Overview

Zebra RFID SDK for Windows and Windows Mobile provides a .NET application programming interface for the Zebra RFD8500 reader.

OS Compatibility

- Windows 7 (and above) with .NET framework 4.5
- Windows Mobile with Compact Framework 3.5.
Mobile Device Compatibility

- Zebra MC55
- Zebra MC67

Device Compatibility

- RFD8500

Version History

Version 2.1 – 11/2018

1. Added Access Operations – Tag Read, Tag Write, Tag Lock and Tag Kill
2. Configuration Updates
   a. Antenna – Link Profile, Tari
   b. Singulation – Inventory State, SL Flag
   c. Tag Reporting – PC, RSSI, Phase, Channel Index, Tag Seen Count and Report Unique Tags
   d. Regulatory - Channel Index
   e. Power Management – Dynamic Power
3. Added Pre-Filters
4. Added Tag Locate

Version 2.0 – 06/2018

1. Windows Desktop new SDK.

Version 1.2 – 02/2018

1. Batch mode support added.
2. Added serial port number into RFID Device properties.
3. Removed serial port communication.
4. Stability improvements.
5. Removed serial port number (COMX) from RFID Device properties.
6. Connection logic has been changed when RFID serial port get non-maximum serial port id.
7. Operation summary notification events.
8. ERROR_INSUFFICIENT_POWER exception introduced.
9. More informative exception messages.
10. Diagnostic logs.
11. Battery, Power and Temperature notifications.
12. Configure Beeper volume.

Version 1.1 – 07/2017

1. See reader capabilities.
2. PeakRSSI and TagSeenCount in TagData.
3. Configure Start/Stop triggers.
4. Get status events when inventory starts and stops.
5. Get an array of Tags that was read by the reader when data is not attached to ReadNotify event.
6. Purge all tags present in Dll and Reader queues when data is not attached to ReadNotify event.

Version 1.0 – 06/2017

1. Initial release of RFID SDK for Windows which has following capabilities
   • Getting list of available/active (connected) readers.
   ▪ New device pairing.
   ▪ See reader asset information like Bluetooth Name, Bluetooth address.
   ▪ Connecting and disconnecting to readers.
   ▪ Perform inventory and stop.
   ▪ Receive tag id through ReadNotify events.
   ▪ Control antenna power.
   ▪ Session and tag population control.
   ▪ Saving settings and setting reader into factory defaults.
   ▪ See the region of the reader.
   ▪ Get status events when scanner disconnects/out of range/battery drain.
   ▪ Get events when a new device pairs and un-pairs.
   ▪ Get version information.

Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Location</th>
<th>Path in the Zip package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows SDK Library</td>
<td>SDK API DLL and it’s dependencies</td>
<td>\Windows.Desktop.SDK</td>
</tr>
<tr>
<td>Windows Demo Application Binaries</td>
<td>Binaries of the Demo application</td>
<td>\Windows.Desktop.DemoApp</td>
</tr>
<tr>
<td>Windows Mobile SDK Library</td>
<td>SDK API DLL and it’s dependencies</td>
<td>\Windows.Mobile.SDK</td>
</tr>
<tr>
<td>Windows Mobile Demo Application Binaries</td>
<td>Binaries of the Demo application</td>
<td>\Windows.Mobile.DemoApp</td>
</tr>
</tbody>
</table>

©2018 ZIH Corp. All rights reserved. Zebra and the stylized Zebra head are trademarks of ZIH Corp., registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners.
Installation – Requirements

SDK will be distributed as a zip package that contains the components mentione in the components section of this document. Users may un-zip the package into their development environment and use SDK library in their projects.

Prerequisites

Microsoft Windows Mobile - .NET Compact Framework 3.5

Using Demo application

Note: This SDK has only been tested using the Microsoft Bluetooth Communication stack. Use of proprietary drivers using a different communication stack/layer may have undefined results.

1. The demo application exe (Symbol.RFID.SDK.DemoApp.exe) can be found inside “\Windows.Desktop.DemoApp” folder. Once you launch the application, if you have an already paired RFD8500, application will populate it and show in the combo box in following format <Bluetooth friendly name>
2. Bluetooth friendly names consist of device the family name and serial number which users can match with the label available on the device.
3. To pair a new device, the RFD8500 device should be discoverable, in range and the pairing request should be authenticated by pressing the yellow trigger of the device when device blink its blue LED faster. Please watch the video (https://www.youtube.com/watch?v=JxLkE7GVTrQ) for more information.
4. If the user restores the device to a factory defaults configuration, the device will disconnect. To connect the reader again, user may have to un-pair and re-pair the device.

Known Issues/Limitations

1. This SDK has only been tested only with Microsoft Bluetooth Driver stack and behavior with any other Bluetooth driver stacks is unknown.
2. New device pair notification (RFIDReaderAppeared) may not occur intermittently. Use the GetAvailableRFIDReaderList for most up-to-date reader list.
3. Session 0 and 1 inventory may reduce the performance as it takes time to process the data provide events. So, subsequent commands may fail (throw exceptions) after stopping the inventory.
4. TagTransitTime implementation is not complete in Antennas.SingulationControl.
5. The device disconnection status notification takes more than 20s to report the disconnection.
6. The SDK uses an emulated Bluetooth serial port to communicate with the device. For the SDK to function properly, the serial port should not be opened by any other application or service.
7. The RFD8500 automatically switches to a SNAPI USB communication mode when it is plugged in via the USB cable (disables Bluetooth). As such, it cannot be used with this SDK API while plugged-in.
8. GetReadTags and PurgeTags consider only the DLL queue. Batched data inside the reader is not considered.