Instructions

- To Install the SDK using Android Studio
    - Copy the .aar file into the libs folder of your app and copy the file path to this location
    - 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 ‘org.javatuples.javatuples:1.2’
    - If user’s 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 a .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
    - Manage Permissions
        - Add the following to the AndroidManifest (for wifi)
            - <uses-permission android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE" />

            - <uses-permission android:name="android.permission.INTERNET" />
            - <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
        - Add the following to the AndroidManifest (for bluetooth)
            - <uses-permission android:name="android.permission.BLUETOOTH" />
            - <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
            - <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

        - Prompt users to grant the necessary permissions
            - ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 0);
        - Highly recommended checking that required permissions are enabled
- Implement PrinterDiscoveryListener
    - public void printerDiscovered(DiscoveredPrinterInformation discoveredPrinterInformation) is called when a printer is found
    - This information is used to connect to a printer.
    - public void printerDiscoveryStarted and public void printerDiscoveryStopped in case this information is needed
- 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, call printerDiscovery.startWifiPrinterDiscovery();
    - For Bluetooth Low Energy, call printerDiscovery.startBlePrinterDiscovery()
    - For Bluetooth, 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)
    - You must not call a “startDiscovery” method in its own thread since the scan occurs in the background until printerDiscovery.stopPrinterDiscovery() is called.

- 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, discoveredPrinterInformation, printerUpdateListeners);
    - 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