

Instructions

- To Install the SDK using Android Studio
 - Copy the .aar file into the libs folder of your app, right click it, and copy “Path From Content Root” (you may have to create the libs folder under “app” folder).
 - Navigate to File > Project Structure > Dependencies and tap the “+” to add a new .aar dependency
 - Paste the file path into the popup window
 - Rebuild app
- In app/build.gradle user must have:
 - minSdkVersion 23 or above
 - Implementation ‘androidx.appcompat:appcompat:1.4.2’
 - Implementation ‘com.google.android.gms:play-services-tasks:18.0.2’
 - implementation group: 'org.javatuples', name: 'javatuples', version: '1.2’
 - If users device is Android 6 (API 23):
 - Add dependency: coreLibraryDesugaring
‘com.android.tools.desugar_jdk_libs:1.1.5’
 - Add ‘coreLibraryDesugaringEnabled = true’ to compileOptions
- To open a template:
 - Create an InputStream from the selected .bwt file
 - Pass the inputStream to the getTemplate method
 - Template template = TemplateFactory.getTemplate(inputStream);
- To apply data to template objects:
 - Create a list to hold the template objects
 - TemplateObjectData[] objectsList = template.getTemplateData();
 - Filter/search for the appropriate object using object.getName()
 - Apply desired value using object.setValue(“desired value”)
- To get a bitmap preview of a label:
 - Call template.getPreview(labelNumber, dpi, maxPixelWidthAndHeight) where:
 - labelNumber is the 0-based index of the label (0 unless the using a multi label template)
 - Dpi = dots per inch (usually 96)
 - maxPixelWidthAndHeight = desired size
- To discover printers:
 - Make sure to prompt the user to enable the following permissions (these are included in the SDK’s AndroidManifest so adding them in your app’s manifest is redundant).
 - If your device is **higher** than API level 31 call:
 - ActivityCompat.requestPermissions(this, new String[]{
Manifest.permission.BLUETOOTH_CONNECT,
Manifest.permission.BLUETOOTH_SCAN,

```
Manifest.permission.ACCESS_FINE_LOCATION,  
Manifest.permission.BLUETOOTH}, 0);
```

- If your device is **lower** than API level 31 call:
 - `ActivityCompat.requestPermissions(this, new String[]{
Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.BLUETOOTH}, 0);`
- Create new `PrinterDiscoveryImpl` which takes the app context
 - `PrinterDiscoveryImpl printerDiscovery = new
PrinterDiscoveryImpl(getApplicationContext());`
- Override system methods
 - From the activity that will show the discovered printers, override the `onResume()`, `onPause()`, and `onDestroy()` methods and call the new `discoveryImpl` in each of them respectively.
 - Example: in `onResume()` you will call `discoveryImpl.onResume();`
- Start discovery
 - For **Wifi (M611)** call:
 - `printerDiscovery.startWifiPrinterDiscovery();`
 - For **Bluetooth Low Energy (M211)** call:
 - `printerDiscovery.startBlePrinterDiscovery();`
 - For **Bluetooth (M611)** call:
 - `printerDiscovery.startBluetoothPrinterDiscovery(false);`
 - If the parameter is true, it will only discover printers already paired to the phone.
 - If false, it will return printers paired to the phone and search nearby for printers. (if a printer is both already paired and turned on nearby, it will be found twice)
 - `printerDiscovery.getDiscoveredPrinterNames()` returns the names of all discovered printers at the time of the call.
 - `printerDiscovery.stopPrinterDiscovery()` stops all discovery processes.
- To connect to a printer:
 - Implement `PrinterUpdateListener`
 - Subscribes to printer changes if desired
 - Use a `DiscoveredPrinterInformation` (returned from the printer discovery process)
 - `PrinterDetails printerDetails =
PrinterConnectionFactory.connectToDiscoveredPrinter(context,
Name of desired printer as a string,
Listener context of the class that implements the update listener interface;`
 - You may check if you have ownership of the printer with `PrinterConnectionFactory.HaveOwnership();`
 - This takes time so it's recommended to show a spinner, etc

- To disconnect from a printer:
 - `printerDetails.disconnect();`

- To print:
 - Create and set `PrintingOptions`
 - `PrintingOptions printingOptions = new PrintingOptions();`
 - `printingOptions.setCutOption(CutOption.EndOfLabel);`
 - `printingOptions.setNumberOfCopies(1);`
 - Pass the template and options to the print method
 - `PrintingStatus printingStatus = printerDetails.print(template, printingOptions, DontPrintTrailerFlag (pass in null as default));`
 - Returns either `PrintingSucceeded` or `PrintingFailed`
 - This takes time so it's recommended to show a spinner, etc

- To automatically connect to a previously-connected printer when the app starts:
 - After initializing `PrinterDiscoveryImpl`, call `printerDiscovery.getLastConnectedPrinterName();`
 - If this isn't null or an empty string, pass this string into `printerDiscovery.connectToDiscoveredPrinter` instead of starting discovery