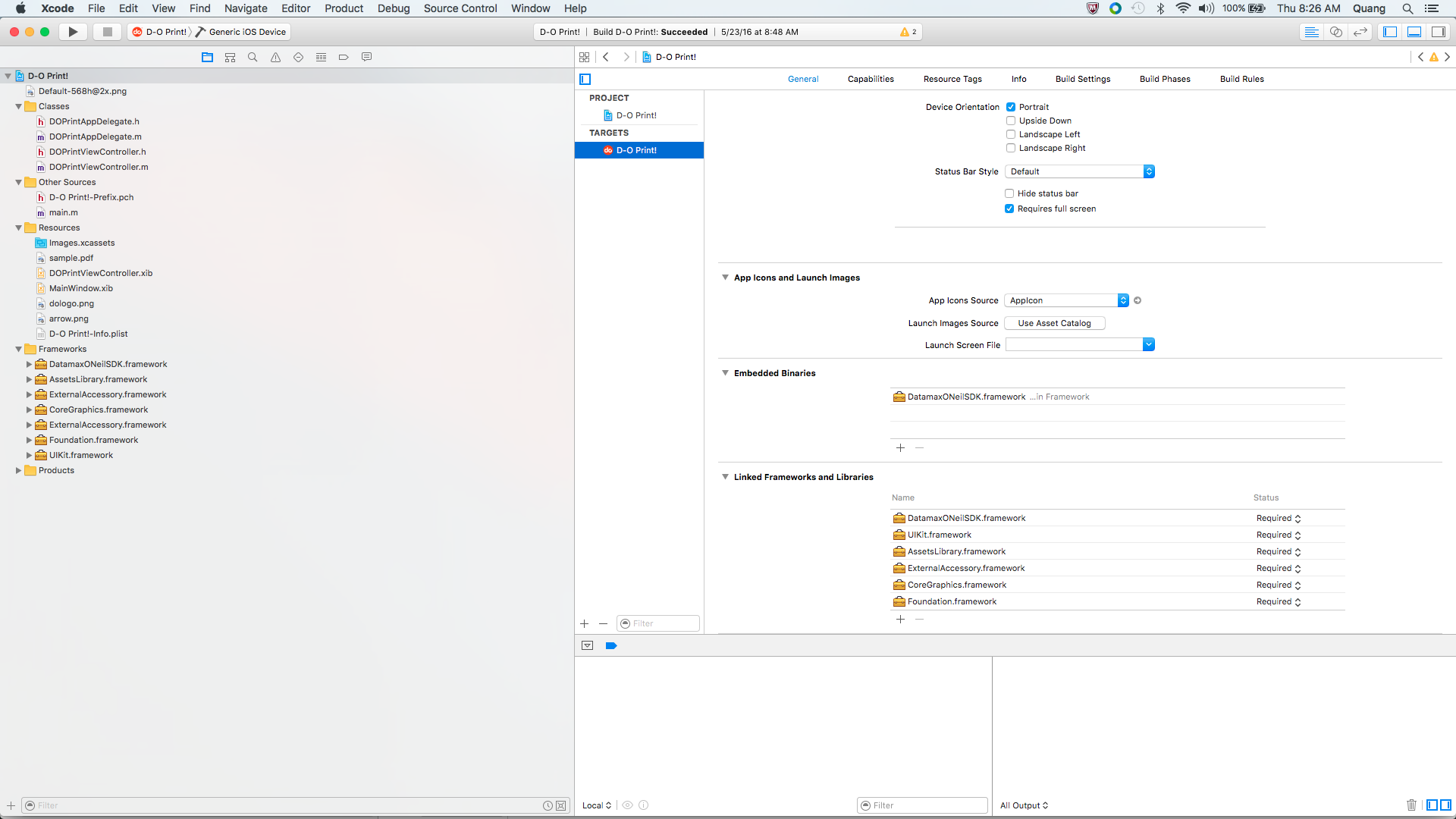
1. Create a new project or open an existing project 
2. Once created, you should the project settings on the screen. Right-click on the Framework folder on project explorer and add files.



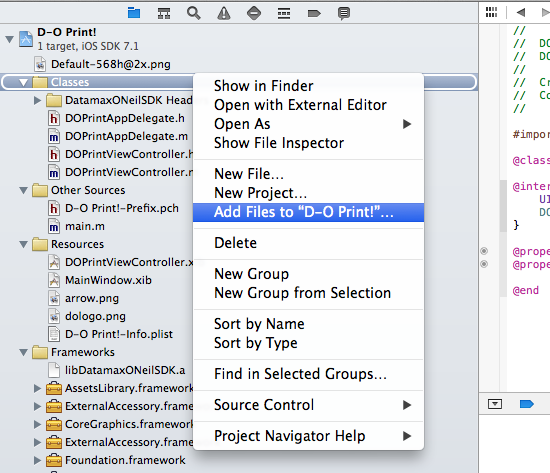
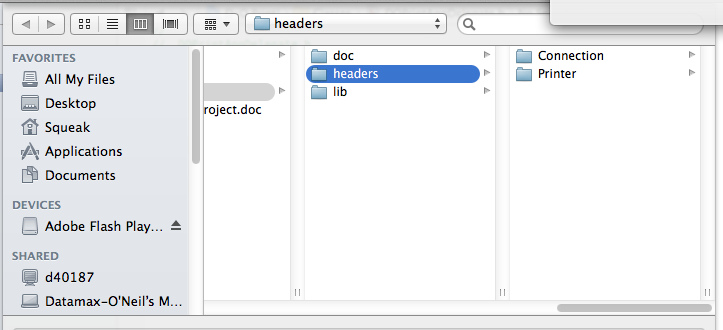
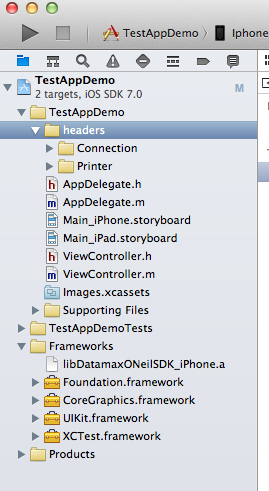
* 1. For Static Library, navigate to the **libDatamaxOneilSDK.a** file and add that to your project.



* 1. For Framework, Navigate to the **DatamaxONeilSDK.framework** file and add that to your project. In your project settings, you should see the following:

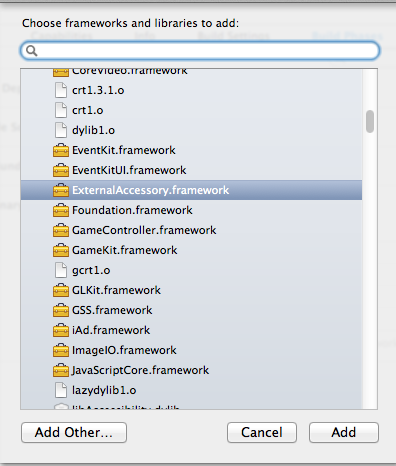


1. For Static Library, add the SDK headers to your project (Step is only for static library version of iOS SDK). Right-click on the project and select add files to Project. Navigate to the provided **headers** folder and add it to the project. It should look like below.

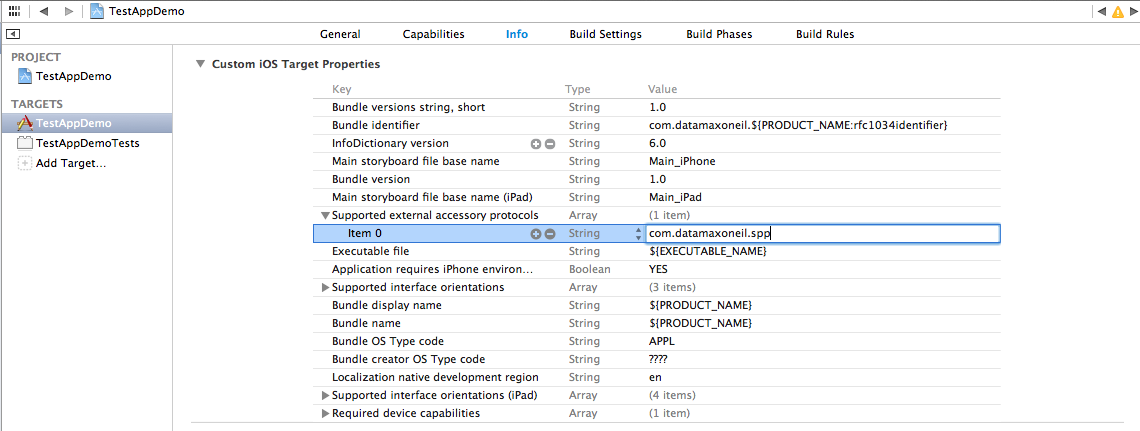
 

1. If you want to communicate to printer via Bluetooth, you need to add the **ExternalAccessory.framework** to this project. (Apex 2i, 3i, 4i)

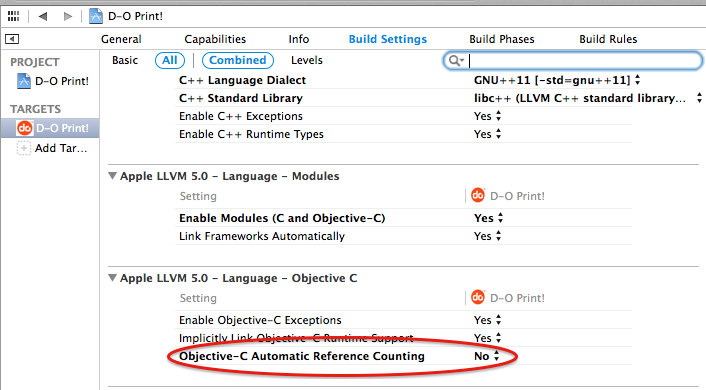
* In project settings, go to **Build Phases** and under **Link Binary with Libraries**, add the **ExternalAccessory.framework** to the project.



* In project settings, go to **Info**. Add the following property (right-click and add row): **Supported external accessory protocols.**  Under **Item 0**, type in **“com.datamaxoneil.spp”**.



1. Under **Build Settings** of the project settings, scroll down to **Objective-C Automatic Reference Counting** (under **Apple LLVM 5.0 – Language – Objective-C**) and set the value to **Yes**.



1. To use the print feature from DO SDK, include appropriate header file (\*.h) in the implementation file (\*.m).
   1. Static Library

#import <ExternalAccessory/ExternalAccessory.h>

#include "Connection\_BluetoothEA.h"

#include "Connection\_TCP.h"

#include "ParametersEZ.h"

#include "ParametersLP.h"

#include "DocumentEZ.h"

#include "DocumentLP.h"

#include "ParametersExPCL\_PP.h"

#include "ParametersExPCL\_LP.h"

#include "DocumentExPCL\_PP.h"

#include "DocumentExPCL\_LP.h"

#include "DocumentDPL.h"

#include "ParametersDPL.h"

#include "UPSMessage.h"

#include "GeneralStatus\_ExPCL.h"

#include "VersionInformation\_ExPCL.h"

#include "MagneticCardData\_ExPCL.h"

#include "BluetoothConfiguration\_ExPCL.h"

#include "AvalancheSettings.h"

#include "BatteryCondition.h"

#include "BluetoothConfiguration.h"

#include "FontList.h"

#include "FormatList.h"

#include "GeneralConfiguration.h"

#include "GeneralStatus.h"

#include "GraphicList.h"

#include "IrDAConfiguration.h"

#include "LabelConfiguration.h"

#include "MagneticCardConfiguration.h"

#include "MagneticCardData.h"

#include "ManufacturingDate.h"

#include "MemoryStatus.h"

#include "PrinterOptions.h"

#include "PrintheadStatus.h"

#include "SerialNumber.h"

#include "SmartCardConfiguration.h"

#include "SSP\_CommParameters.h"

#include "TCPIPStatus.h"

#include "UpgradeData.h"

#include "VersionInformation.h"

#include "PrinterInformation\_DPL.h"

#include "MediaLabel\_DPL.h"

#include "SystemSettings\_DPL.h"

#include "SensorCalibration\_DPL.h"

#include "PrintSettings\_DPL.h"

#include "Fonts\_DPL.h"

#include "SerialPortConfiguration\_DPL.h"

#include "AvalancheEnabler\_DPL.h"

#include "AutoUpdate\_DPL.h"

#include "BluetoothConfiguration\_DPL.h"

#include "NetworkGeneralSettings\_DPL.h"

#include "NetworkWirelessSettings\_DPL.h"

#include "NetworkEthernetSettings\_DPL.h"

#include "Miscellaneous\_DPL.h"

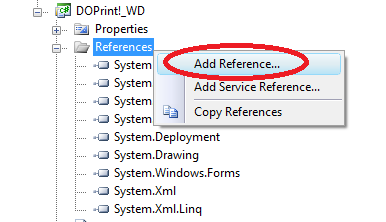
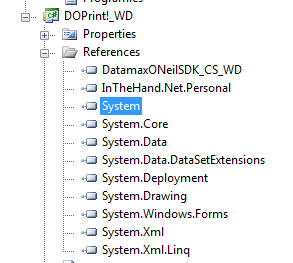
* 1. Framework

#import <ExternalAccessory/ExternalAccessory.h>

#import <DatamaxONeilSDK/DatamaxONeilSDK.h>

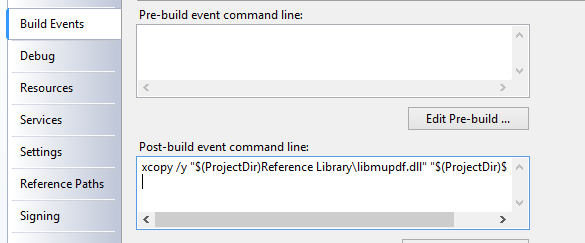
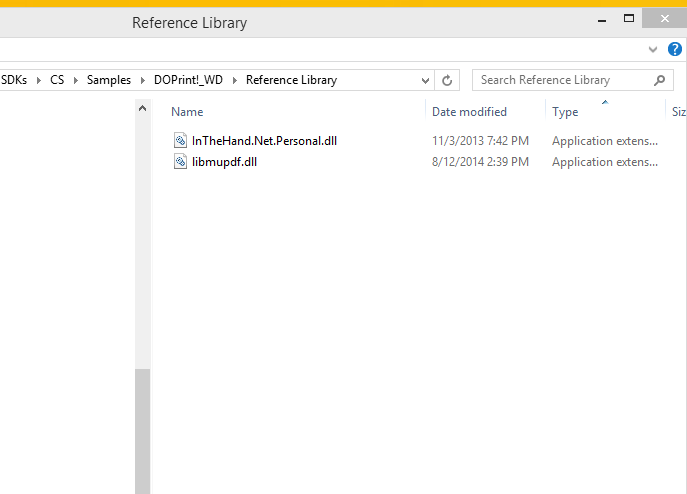
### Datamax-O’Neil Windows .NET C# SDK for Windows Desktop/ CE, Mobile 5.0+

1. Create a new project or open an existing C# project in Visual Studio 2008. Expand the **References** group in **Solution Explorer**. Under **Browse Tab,** navigate to the provided **DatamaxONeilSDK\_CS\_WD.dll (or DatamaxOneilSDK\_CS\_WM.dll for Win CE/Mobile project)** and **InTheHand.Net.Personal.dll** and add them to your project.

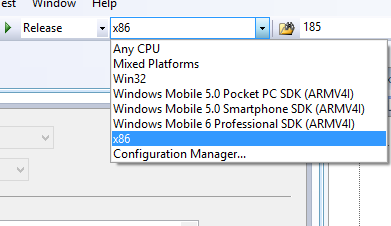
 

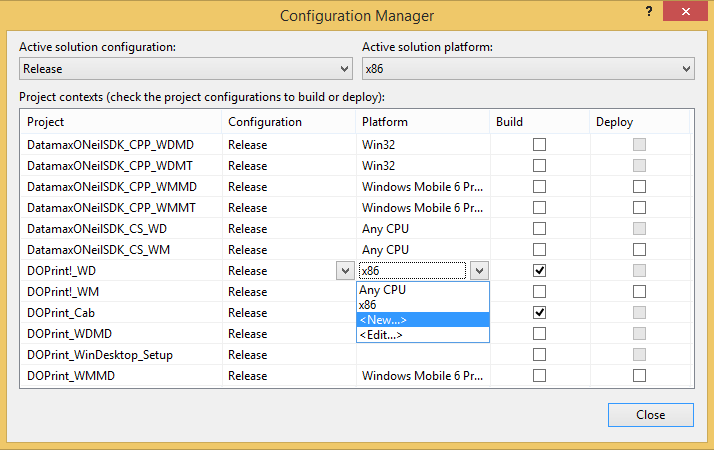
1. For **Windows Desktop**, copy the provided **libmudpf.dll** in the **Reference Library** folder to your project location.
   1. Right-click on your project inside Visual Studio and click **Properties.** Go to **Build Events,** and add the below to Post-build event. {LOCATION\_OF\_LIBMUDF } is to be replaced with the location of your libmupdf.dll inside your project. (eg. **Reference Library\libmupdf.dll**).

*xcopy /y "$(ProjectDir){LOCATION\_OF\_LIBMUDF }" "$(ProjectDir)$(OutDir)"*



* 1. On 64-bit machines, change the CPU platform of your .NET project from **Any CPU** to **x86**. If you do not see **x86**, go to the **Configuration Manager** and add it to your project.





1. In the source code of your project (eg. **DOPrint!\_WD.cs**), add the following namespaces to your project. This will allow you to call our SDK APIs in your application.

using Honeywell.Connection;

using Honeywell.Printer;

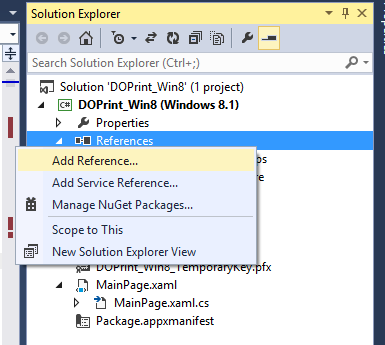
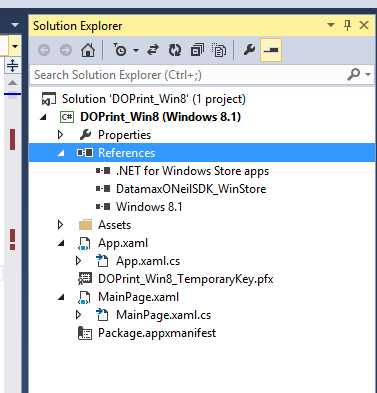
using Honeywell.Printer.Configuration.DPL;

using Honeywell.Printer.Configuration.EZ;

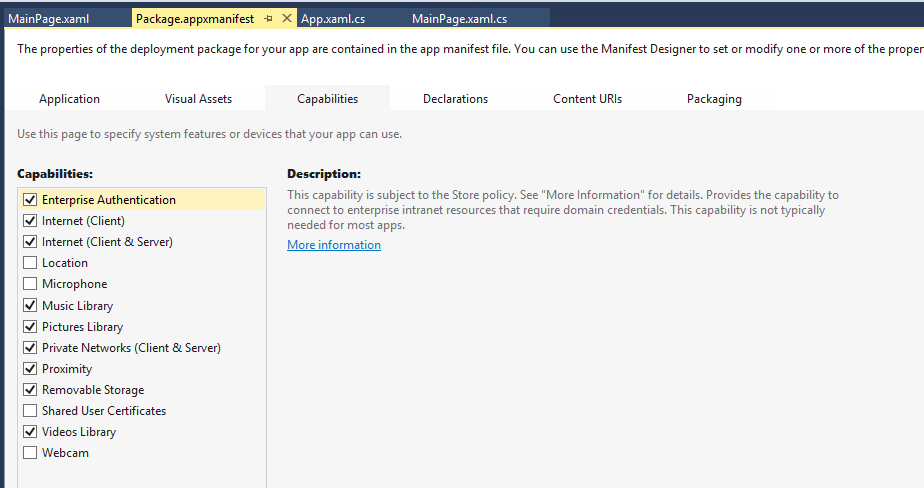
using Honeywell.Printer.Configuration.ExPCL;

### Datamax-O’Neil Windows .NET C# SDK for Windows Store App/Windows Phone 8

1. Create a new project or open an existing C# project in Visual Studio 2013. Expand the **References** group in **Solution Explorer**. Right-click on **References** and click **Add References.** Under **Browse,** navigate to the provided **DatamaxONeilSDK\_WinStore.dll (or DatamaxOneilSDK\_WinPhone)** and add them to your project.

1. In the Project manifest file (eg. **Package.appxmanifest**), click on **Capbabilities** and check the following if it is not already checked: **Internet, Internet (Client & Server), and Private Networks (Client & Server).** Other optional capabilities such as **Removable Storage** and **Music, Picture, and Video Libraries** may also be selected depending on your application.



1. For Bluetooth connections, right-click **Package.appxmanifest** and click **Open With 🡪 XML (Text) Editor**. Add the following to the capabilities section:

<m2:DeviceCapability Name="bluetooth.rfcomm">

<m2:Device Id="any">

<m2:Function Type="name:serialPort" />

</m2:Device>

</m2:DeviceCapability>



1. In the source code of your project (eg. **MainPage.xaml.cs**), add the following namespaces to your project. This will allow you to call our SDK APIs in your application.

using Honeywell.Connection;

using Honeywell.Printer;

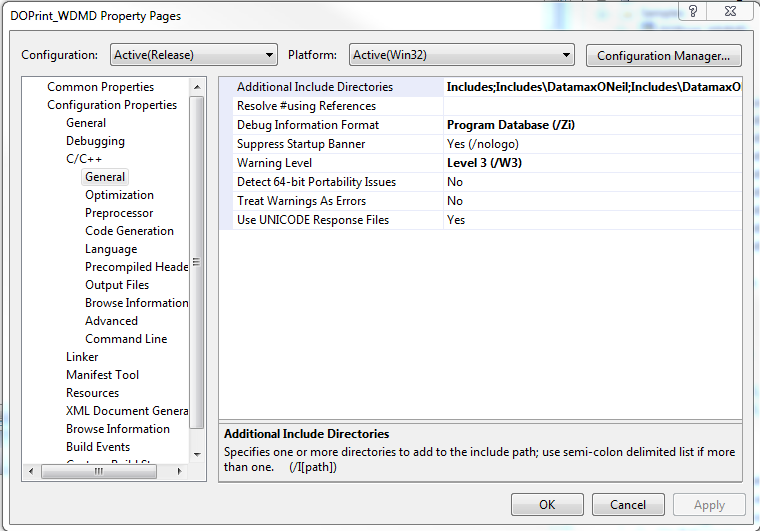
using Honeywell.Printer.Configuration.DPL;

using Honeywell.Printer.Configuration.ExPCL;

using Honeywell.Printer.Configuration.EZ;

### Datamax-O’Neil Windows C++ SDK for Windows Desktop/ CE, Mobile 5.0+

1. Create a new Visual C++ project or open an existing C++ project in Visual Studio 2008.
2. Right-click on your project and click **Properties** and configure your project as follow:
   1. If the application uses **MFC** or **ATL,** specify the following in **Configuration Properties 🡪General**
      1. If you are referencing libraries statically, set options to **Use MFC in a Static Library** or **Static Link to ATL**
      2. If you are referencing libraries dynamically, set option to **Use MFC in a Shared DLL** or **Dynamic Link to ATL**
   2. Under **Configuration Properties 🡪 C/C++ 🡪 General** on the left pane, specify the location to reference the provided **Includes** folder. For example, **Includes;Includes\DatamaxONeil;Includes\DatamaxONeil\Connection;Includes\DatamaxONeil\Printer;Includes\DatamaxONeil\Printer\Configuration\DPL;Includes\DatamaxONeil\Printer\Configuration\ExPCL;Includes\DatamaxONeil\Printer\Configuration\EZ**

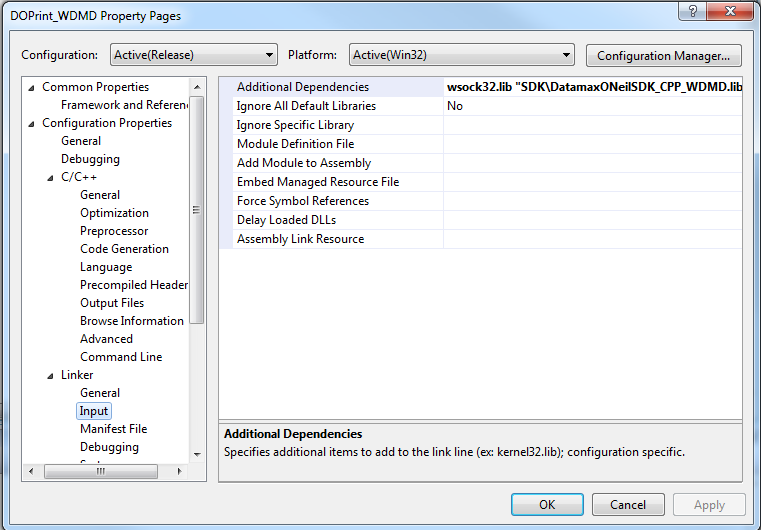


* 1. Under **Configuration Properties 🡪 C/C++ 🡪 Code Generation**, specify what runtime library you are going to use. If you are referencing libraries dynamically, set option to **Multi-threaded DLL (/MD).** If you are referencing libraries statically, set option to **Multi-threaded (/MT)**
  2. For Desktop Applications,
     1. Under **Configuration Properties 🡪 Linker 🡪 Input**
        1. If you are referencing libraries statically, add the following:

**wsock32.lib "{*location\_of\_sdk*}\DatamaxONeilSDK\_CPP\_WDMT.lib"**

* + - 1. If you are referencing libraries dynamically, add the following:

**wsock32.lib "{*location\_of\_sdk*}\DatamaxONeilSDK\_CPP\_WDMD.lib"**

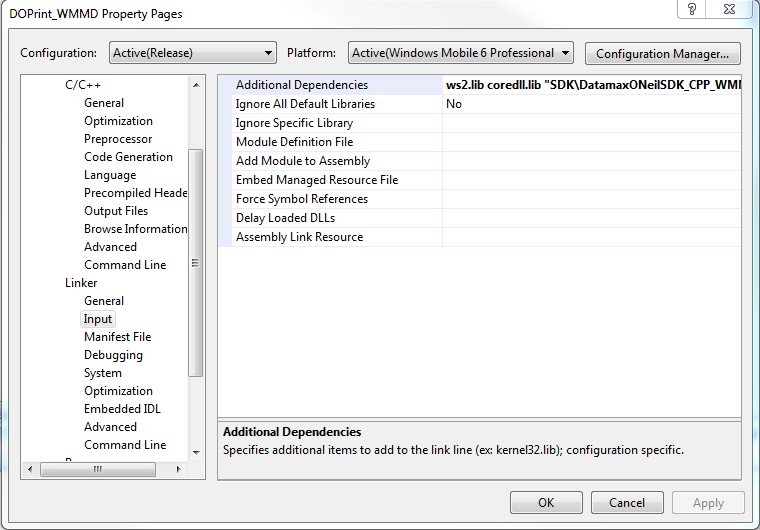


* 1. For Win CE/Mobile application
     1. Under **Configuration Properties 🡪 Linker 🡪 Input**
        1. If you are referencing libraries statically, add the following:

**ws2.lib coredll.lib "{*location\_of\_sdk*}\DatamaxONeilSDK\_CPP\_WMMT.lib"**

* + - 1. If you are referencing libraries dynamically, add the following:

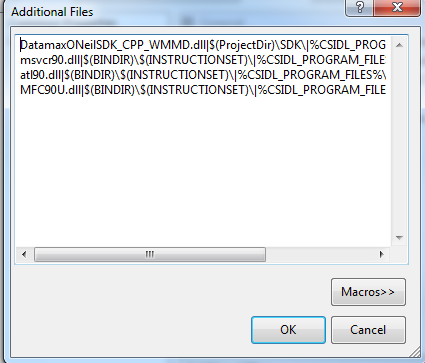
**ws2.lib coredll.lib "{*location\_of\_sdk*}\DatamaxONeilSDK\_CPP\_WMMD.lib"**



* + 1. Under **Configuration Properties 🡪 Deployment.** Add the following under **Additional Files** to deploy the DLL onto the device along with the application:
       1. If you are referencing libraries statically, add the following:

**DatamaxONeilSDK\_CPP\_WMMT.dll|$(ProjectDir)\SDK\|%CSIDL\_PROGRAM\_FILES%\$(ProjectName)|0**

* + - 1. If you are referencing libraries dynamically, add the following: **DatamaxONeilSDK\_CPP\_WMMD.dll|$(ProjectDir)\SDK\|%CSIDL\_PROGRAM\_FILES%\$(ProjectName)|0**



1. In the implementation file (\*.cpp) of your source code, add the SDK headers. You will then be able to use the Datamax-O’Neil SDK API.

//Connection classes

#include "ConnectionBase.h"

#include "Connection\_Serial.h"

#include "Connection\_BluetoothMsftStack.h"

#include "Connection\_BluetoothSPP.h"

#include "Connection\_TCP.h"

//Document and Parameter classes

#include "Parameters.h"

#include "ParametersEZ.h"

#include "ParametersDPL.h"

#include "ParametersLP.h"

#include "ParametersExPCL\_LP.h"

#include "ParametersExPCL\_PP.h"

#include "Document.h"

#include "DocumentEZ.h"

#include "DocumentDPL.h"

#include "DocumentLP.h"

#include "DocumentExPCL\_LP.h"

#include "DocumentExPCL\_PP.h"

//Query EZ

#include "FontList.h"

#include "BatteryCondition.h"

#include "BluetoothConfiguration.h"

#include "CardReaderStatus.h"

#include "ConfigurationVersion.h"

#include "FontList.h"

#include "FormatList.h"

#include "GeneralConfiguration.h"

#include "GeneralStatus.h"

#include "GraphicList.h"

#include "IrDAConfiguration.h"

#include "LabelConfiguration.h"

#include "MagneticCardConfiguration.h"

#include "MagneticCardData.h"

#include "ManufacturingDate.h"

#include "MemoryStatus.h"

#include "PrinterOptions.h"

#include "PrinterState.h"

#include "PrintheadStatus.h"

#include "SerialNumber.h"

#include "SmartCardConfiguration.h"

#include "TCPIPStatus.h"

#include "UpgradeData.h"

#include "VersionInformation.h"

#include "AvalancheSettings.h"

#include "ListFilesShort.h"

#include "ListFilesLong.h"

#include "SSP\_CommParameters.h"

//Query DPL

#include "PrinterData\_DPL.h"

#include "PrinterState\_DPL.h"

#include "AutoUpdate\_DPL.h"

#include "AvalancheEnabler\_DPL.h"

#include "BluetoothConfiguration\_DPL.h"

#include "Fonts\_DPL.h"

#include "Images\_DPL.h"

#include "LanguageFiles\_DPL.h"

#include "MediaLabel\_DPL.h"

#include "MemoryModules\_DPL.h"

#include "Miscellaneous\_DPL.h"

#include "NetworkEthernetSettings\_DPL.h"

#include "NetworkWirelessSettings\_DPL.h"

#include "NetworkGeneralSettings\_DPL.h"

#include "PrinterInformation\_DPL.h"

#include "PrintSettings\_DPL.h"

#include "SerialPortConfiguration\_DPL.h"

#include "SensorCalibration\_DPL.h"

#include "SystemSettings\_DPL.h"

//Query ExPCL

#include "BluetoothConfiguration\_ExPCL.h"

#include "GeneralStatus\_ExPCL.h"

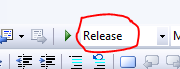
#include "MagneticCardData\_ExPCL.h"

#include "PrinterState\_ExPCL.h"

#include "PrinterData\_ExPCL.h"

#include "VersionInformation\_ExPCL.h"

1. When building your project, make sure to build under **Release** configuration to prevent any conflicts.



# DATAMAX-O’NEIL SDK API GUIDE

(Note: For Full API Documentation, please view the index.html file in Documentation folder of the SDK package):

Windows SDKs: **Documentation folder -> html folder -> index.html**

iOS SDK: **SDK folder -> doc -> html -> index.html**

Android/Java SDK: **doc -> index.html**

## Establishing connection

Datamax-O’Neil SDK supports several types of connection including Bluetooth, TCP/IP, and Serial depending on the platform you use. Below are examples of how to establish connection with Datamax-O’Neil printers.

### Datamax-O’Neil Android/Java SDK

//====FOR BLUETOOTH CONNECTIONS========//

ConnectionBase conn = Connection\_Bluetooth.*createClient*(“00:17:AC:01:51:B9”);

conn.open();

//====FOR TCP/IP CONNECTIONS========//

ConnectionBase conn = Connection\_TCP.*createClient*(“192.168.101.105”, 515);

conn.open();

### Datamax-O’Neil iOS SDK

//====FOR BLUETOOTH CONNECTIONS========//

accessoryArray = [[NSMutableArray alloc]init];

// load our data from a plist file inside our app bundle

NSArray \*supportedProtocols = [[NSArray alloc] initWithArray:[[NSBundle mainBundle] objectForInfoDictionaryKey:@"UISupportedExternalAccessoryProtocols"]];

//Get supported protocols from application bundle

NSArray \*supportedProtocols = [[NSArray alloc] initWithArray:[[NSBundle mainBundle]objectForInfoDictionaryKey:@"UISupportedExternalAccessoryProtocols"]];

//Create client and obtain external accessory list(connected devices)

ConnectionBase\* conn = [[Connection\_BluetoothEA alloc] createClient];

NSArray \*accessoryList = [[NSArray alloc] initWithArray:[(Connection\_BluetoothEA \*)conn accessoryList]];

//Populate list with connected Bluetooth printers

if ([accessoryList count] > 0)

{

// Loop through the connected device list

for (EAAccessory \*accessory in accessoryList) {

//For each protocol name of the accessory

for (NSString \*protocolName in [accessory protocolStrings])

{

//for each supported protocol

for (NSString \*supportedProtocol in supportedProtocols)

{

//if the protocol name is supported

if ([protocolName isEqualToString:supportedProtocol])

{

//add name of the device

[deviceActionSheet addButtonWithTitle:[NSString stringWithFormat:@"%@ (%@)",[accessory name],[accessory modelNumber]]];

[accessoryArray addObject:accessory];

}

}//end supportedprotocol loop

} //end protocal name for loop

}//end accessory for loop

}//end if

selectedAccessory = [accessoryArray objectAtIndex:buttonIndex];

//setup connection for selected bluetooth printer

[(Connection\_BluetoothEA \*)conn setupConnectionForAccessory:selectedAccessory withProtocolString:[supportedProtocols objectAtIndex:0]];

[conn open];

//====FOR TCP/IP CONNECTIONS========//

ConnectionBase\* conn = [[Connection\_TCP alloc] createClient:@”192.168.101.105” targetPort:515];

conn.connTimeout = 5;

[conn open];

You will need to implement three delegate functions in order to handle when connection open, error, and close. (Below is taken from DOPrint Sample):

#pragma mark - ConnectionBase Delegates

-(void)connectionDidOpen:(id)connection {

//Selected Print

if(self.actionMode.selectedSegmentIndex == 0) {

//------------------------------------------------------------------

//Send data to printer via background thread

//------------------------------------------------------------------

backgroundThread = [[NSThread alloc]initWithTarget:self selector:@selector(threadPerformPrint) object:nil];

backgroundThread.name = @"Print";

[backgroundThread start];

}

//Selected Query

else {

//------------------------------------------------------------------

//Send data to printer via background thread

//------------------------------------------------------------------

backgroundThread = [[NSThread alloc]initWithTarget:self selector:@selector(threadPerformQuery) object:nil];

backgroundThread.name = @"Query";

[backgroundThread start];

}

}

-(void)connectionFailed:(id)connection withError:(NSError \*)error {

}

-(void)connectionDidClosed:(id)connection {

//Get connection status id

int connStatus = [conn getConnectionStatusID];

if(setAbort) {

[self updateStatusWithText:@"Connection attempt aborted."];

setAbort = false;

}

//Stream close successfully

if (connStatus == NSStreamStatusClosed)

{

[self updateStatusWithText:@"Connection closed."];

}//end else

//Connection was not able to establish

else if (connStatus == NSStreamStatusNotOpen)

{

[self updateStatusWithText:@"Connection was unable to establish or is not opened."];

}

//We failed to open connection after 5 seconds

else if(connStatus == NSStreamStatusOpening)

{

[self updateStatusWithText:[NSString stringWithFormat:@"Connection timed out after %d seconds.", conn.connTimeout]];

}

//Error in stream has occurred

else if (connStatus == NSStreamStatusError)

{

//Get last stream error that occurred

NSError \*streamError = [conn getLastStreamError];

//Update status

[self updateStatusWithText:[NSString stringWithFormat:@"Error: %@", (streamError != nil)?[streamError localizedDescription] : nil]];

}

//Update UI

[self enableControls:true];

conn = nil;

}

### Datamax-O’Neil .NET C# SDKs

//====FOR BLUETOOTH CONNECTIONS========//

ConnectionBase conn = (Connection\_Bluetooth32Feet)Connection\_Bluetooth32Feet.CreateClient(“00:17:AC:01:51:B9”,“0000”);

conn.Open();

//====FOR TCP/IP CONNECTIONS========//

ConnectionBase conn = (Connection\_TCP)Connection\_TCP.CreateClient(“192.168.101.105”, 515);

conn.Open();

### Datamax-O’Neil .NET C# SDK for Windows Store (Windows 8/8.1)

//====FOR BLUETOOTH CONNECTIONS========//

//NOTE: PRINTER MUST BE PAIRED WITH DEVICE or PC BEFOREHAND.

ConnectionBase conn = (Connection\_Bluetooth)Connection\_Bluetooth.CreateClient(“00:17:AC:01:51:B9”);

**//THIS IS IMPORTANT AND MUST BE CALLED IN THE UI or MAIN THREAD. Connection must be initialized before opening connection. A message will pop up asking you to allow or block device.**

conn.Init();

conn.Open();

//====FOR TCP/IP CONNECTIONS========//

ConnectionBase conn = (Connection\_TCP)Connection\_TCP.CreateClient(“192.168.101.105”, 515);

conn.Init();

conn.Open();

### Datamax-O’Neil C++ SDKs

//====FOR BLUETOOTH CONNECTIONS========//

ConnectionBase\* conn = (Connection\_BluetoothMsftStack\*)Connection\_BluetoothMsftStack::CreateClient(“00:17:AC:01:51:B9);

conn.Open();

//====FOR TCP/IP CONNECTIONS========//

ConnectionBase\* conn = (Connection\_TCP\*)Connection\_TCP::CreateClient(“192.168.101.105”, 515);

conn.Open();

## Creating a Print Job for Printing

There are two classes you must use to form a print job: **Document** class and **Parameters** class. **Document** class is the class to create a print job while the **Parameters** class allows you to modify print job with additional settings such as rotation, horizontal/vertical multiplier, and etc. Below are examples of creating print jobs for each type of Datamax-O’Neil printers.

### Datamax-O’Neil Android/Java SDK

//====DPL Printers(eg. RL3, RL4, etc.)========//

DocumentDPL docDPL = **new** DocumentDPL();

ParametersDPL paramDPL = **new** ParametersDPL();

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

DocumentEZ docEZ = **new** DocumentEZ("MF204"); //EZ mode. MF204 is the font name

ParametersEZ paramEZ = **new** ParametersEZ();

DocumentEZ docLP = **new** DocumentLP("!"); //LinePrint mode.“!” is the font name

ParametersLP paramLP = **new** ParametersLP();

//====Apex Printers(Apex 2, Apex 3, etc..)========//

DocumentExPCL\_LP docExPCL\_LP = **new** DocumentExPCL\_LP(3); //Line Print mode. “3” is the font index.

ParametersExPCL\_LP paramExPCL\_LP = **new** ParametersExPCL\_LP();

DocumentExPCL\_PP docExPCL\_PP = **new** DocumentExPCL\_PP(PaperWidth.*PaperWidth\_384*); //Page print mode

ParametersExPCL\_PP paramExPCL\_PP = **new** ParametersExPCL\_PP();

### Datamax-O’Neil iOS SDK

//====DPL Printers(eg. RL3, RL4, etc.)========//

DocumentDPL\* documentDPL = [[DocumentDPL alloc] init];

ParametersDPL\* param = [[ParametersDPL alloc] init];

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

DocumentEZ\* documentEZ = [[DocumentEZ alloc] initWithFiveCharacterFont: @"MF204"];

ParametersEZ\* paramEZ = [[ParametersEZ alloc] init];

DocumentLP\* documentLP = [[DocumentLP alloc] initWithOneCharacterFont:’!’];

ParametersLP\* param = [[ParametersLP alloc] init];

//====Apex Printers(Apex 2, Apex 3, etc..)========//

DocumentExPCL\_LP\* documentExPCL\_LP = [[DocumentExPCL\_LP alloc] initWithFont:3]; //Line Print mode. “3” is the font index.

ParametersExPCL\_LP\* paramExPCL\_LP = [[ParametersExPCL\_LP alloc] init];

DocumentExPCL\_PP\* documentExPCL\_PP = [[DocumentExPCL\_PP alloc] initWithPaperWidth:PaperWidth\_384]; //Page Print mode.

ParametersExPCL\_PP\* paramExPCL\_PP = [[ParametersExPCL\_PP alloc] init];

### Datamax-O’Neil .NET C# SDKs

//====DPL Printers(eg. RL3, RL4, etc.)========//

DocumentDPL docDPL = new DocumentDPL();

ParametersDPL paramDPL = new ParametersDPL();

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

DocumentEZ docEZ = new DocumentEZ("MF204");//EZ mode. MF204 is the font name

ParametersEZ paramEZ = new ParametersEZ();

DocumentLP docLP = new DocumentLP("!"); //LinePrint mode.“!” is the font name

ParametersLP paramLP = new ParametersLP();

//====Apex Printers(Apex 2, Apex 3, etc..)========//

DocumentExPCL\_LP docExPCL\_LP = new DocumentExPCL\_LP(3); //Line Print mode. “3” is the font index.

ParametersExPCL\_LP paramExPCL\_LP = new ParametersExPCL\_LP();

DocumentExPCL\_PP docExPCL\_PP = new DocumentExPCL\_PP(DocumentExPCL\_PP.PaperWidthValue.PaperWidth\_384); //Page print mode

ParametersExPCL\_PP paramExPCL\_PP = new ParametersExPCL\_PP();

### Datamax-O’Neil .NET C# SDKs for Windows Store (Windows 8/8.1)

//====DPL Printers(eg. RL3, RL4, etc.)========//

DocumentDPL docDPL = new DocumentDPL();

ParametersDPL paramDPL = new ParametersDPL();

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

DocumentEZ docEZ = new DocumentEZ("MF204"); //EZ mode. MF204 is the font name

ParametersEZ paramEZ = new ParametersEZ();

DocumentLP docLP = new DocumentLP("!"); //LinePrint mode “!” is the font name

ParametersLP paramLP = new ParametersLP();

//====Apex Printers(Apex 2, Apex 3, etc..)========//

DocumentExPCL\_LP docExPCL\_LP = new DocumentExPCL\_LP(3); //Line Print mode. “3” is the font index.

ParametersExPCL\_LP paramExPCL\_LP = new ParametersExPCL\_LP();

DocumentExPCL\_PP docExPCL\_PP = new DocumentExPCL\_PP(DocumentExPCL\_PP.PaperWidthValue.PaperWidth\_384); //Page print mode

ParametersExPCL\_PP paramExPCL\_PP = new ParametersExPCL\_PP();

### Datamax-O’Neil C++ SDKs

//====DPL Printers(eg. RL3, RL4, etc.)========//

DocumentDPL\* docDPL = new DocumentDPL();

ParametersDPL paramDPL;

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

DocumentEZ \*docEZ = new DocumentEZ("MF204");//EZ mode. MF204 is the font name

ParametersEZ paramEZ;

DocumentLP \*docLP = new DocumentLP("!");//LP mode. “1” is the font name

ParametersLP paramLP;

//====Apex Printers(Apex 2, Apex 3, etc..)========//

DocumentExPCL\_LP \*docExPCL\_LP = new DocumentExPCL\_LP(3);//Line Print mode. “3” is the the font index.

ParametersExPCL\_LP paramExPCL\_LP;

DocumentExPCL\_PP \*docExPCL\_PP = new DocumentExPCL\_PP(PaperWidth::PaperWidth\_384);//Page print mode

ParametersExPCL\_PP paramExPCL\_PP;

## Printing

Below are examples of printing text to printer.

### Datamax-O’Neil Android/Java SDK

//====DPL Printers(eg. RL3, RL4, etc.)========//

docDPL.writeTextInternalBitmapped("Hello World",1,5,5);

//write normal ASCII Text Scalable

docDPL.writeTextScalable("Hello World","00",25,5);

conn.write(docDPL.getDocumentData());

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

//EZ Mode

docEZ.writeText("Customer Code: 00146",50,1);

conn.write(docEZ.getDocumentData());

//LP Mode

docLP.writeText("Customer Code: 00146");

conn.write(docLP.getDocumentData());

//====Apex Printers(Apex 2, Apex 3, etc..)========//

//Line Print Mode

paramExPCL\_LP.setFontIndex(5);

docExPCL\_LP.writeText("Hello World I am a printing sample (Font - K5)",paramExPCL\_LP);

conn.write(docExPCL\_LP.getDocumentData());

//Page Print Mode

docExPCL\_PP.drawText(0,1600,**true**,RotationAngle.*RotationAngle\_0*,"<f=1>This is a sample");

conn.write(docExPCL\_PP.getDocumentData());

### Datamax-O’Neil iOS SDK

//====DPL Printers(eg. RL3, RL4, etc.)========//

[documentDPL writeTextInternalBitmapped:@"Hello World" withFontID:1 atRow:5 atColumn:5];

//write normal ASCII Text Scalable

[documentDPL writeTextScalable:@"Hello World" withFontID:@"00" atRow:25 atColumn:5];

[conn writeData: [documentDPL getDocumentData]];

[documentDPL release];

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

//EZ Mode

[documentEZ writeText: @"Customer Code: 00146" atRow:50 atColumn:1 withParameters:paramEZ];

[conn writeData: [documentEZ getDocumentData]];

[documentEZ release];

//LP Mode

[documentLP writeText:@"Customer Code: 00146"];

[conn writeData: [documentLP getDocumentData]];

[documentLP release];

//====Apex Printers(Apex 2, Apex 3, etc..)========//

//Line Print Mode

paramExPCL\_LP.fontIndex = 5;

[documentExPCL\_LP writeText: @"Hello World I am a printing sample (Font - K5)" withParameters:paramExPCL\_LP];

[conn writeData: [documentExPCL\_LP getDocumentData]];

[documentExPCL\_LP release];

//Page Print Mode

[documentExPCL\_PP drawTextX:0 Y:1600 color:1 angle:0 string:@"<f=1>This is a sample"];

[conn writeData: [documentExPCL\_PP getDocumentData]];

[documentExPCL\_PP release];

### Datamax-O’Neil .NET C# SDKs

//====DPL Printers(eg. RL3, RL4, etc.)========//

docDPL.WriteTextInternalBitmapped("Hello World", 1, 5, 5);

//write normal ASCII Text Scalable

docDPL.WriteTextScalable("Hello World", "00", 25, 5);

conn.Write(docDPL.GetDocumentData());

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

//EZ Mode

docEZ.WriteText("Customer Code: 00146", 50, 1);

conn.Write(docEZ.getDocumentData());

//LP Mode

docLP.WriteText("Customer Code: 00146");

conn.Write(docLP.GetDocumentData());

//====Apex Printers(Apex 2, Apex 3, etc..)========//

//Line Print Mode

paramExPCL\_LP.FontIndex = 5;

docExPCL\_LP.WriteText("Hello World I am a printing sample (Font - K5)", paramExPCL\_LP);

conn.Write(docExPCL\_LP.GetDocumentData());

//Page Print Mode

docExPCL\_PP.DrawText(0, 1600, true, (ParametersExPCL\_PP.RotationAngle)0, "<f=1>This is a sample");

conn.Write(docExPCL\_PP.GetDocumentData());

### Datamax-O’Neil .NET C# SDKs for Windows Store (Windows 8/8.1)

//====DPL Printers(eg. RL3, RL4, etc.)========//

docDPL.WriteTextInternalBitmapped("Hello World", 1, 5, 5);

//write normal ASCII Text Scalable

docDPL.WriteTextScalable("Hello World", "00", 25, 5);

conn.Write(docDPL.GetDocumentData());

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

//EZ Mode

docEZ.WriteText("Customer Code: 00146", 50, 1);

conn.Write(docEZ.getDocumentData());

//LP Mode

docLP.WriteText("Customer Code: 00146");

conn.Write(docLP.GetDocumentData());

//====Apex Printers(Apex 2, Apex 3, etc..)========//

//Line Print Mode

paramExPCL\_LP.FontIndex = 5;

docExPCL\_LP.WriteText("Hello World I am a printing sample (Font - K5)", paramExPCL\_LP);

conn.Write(docExPCL\_LP.GetDocumentData());

//Page Print Mode

docExPCL\_PP.DrawText(0, 1600, true, (ParametersExPCL\_PP.RotationAngle)0, "<f=1>This is a sample");

conn.Write(docExPCL\_PP.GetDocumentData());

### Datamax-O’Neil C++ SDKs

//====DPL Printers(eg. RL3, RL4, etc.)========//

docDPL->WriteTextInternalBitmapped("Hello World",1,5,5);

//write normal ASCII Text Scalable

docDPL->WriteTextScalable("Hello World","00",25,5);

conn->Write(docDPL->GetDocumentData());

delete docDPL;

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

//EZ Mode

docEZ->WriteText("Customer Code: 00146",50,1);

conn->Write(docEZ->GetDocumentData());

delete docEZ;

//LP Mode

docLP->WriteText("Customer Code: 00146");

conn->Write(docLP->GetDocumentData());

delete docLP;

//====Apex Printers(Apex 2, Apex 3, etc..)========//

//Line Print Mode

paramExPCL\_LP.SetFontIndex(5);

docExPCL\_LP->WriteText("Hello World I am a printing sample (Font - K5)",paramExPCL\_LP);

conn->Write(docExPCL\_LP->GetDocumentData());

delete docExPCL\_LP;

//Page Print Mode

docExPCL\_PP->DrawTextPP(0,1600,1,(RotationAngle::Values)0,"<f=1>This is a sample");

conn->Write(docExPCL\_PP->GetDocumentData());

delete docExPCL\_PP;

## Querying

Below are examples of querying printer’s properties**. Note: The examples below will query printer’s general information (PrinterInformation\_DPL/GeneralStatus class). If you want to query other information(Bluetooth information, network config, etc..), please see the documentation to see what classes are available**

### Datamax-O’Neil Android/Java SDK

//====DPL Printers(eg. RL3, RL4, etc.)========//

//Query Printer info

PrinterInformation\_DPL printerInfo = **new** PrinterInformation\_DPL();

printerInfo.queryPrinter(conn,1000);

String message = "";

**if** (printerInfo.getValid() == **false**)

{

message = "No response from printer\r\n";

}

**else**

{

message = String.*format*("Firmware Version: %s\n", printerInfo.getVersionInformation());

}

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

GeneralStatus genStatus = **new** GeneralStatus();

genStatus.queryPrinter(conn,1000);

String message = "";

**if** (genStatus.getValid() == **false**)

{

message = "No response from printer\r\n";

}

**else**

{

message = String.*format*("Battery Voltage: %d\n", genStatus.getBatteryTempandVoltageStatus().name());

}

//====Apex Printers(Apex 2, Apex 3, etc..)========//

GeneralStatus\_ExPCL generalStatus = **new** GeneralStatus\_ExPCL();

generalStatus.queryPrinter(conn,1000);

String message = "";

**if** (generalStatus.getValid() == **false**)

{

message = "No response from printer\r\n";

}

**else**

{

message = String.*format*("Battery Voltage: %d\n", generalStatus.getBatteryVoltage());

}

### Datamax-O’Neil iOS SDK

//====DPL Printers(eg. RL3, RL4, etc.)========//

//Query Printer info

PrinterInformation\_DPL\* printerInfo = [[[PrinterInformation\_DPL alloc]init]autorelease];

[printerInfo queryPrinter:conn withTimeout:1000];

//Query message

NSMutableString\* message = [NSMutableString stringWithFormat:@""];

**if** ([printerInfo getValid] == false)

{

[message appendString:@"No response from printer\r\n"];

}

**else**

{

[message appendFormat:@"Firmware Version: %@\n", [printerInfo getVersionInformation]];

}

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

GeneralStatus\* genStatus = [[[GeneralStatus alloc] init] autorelease];

[genStatus queryPrinter:conn withTimeout:1000];

//Query message

NSMutableString\* message = [NSMutableString stringWithFormat:@""];

**if** ([genStatus getValid] == false)

{

[message appendString:@"No response from printer\r\n"];

}

**else**

{

[message appendFormat:@"Battery Temp and Voltage Status: %c\n", [genStatus getBatteryTempandVoltageStatus]];

}

//====Apex Printers(Apex 2, Apex 3, etc..)========//

GeneralStatus\_ExPCL \*generalStatus = [[[GeneralStatus\_ExPCL alloc] init] autorelease];

[generalStatus queryPrinter:conn withTimeout:1000];

//Query message

NSMutableString\* message = [NSMutableString stringWithFormat:@""];

**if** ([generalStatus getValid] == false)

{

[message appendString:@"No response from printer\r\n"];

}

**else**

{

[message appendFormat:@"Battery Voltage: %ld\n", [generalStatus getBatteryVoltage]];

}

### Datamax-O’Neil .NET C# SDKs

//====DPL Printers(eg. RL3, RL4, etc.)========//

//Query Printer info

PrinterInformation\_DPL printerInfo = new PrinterInformation\_DPL();

printerInfo.QueryPrinter(conn, 3000);

String message = "";

**if** (printerInfo.Valid == **false**)

{

message = "No response from printer\r\n";

}

**else**

{

message = String.*Format*("Firmware Version: {0}\n", printerInfo.VersionInformation);

}

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

GeneralStatus genStatus = new GeneralStatus();

genStatus.QueryPrinter(conn, 3000);

String message = "";

**if** (genStatus.Valid == **false**)

{

message = "No response from printer\r\n";

}

**else**

{

message = String.*Format*("Battery Voltage: {0}\n",genStatus.BatteryTempandVoltageStatus);

}

//====Apex Printers(Apex 2, Apex 3, etc..)========//

GeneralStatus\_ExPCL generalStatus = new GeneralStatus\_ExPCL();

generalStatus.QueryPrinter(conn, 3000);

String message = "";

**if** (generalStatus.Valid == **false**)

{

message = "No response from printer\r\n";

}

**else**

{

message = String.*Format*("Battery Voltage: {0}\n", generalStatus.BatteryVoltage);

}

### Datamax-O’Neil .NET C# SDKs for Windows Store (Windows 8/8.1)

//====DPL Printers(eg. RL3, RL4, etc.)========//

//Query Printer info

PrinterInformation\_DPL printerInfo = new PrinterInformation\_DPL();

printerInfo.QueryPrinter(conn, 3000);

String message = "";

**if** (printerInfo.Valid == **false**)

{

message = "No response from printer\r\n";

}

**else**

{

message = String.*Format*("Firmware Version: {0}\n", printerInfo.VersionInformation);

}

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

GeneralStatus genStatus = new GeneralStatus();

genStatus.QueryPrinter(conn, 3000);

String message = "";

**if** (genStatus.Valid == **false**)

{

message = "No response from printer\r\n";

}

**else**

{

message = String.*Format*("Battery Voltage: {0}\n",genStatus.BatteryTempandVoltageStatus);

}

//====Apex Printers(Apex 2, Apex 3, etc..)========//

GeneralStatus\_ExPCL generalStatus = new GeneralStatus\_ExPCL();

generalStatus.QueryPrinter(conn, 3000);

String message = "";

**if** (generalStatus.Valid == **false**)

{

message = "No response from printer\r\n";

}

**else**

{

message = String.*Format*("Battery Voltage: {0}\n", generalStatus.BatteryVoltage);

}

### Datamax-O’Neil C++ SDKs

//====DPL Printers(eg. RL3, RL4, etc.)========//

//Query Printer info

PrinterInformation\_DPL \*printerInfo = new PrinterInformation\_DPL();

printerInfo->QueryPrinter(conn,3000);

CString message(L"");

**if** (printerInfo->GetValid() == **false**)

{

message += L"No response from printer\r\n";

}

**else**

{

message.AppendFormat(L"Firmware Version: %s\r\n", A2BSTR(printerInfo->GetVersionInformation().c\_str()));

}

delete printerInfo;

//====Legacy Printers (OC2, OC3, MF4Te, etc.)========//

GeneralStatus \*genStatus = new GeneralStatus();

genStatus->QueryPrinter(conn,3000);

CString message(L"");

**if** (genStatus->GetValid() == **false**)

{

message += L"No response from printer\r\n";

}

**else**

{

message.AppendFormat(L"Battery Temp and Voltage Status: %u\r\n", genStatus->GetBatteryTempandVoltageStatus());

}

delete genStatus;

//====Apex Printers(Apex 2, Apex 3, etc..)========//

GeneralStatus\_ExPCL \*generalStatus = new GeneralStatus\_ExPCL();

generalStatus->QueryPrinter(conn,4000);

CString message(L"");

**if** (generalStatus.GetValid() == **false**)

{

message += L"No response from printer\r\n";

}

**else**

{

message.AppendFormat(L"Battery Voltage: %ld\r\n", generalStatus->GetBatteryVoltage());

}

delete generalStatus;

## Terminating Connection

To terminate connection, use the ConnectionBase.Close() method or setting the IsClosing property to true. Below are examples.

### Datamax-O’Neil Android/Java SDK

//====Method 1========//

conn.close();

//====Method 2========//

conn.setIsClosing(true);

### Datamax-O’Neil iOS SDK

//====Method 1========//

[conn close];

[conn release];

conn = nil;

### Datamax-O’Neil .NET C# SDKs

//====Method 1========//

conn.Close();

//====Method 2========//

conn.IsClosing = true;

### Datamax-O’Neil .NET C# SDKs for Windows 8/8.1

//====Method 1========//

conn.Close();

### Datamax-O’Neil C++ SDKs

//====Method 1========//

conn->Close();

//====Method 2========//

conn->SetIsClosing(true);

delete generalStatus;