Honeywell Mobility Scanning SDK for Xamarin V1.34

NEW IN THIS RELEASE

- BarcodeReader method enhanced to default to internal scanner in case of Honeywell handheld computers and external scanner connected to COM1 port in case of Honeywell THOR VM1A.
- Fixed Signing related Security issues.
- CT40, CT60, CN80, CN85 and VM1A Android Oreo support

VERSION HISTORY

Version	Description
1.34	• BarcodeReader method enhanced to default to internal scanner in case of Honeywell handheld computers and external scanner connected to COM1 port in case of Honeywell THOR VM1A. • Fixed Signing related Security Issues. • CT40,CT60,CN80,CN85 And VM1A Android Oreo Support.
1.32	• Fixed the BarcodeReader constructor hanging issue introduced in the v1.31 release. This issue was manifested when the constructor was called without calling the BarcodeReader.GetConnectedBarcodeReaders method first. • Added a new scanner setting for specifying a data editing plug-in to modify the barcode data before it is delivered in the BarcodeDataReady event.
1.31	• Added a new method to enable or disable scanning. • Added a new event for receiving notification when the reader's connection state changes. • Added new scanner and symbology setting properties in the BarcodeReaderSettingKeys class: o Code128ShortMargin o Code93HighDensity o DataProcessorCharset o DataProcessorLaunchBrowser o DataProcessorLaunchEZConfig o DataProcessorPrefix o DataProcessorScanToIntent o DataProcessorSuffix o DataProcessorSymbologyPrefix o DotCodeEnabled o DotCodeMaximumLength o DotCodeMinimumLength o GridMatrixEnabled o GridMatrixMaximumLength o GridMatrixMinimumLength o MsiOutOfSpecSymbol o MsiShortMargin o TriggerScanDelay o TriggerScanMode o TriggerScanSameSymbolTimeout o TriggerScanSameSymbolTimeoutEnabled

	○ TriggerTimeout Note: These settings depend on the support of the Data Collection Service (or Scanning on newer computers). • Added new setting value properties in the BarcodeReaderSettingValues class. • The BarcodeReader.GetConnectedBarcodeReaders method and the BarcodeReader constructors no longer require the context parameter on the Android platform. It is default to Android.App.Application.Context. • Updated the BarcodeReaderSample application to demonstrate enabling/disabling scanning and trigger scan modes. Note: The logic of the Continuous switch in the BarcodeReaderSample application was updated to use the new TriggerScanMode setting to set the scan mode to continuous mode. In this mode, the reader continuously decodes unique barcodes when the scan trigger is pressed until the trigger is released. The scan mode applies to both hardware and software trigger. • Fixed the BarcodeReaderSample solution to build Debug in Debug configuration.
1.20	• Initial release • Supports Honeywell Android mobile computers. • Gets a list of connected barcode readers. • Opens or closes a connection to an internal barcode reader or a supported external barcode reader (such as the ring scanner for Dolphin 75e). • Receives scanned barcode data via events. • Programmatically triggers the scanner. • Configures the symbology and decoder settings.

SUPPORTED MOBILE COMPUTERS

- CN51 Android 6.0
- CN75, CN75e, CK75 Android 6.0
- Dolphin CN80 Android 7.1.1,8.1.0
- Dolphin CT50 Android 4.4 and 6.0
- Dolphin CT60 Android 7.1.1,8.1.0
- Dolphin CT40 Android 8.1.0
- Dolphin CN85 Android 8.1.0
- Dolphin 75e Android 4.4 and 6.0
- EDA50, EDA50K, EDA70 Android 7.1.1
- THOR VM1A Android 8.1.0

REQUIREMENTS

Android Platform

- Honeywell Data Collection Service (which is called Scanning on newer computers) is required to support the scanning API. Data Collection Service is preinstalled on the Honeywell Android computers. Note: This SDK does not support CN51 Android 4.x because it has an older and incompatible Data Collection Service.

KNOWN ISSUES

None

ANDROID APPLICATION TIPS

- If you use Scanning SDK versions prior to v1.31, the following line should be added to the AndroidManifest.xml file:

  ```
  <uses-permission android:name="com.honeywell.decode.permission.DECODE" />
  ```

  If you use Visual Studio 2015 or 2017 for development, the AndroidManifest.xml file is located in the Properties subfolder of the Xamarin.Android project you created.
- The common practice on the Android platform is to open the scanner when the scanning activity is about to be displayed, and close the scanner when the scanning activity is about to be hidden. This allows other applications that rely on the scan wedge to be able to receive barcode data as keyboard input.
- If the application plans to work with multiple scanners, it is best to create one BarcodeReader object per scanner and open each scanner respectively. The application may use the same BarcodeDataReady event handler to handle the data from multiple scanners.
- Before the application opens the scanner via the BarcodeReader class, it may receive barcode data via the scan wedge as keyboard input. The wedge data would appear in the current focused text input field. You may add additional UI logic to either disable certain UI controls until the scanner is opened or set default focus to non-text fields.