



# AsReader Finger-Type

## C# SDK Reference Guide

ASR-023B

## Revision History

Ver.	Description	Date
V1.0	Initial version	2020/10/30
V1.1	Update the parameter of the function WhenReceivedBluetoothDevice from "BluetoothDevice device" to "List<BluetoothDevice> devices".	2020/12/7

# Contents

<b>Introduction .....</b>	<b>4</b>
<b>1. Import and Usage of SDK .....</b>	<b>5</b>
1.1. Import SDK .....	5
1.2. SDK Usage .....	7
1.2.1. Reference the Namespace .....	7
1.2.2. Search Devices .....	7
1.2.3. Stop Searching .....	8
1.2.4. Connect to the Device .....	8
1.2.5. Barcode Scan .....	8
1.2.6. Stop Scanning .....	9
1.2.7. Disconnection .....	9
<b>2. AsFinger Class .....</b>	<b>10</b>
2.1. Functions .....	10
2.1.1. Connect .....	10
2.1.2. Disconnect .....	10
2.1.3. StartScan .....	11
2.1.4. StopScan .....	12
2.1.5. GetBattery .....	12
2.1.6. GetFirmwareVersion .....	13
2.1.7. GetSDKVersion .....	13
2.1.8. GetBluetoothDevice .....	14
2.1.9. SendData .....	14
2.2. Delegates .....	15
2.2.1. WhenAsFingerIsConnected .....	15
2.2.2. WhenReceivedBarcodeData .....	16
2.2.3. WhenReceivedBattery .....	16

2.2.4. WhenReceivedFirmwareVersion .....	17
2.2.5. WhenReceivedData .....	18
<b>3. AsFingerManager Class .....</b>	<b>19</b>
3.1. Functions.....	19
3.1.1. StartSearching .....	19
3.1.2. StopSearching .....	19
3.1.3. GetSearchBluetoothType .....	20
3.2. Delegates .....	20
3.2.1. WhenReceivedBluetoothDevice .....	20
<b>4. BluetoothDevice Class .....</b>	<b>22</b>
4.1. Properties .....	22
<b>5. Types Class .....</b>	<b>23</b>
5.1. Enumeration Types .....	23
5.1.1. BluetoothType .....	23

## Introduction

### **Main purposes of this paper:**

- Guide developers to build the development environment so that developers can use the AsReader SDK library to develop Windows Desktop Applications.
- Explain the SDK library to the users.

### **Development tools:**

- Visual Studio 2019

### **System requirements:**

- Windows 10 1703+

# 1. Import and Usage of SDK

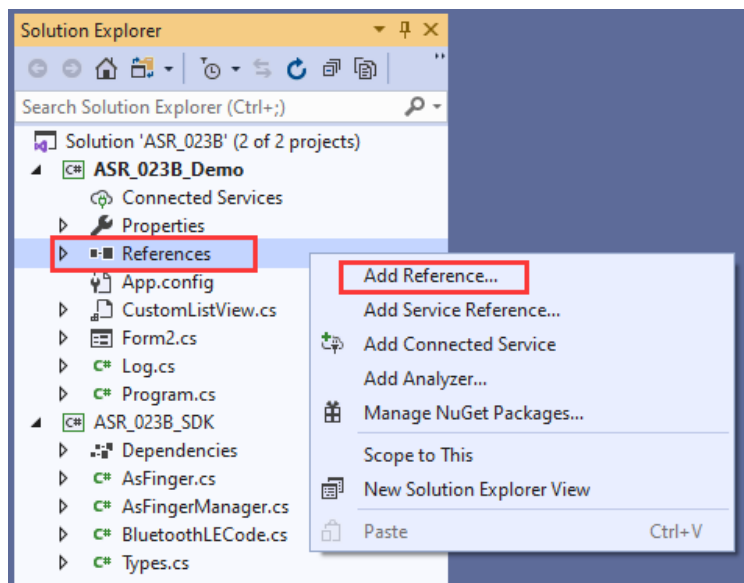
## 1.1. Import SDK

1. Create a new Windows Forms application.

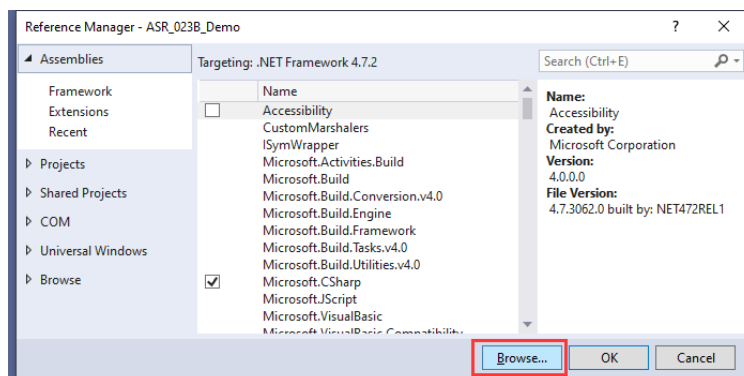
Copy the SDK into the project folder. If you are debugging, copy it to the bin/Debug folder of the application's current directory. If you are releasing, copy it to the bin/Release folder of the application's current directory.

2. Add Reference

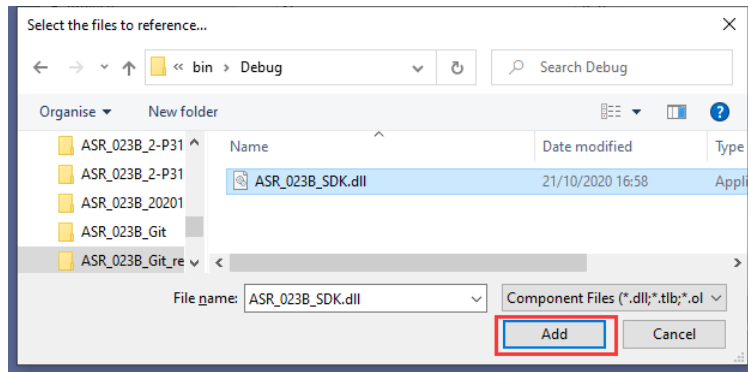
- 1) Right click the "References" and choose the "Add Reference".



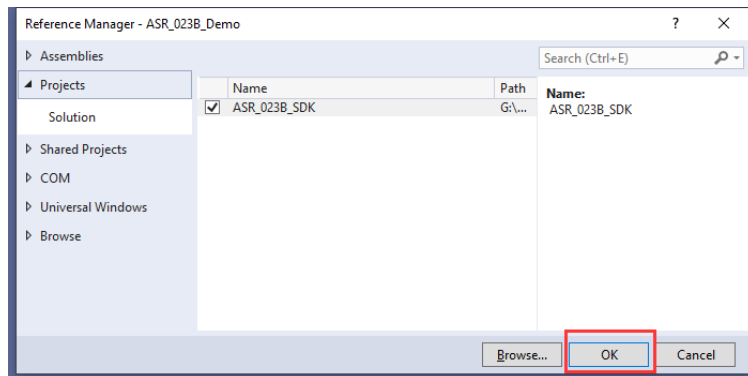
- 2) Click the "Browse" button.



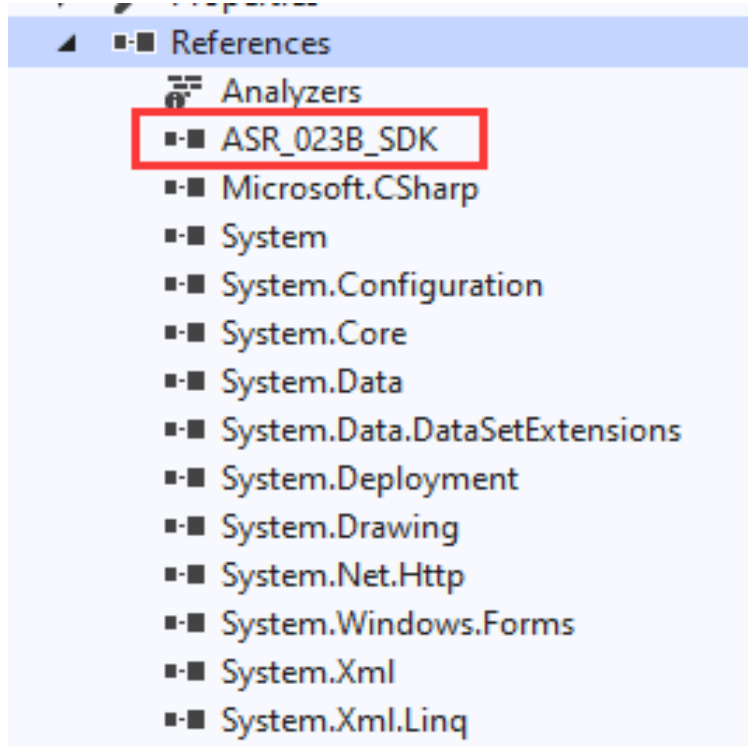
- 3) In the "Select the Files Reference" window, Select the SDK files in the corresponding path of the project and click the "Add" button.



- 4) Click the "OK" button on the Reference Manager page to complete the addition of SDK files.



- 5) After the reference is added successfully, it will be displayed in the reference list, as shown in the figure below:



## 1.2. SDK Usage

Here is a brief description of the use of the SDK, containing connecting devices to scanning barcodes and so on.

### 1.2.1. Reference the Namespace

```
using ASR_023B_SDK;
```

### 1.2.2. Search Devices

#### 1. Get the object

Call the "Public static AsFingerManager SharedAsFingerManager()" function in the "AsFingerManager" class to get the singleton object for AsFingerManager.

```
asFingerManager = AsFingerManager.SharedAsFingerManager();  
asFingerManager.whenReceivedAsFingerDevice += WhenReceivedAsFinger;
```

#### 2. Assign a delegate

##### 1) Create a function whose parameter type is BluetoothDevice.

This delegate function is used to listen for the event that an AsReader Finger-Type device is found. Once an AsReader Finger-Type device is found, the delegate function is called back. (The screenshot below is only sample code.)

```
private void WhenReceivedBluetoothDevice(List<BluetoothDevice> devices)  
{  
    //It is to process the searched AsReader Finger-Type devices.  
}
```

##### 2) Assign to the whenReceivedAsFingerDevice delegate.

Assign the function created in Section [1.2.2-2-\(1\)](#) to the whenReceivedAsFingerDevice delegate in the AsFingerManager class.

```
asFingerManager = AsFingerManager.SharedAsFingerManager();  
asFingerManager.whenReceivedAsFingerDevice += WhenReceivedAsFinger;
```

#### 3. Perform searching

Call the function "public void StartSearching(BluetoothType Type)" in the AsFingerManager class to search for the AsReader Finger-Type devices. To call this function, a BluetoothType type parameter need to be passed into it. This parameter is used to limit the Bluetooth mode (Bluetooth mode: SPP or BLE) used to search for AsReader Finger-Type devices. Once any AsReader Finger-Type device is found, the function created in Section [1.2.2-2-\(1\)](#) will be called back.

```
Types.BluetoothType bluetoothType = Types.BluetoothType.BluetoothBLE;  
//Types.BluetoothType bluetoothType = Types.BluetoothType.BluetoothSPP;  
asFingerManager.StartSearching(bluetoothType);
```



### 1.2.3. Stop Searching

Call the "public void StopSearching()" function in the AsFingerManager class to stop the search for the AsReader Finger-Type devices.

```
asFingerManager.StopSearching();
```

### 1.2.4. Connect to the Device

#### 1. Get the object

Call the "public static AsFinger SharedAsFinger()" function in the AsFinger class to get the AsFinger singleton object.

```
asFinger = AsFinger.SharedAsFinger();
```

#### 2. Assign a delegate

##### 1) Create a function of parameter type bool

This delegate function is used to listen for changes in the connection state of the AsReader Finger-Type device. The function is called back once the connection state changes. When the value of parameter isConnected is true, it means connected; when it is false, it means not connected. (The screenshot below is only sample code.)

```
private void WhenAsFingerIsConnected(bool isConnected)
{
    //It is to process the result of the connection being successful or not.
}
```

##### 2) Assign to the whenAsFingerIsConnected delegate.

Assign the function created in Section [1.2.4-2-\(1\)](#) to the whenAsFingerIsConnected delegate in the AsFinger class.

```
asFinger.whenAsFingerIsConnected = WhenAsFingerIsConnected;
```

#### 3. Connect to the device

Call the "public void Connect(BluetoothDevice device)" function in the AsFinger class to connect to the AsReader Finger-Type device. The parameter value passed in to this function is the object of one of the devices in the AsReader Finger-Type device list passed in when the function created in Section 1.2.2-2-(1) is called back. Once the device is successfully connected, the function created in section [1.2.4-2-\(1\)](#) will be called back, with the parameter isConnected passing in the value true.

```
BluetoothDevice device = DeviceDic[deviceID];
asFinger.Connect(device);
```

### 1.2.5. Barcode Scan

#### 1. Assign a delegate

##### 1) Create a function whose parameter type is byte[].

This delegate function is used to listen for the event that the AsReader Finger-Type device scanned barcode data. Once the device scans any barcode data, the delegate will be called back.

```
private void WhenReceiveBarcodeData(byte[] data)
{
    //It is to process the scanned barcode data
}
```

- 2) Assign to the whenReceivedBarcodeData delegate.

Assign the functions created in Section [1.2.5-1-\(1\)](#) to the whenReceivedBarcodeData delegate in the AsFinger class.

```
asFinger.whenReceivedBarcodeData = WhenReceiveBarcodeData;
```

2. Scan barcodes

Call the "public void StartScan()" function in the AsFinger class to scan barcodes. Once any barcode is scanned, the function created in section [1.2.5-1-\(1\)](#) will be called back, with the parameter "data" being passed in a value that is the scanned barcode data.

```
asFinger.StartScan();
```

### 1.2.6. Stop Scanning

Call the "public void StopScan()" function in the AsFinger class to stop scanning.

```
asFinger.StopScan();
```

### 1.2.7. Disconnection

Call the "public Void Disconnect()" function in the AsFinger class to disconnect the AsReader Finger-Type device. Once disconnected from the device, the function created in section [1.2.4-2-\(1\)](#) will be called back with the parameter "isConnected" being passed in with a value of false.

```
asFinger.Disconnect();
```

## 2. AsFinger Class

### 2.1. Functions

#### 2.1.1. Connect

Function	void Connect(BluetoothDevice device);			
Parameters	Names	In/Out	Types	Descriptions
	device	-	BluetoothDevice	Contains the Mac address and name of the AsReader Finger-Type device, and the BluetoothDevice object.
Return Values	-	-	-	-
<p>■Function Description: Connects to the AsReader Finger-Type device. Once this function is called, the connection result is called back through the "WhenAsFingerIsConnected" (see <a href="#">2.2.1</a>) delegate in the AsFinger class.</p> <p>■Conditions for Call: When any AsReader Finger-Type device has been found. That is, this function can be called only when the delegate function (see <a href="#">1.2.2-2-(1)</a>) assigned by the "WhenReceivedBluetoothDevice" delegate (see <a href="#">3.2.1</a>) of the AsFingerManager class is called back and with the parameter is not null.</p> <p>■Sample Code: (Note: asFinger is a singleton object of the AsFinger class) asFinger.Connect(device);</p>				

#### 2.1.2. Disconnect

Function	void Disconnect();			
Parameters	Names	In/Out	Types	Descriptions
	-	-	-	-
Return Values	-	-	-	-

■Function Description:

Disconnects from the AsReader Finger-Type device.

Once this function is called, the connection state is called back through the "WhenAsFingerIsConnected" delegate of the AsFinger class (see [2.2.1](#)).

■Conditions for Call:

When any AsReader Finger-Type device has been connected to. That is, this function can be called only when the delegate function (see [1.2.4-2-\(1\)](#)) assigned by the "WhenAsFingerIsConnected" delegate (see [2.2.1](#)) is called back and with the parameter is in a value of "true".

■Sample Code: (Note: asFinger is a singleton object of the AsFinger class)

```
asFinger.Disconnect();
```

### 2.1.3. StartScan

Function	void StartScan();			
Parameters	Names	In/Out	Types	Descriptions
	-	-	-	-
Return Values	-	-	-	-

■Function Description:

Starts to scan barcodes.

Once this function is called, the scanned barcode data will be called back through the "WhenReceivedBarcodeData" delegate of the AsFinger class (see [2.2.2](#)).

■Conditions for Call:

When any AsReader Finger-Type device has been connected to. That is, this function can be called only when the delegate function (see [1.2.4-2-\(1\)](#)) assigned by the "WhenAsFingerIsConnected" delegate (see [2.2.1](#)) is called back and with the parameter is in a value of "true".

■Sample Code: (Note: asFinger is a singleton object of the AsFinger class)

```
asFinger.StartScan();
```

### 2.1.4. StopScan

Function	void StopScan();			
Parameters	Names	In/Out	Types	Descriptions
	-	-	-	-
Return Values	-	-	-	-

**■Function Description:**  
 Stops scanning barcodes.

**■Conditions for Call:**  
 When any AsReader Finger-Type device has been connected to. That is, this function can be called only when the delegate function (see [1.2.4-2-\(1\)](#)) assigned by the "WhenAsFingerIsConnected" delegate (see [2.2.1](#)) is called back and with the parameter is in a value of "true".

**■Sample Code:** (Note: asFinger is a singleton object of the AsFinger class)  
 asFinger.StopScan();

### 2.1.5. GetBattery

Function	void GetBattery();			
Parameters	Names	In/Out	Types	Descriptions
	-	-	-	-
Return Values	-	-	-	-

**■Function Description:**  
 Gets the battery value of the current connected AsReader Finger-Type device. Once this function is called, the battery value will be called back through the "WhenReceivedBattery" delegate of the AsFinger class (see [2.2.3](#)).

**■Conditions for Call:**  
 When any AsReader Finger-Type device has been connected to. That is, this function can be called only when the delegate function (see [1.2.4-2-\(1\)](#)) assigned by the "WhenAsFingerIsConnected" delegate (see [2.2.1](#)) is called back and with the parameter is in a value of "true".

■Sample Code: (Note: asFinger is a singleton object of the AsFinger class)  
 asFinger.GetBattery();

### 2.1.6. GetFirmwareVersion

Function	void GetFirmwareVersion();			
Parameters	Names	In/Out	Types	Descriptions
	-	-	-	-
Return Values	-	-	-	-
<p>                     ■Function Description:                      Gets the firmware version of the current connected AsReader Finger-Type device.                      Once this function is called, the battery value will be called back through the "WhenReceivedFirmwareVersion" delegate of the AsFinger class (see <a href="#">2.2.4</a>).                 </p> <p>                     ■Conditions for Call:                      When any AsReader Finger-Type device has been connected to. That is, this function can be called only when the delegate function (see <a href="#">1.2.4-2-(1)</a>) assigned by the "WhenAsFingerIsConnected" delegate (see <a href="#">2.2.1</a>) is called back and with the parameter is in a value of "true".                 </p> <p>                     ■Sample Code: (Note: asFinger is a singleton object of the AsFinger class)                      asFinger.GetFirmwareVersion();                 </p>				

### 2.1.7. GetSDKVersion

Function	string GetSDKVersion();			
Parameters	Names	In/Out	Types	Descriptions
	-	-	-	-
Return Values	-	Out	String	SDK version
<p>                     ■Function Description:                      Gets the SDK version.                 </p> <p>                     ■Conditions for Call:                 </p>				

None.

■Sample Code: (Note: asFinger is a singleton object of the AsFinger class)

```
string ver = asFinger.GetSDKVersion(); // ver: SDK version
```

### 2.1.8. GetBluetoothDevice

Function	BluetoothDevice GetBluetoothDevice();			
Parameters	Names	In/Out	Types	Descriptions
	-	-	-	-
Return Values	-	Out	BluetoothDevice	BluetoothDevice object (see <a href="#">4 BluetoothDevice Class</a> ).
<p>■Function Description: Returns the BluetoothDevice object of the AsFinger class.</p> <p>■Conditions for Call: None.</p> <p>■Sample Code: (Note: asFinger is a singleton object of the AsFinger class) asFinger.GetCurrentDevice();</p>				

### 2.1.9. SendData

Function	void SendData(byte[]data);			
Parameters	Names	In/Out	Types	Descriptions
	data	In	byte[]	Custom data
Return Values	-	-	-	-
<p>■Function Description: Sends the Custom data. Once this function is called, the received data will be called back through the "WhenReceivedData" delegate of the AsFinger class (see <a href="#">2.2.5</a>).</p> <p>■Conditions for Call: When any AsReader Finger-Type device has been connected to. That is, this function can be</p>				

called only when the delegate function (see [1.2.4-2-\(1\)](#)) assigned by the "WhenAsFingerIsConnected" delegate (see [2.2.1](#)) is called back and with the parameter is in a value of "true".

■Sample Code: (Note: asFinger is a singleton object of the AsFinger class)

```
string data = @"%%FIRM-VER";
```

```
asFinger.SendData(System.Text.UTF8Encoding.ASCII.GetBytes(data));
```

## 2.2. Delegates

### 2.2.1. WhenAsFingerIsConnected

Delegate	delegate void WhenAsFingerIsConnected(bool isConnected);			
Parameters	Names	In/Out	Types	Descriptions
	isConnected	In	Bool	The connect status of the AsReader Finger-Type: true: connected false: not connected
Return Values	-	-	-	-

■Delegate Description:

Listens to the connect status of the AsReader Finger-Type.

■Conditions for Callback:

1. When the function Connect (see [2.1.1](#)) or Disconnect (see [2.1.2](#)) is called.
2. When the connection status with the AsReader Finger-Type device changes.

■Sample Code:

```
WhenAsFingerIsConnected whenAsFingerIsConnected = WhenAsFingerIsConnectedProc;
```

```
void WhenAsFingerIsConnectedProc(bool isConnected){
```

```
    if (isConnected) {
```

```
        // The processing of the case "connected"
```

```
    } else {
```

```
        // The processing of the case "not connected"
```

```
    }
```

```
}
```

```
whenAsFingerIsConnected(isConnected);
```



//isConnected: the status of the connection. true: connected; false: not connected.

## 2.2.2. WhenReceivedBarcodeData

Delegate	delegate void WhenReceivedBarcodeData(byte[] data);			
Parameters	Names	In/Out	Types	Descriptions
	data	In	byte[]	Barcode data read by the AsReader Finger-Type device
Return Values	-	-	-	-
<p>■Delegate Description: Listen for the data returned when the AsReader Finger-Type device reads the barcode.</p> <p>■Conditions for Callback:</p> <ol style="list-style-type: none"> <li>1. When the "StartScan" (see <a href="#">2.1.3</a>) function is called and there is any barcode data is returned.</li> <li>2. When the scan button of AsReader Finger-Type device is pressed and there is any barcode data is returned.</li> </ol> <p>■Sample Code:</p> <pre>WhenReceivedBarcodeData whenReceivedBarcodeData = WhenReceivedBarcodeDataProc; void WhenReceivedBarcodeDataProc(byte[] data){     // Processes the barcode data read by the AsReader Finger-Type device. } whenReceivedBarcodeData(data);//data: Barcode data read by the AsReader Finger-Type device</pre>				

## 2.2.3. WhenReceivedBattery

Delegate	delegate void WhenReceivedBattery(int battery);			
Parameters	Names	In/Out	Types	Descriptions
	battery	In	Int	The value of the remaining battery power of the AsReader Finger-Type device
Return Values	-	-	-	-
<p>■Delegate Description:</p>				

Listens for the value of the remaining battery power returned by the AsReader Finger-Type device.

■Conditions for Callback:

This delegate can only be called back after the GetBattery (see [2.1.5](#)) function is called.

■Sample Code:

```
WhenReceivedBattery whenReceivedBattery = WhenReceivedBatteryProc;
void WhenReceivedBatteryProc(int battery) {
    // Processes the value of the remaining battery power returned by the AsReader Finger-Type
    // device.
}
whenReceivedBattery(battery); // battery: The value of the remaining power of the battery of the
// AsReader Finger-Type device received.
```

#### 2.2.4. WhenReceivedFirmwareVersion

Delegate	delegate void WhenReceivedFirmwareVersion(string ver);			
Parameters	Names	In/Out	Types	Descriptions
	ver	In	String	The firmware version information for the AsReader Finger-Type device.
Return Values	-	-	-	-

■Delegate Description:

Listens for the firmware version information returned by the AsReader Finger-Type device.

■Conditions for Callback:

This delegate can only be called back after the GetFirmwareVersion (see [2.1.6](#)) function is called.

■Sample Code:

```
WhenReceivedFirmwareVersion whenReceivedFirmwareVersion =
WhenReceivedFirmwareVersionProc;
void WhenReceivedFirmwareVersionProc(string ver){
    // Processes the firmware version information returned by the AsReader Finger-Type device.
}
```

```
whenReceivedFirmwareVersion(ver); // ver: The firmware version information returned by the
// AsReader Finger-Type device
```

### 2.2.5. WhenReceivedData

Delegate	delegate void WhenReceivedData(byte[] data);			
Parameters	Names	In/Out	Types	Descriptions
	data	In	byte[]	The data returned from the AsReader Finger-Type
Return Values	-	-	-	-

■ Delegate Description:

Listens for any data returned by the AsReader Finger-Type device.

■ Conditions for Callback:

When the "StartScan" (see [2.1.3](#)) function is called and there is any barcode data is returned.

When the scan button of AsReader Finger-Type device is pressed and there is any barcode data is returned.

When the GetBattery (see [2.1.5](#)) function is called.

When the GetFirmwareVersion(see [2.1.6](#)) function is called.

When the SendData (see [2.1.9](#)) function is called.

■ Sample Code:

```
WhenReceivedData whenReceivedData = ReceivedDataProc;
```

```
void ReceivedDataProc(byte[] data){
```

```
    // Processes the data returned by the AsReader Finger-Type device.
```

```
}
```

```
whenReceivedData(data);// data: The data returned by the AsReader Finger-Type device.
```

## 3. AsFingerManager Class

### 3.1. Functions

#### 3.1.1. StartSearching

Function	void StartSearching(Types.BluetoothType type);			
Parameters	Names	In/Out	Types	Descriptions
	type	In	Types.BluetoothType	Bluetooth mode: BluetoothBLE: BLE mode BluetoothSPP: SPP mode
Return Values	-	-	-	-
<p>■Function Description: Starts to search the AsReader Finger-Type devices. Once this function is called, the object of the searched AsReader Finger-Type device will be called back through the “WhenReceivedBluetoothDevice” delegate (see <a href="#">3.2.1</a>) of the AsFingerManager class.</p> <p>■Conditions for Call: The Windows Bluetooth is on.</p> <p>■Sample Code: (Note: asFingerManager is a singleton object of the AsFingerManager class) asFingerManager.StartSearching(Types.BluetoothType.BluetoothBLE);</p>				

#### 3.1.2. StopSearching

Function	void StopSearching();			
Parameters	Names	In/Out	Types	Descriptions
	-	-	-	-
Return Values	-	-	-	-
<p>■Function Description: Stops searching the AsReader Finger-Type devices.</p>				

■Conditions for Call:

1. The Windows Bluetooth is on.
2. Currently in the state of searching the Bluetooth devices, that is, the StartSearching (see [3.1.1](#)) function has been called.

■Sample Code: (Note: asFingerManager is a singleton object of the AsFingerManager class)

```
asFingerManager.StopSearching();
```

### 3.1.3. GetSearchBluetoothType

Function	Types.BluetoothType GetSearchBluetoothType();			
Parameters	Names	In/Out	Types	Descriptions
	-	-	-	-
Return Values	-	Out	Types.BluetoothType	The Bluetooth mode used when searching for AsReader Finger-Type devices.

■Function Description:

Gets the Bluetooth mode used when searching for AsReader Finger-Type devices.

■Conditions for Call:

None.

■Sample Code: (Note: asFingerManager is a singleton object of the AsFingerManager class)

```
asFingerManager.GetSearchBluetoothType();
```

## 3.2. Delegates

### 3.2.1. WhenReceivedBluetoothDevice

Delegate	delegate void WhenReceivedBluetoothDevice(List<BluetoothDevice> devices);			
Parameters	Names	In/Out	Types	Descriptions
	devices	In	List<BluetoothDevice>	The list of the objects of the AsReader Finger-Type devices that be found

Return Values	-	-	-	-
<p>■Delegate Description: Listens for the list of the objects of the AsReader Finger-Type devices that be found by Windows Bluetooth.</p> <p>■Conditions for Callback: This delegate can only be called back when the StartSearching (see <a href="#">3.1.1</a>) function is called and an AsReader Finger-Type device is searched.</p> <p>■Sample Code: WhenReceivedBluetoothDevice whenReceivedBluetoothDevice = WhenReceivedBluetoothDeviceProc; void WhenReceivedBluetoothDeviceProc(List&lt;BluetoothDevice&gt; devices) {     // Processes the list of the objects of the AsReader Finger-Type devices that be found. } whenReceivedBluetoothDevice(devices);// devices: The list of the objects of the AsReader Finger-Type devices that be found.</p>				

## 4. BluetoothDevice Class

### 4.1. Properties

Property Names	Properties	Types	Descriptions
DeviceBluetoothType	get/set	Types.BluetoothType	The Bluetooth mode that the AsReader Finger-Type device is currently in
DeviceName	get/set	String	The Bluetooth name of the AsReader Finger-Type device
DeviceID	get/set	String	The MAC address of the AsReader Finger-Type device

## 5. Types Class

### 5.1. Enumeration Types

#### 5.1.1. BluetoothType

Definition	Descriptions
BluetoothBLE	BLE mode
BluetoothSPP	SPP mode