**PenPower A6 Scan Control SDK**

**.NET API Reference**

Version 1.0.1

**PPScanControl Namespace**

**Classes**

[ScanControl](#ScanControl_Class) Class

**Methods:**

[GetApiVersion](#GetApiVersion)

[GetConnectedScanner](#GetConnetedScanner)

[GetOpenedScanner](#GetOpenedScanner)

[Open](#Open)

[IsCalibrateNeeded](#IsCalibrateNeeded)

[Calibrate](#Calibrate)

[StartScan](#StartScan)

[StartScanBlocking](#StartScanBlocking)

[Close](#Close)

**Events:**

[ScanCompleted](#ScanCompleted)

[ButtonPressed](#ButtonPressed)

[ScanCompletedEventArgs](#ScanCompletedEventArgs_Class) Class

[ButtonEventArgs](#ButtonEventArgs_Class) Class

**Enums**

[ScannerType](#ScannerType_definition) Enum

[ImageType](#ImageType_definition) Enum

[ReturnCode](#Return_code_definition) Enum

**ScanControl Class**

1. **GetApiVersion**

Get the version of the SDK library.

|  |
| --- |
| string GetApiVersion(); |

**Parameter: void**

**Return value: string**

Returns the version number of the SDK library, in a three-number dotted format string (ex: “1.0.0”).

1. **GetConnetedScanner**

Get the current connected scanner device ID.

|  |
| --- |
| ScannerType GetConnectedScanner(); |

**Parameter: void**

**Return value: ScannerType**

Returns which type of scanner hardware is detected on the USB bus. If more than one type of scanners is connected, ID A6\_7P3 would be returned.

1. **GetOpenedScanner**

Get the currently opened scanner ID.

|  |
| --- |
| ScannerType GetOpenedScanner(); |

**Parameter: void**

**Return value: ScannerType**

Returns which scanner is currently opened by scan control SDK.

1. **Open**

Open the specified scanner device.

|  |
| --- |
| ReturnCode Open(ScannerType scannerType); |

**Parameters:**

**ScannerType scannerType**

[In] Specify the ID of the device to initialize.

**Return value: ReturnCode**

See return code definition.

1. **IsCalibrateNeeded**

Test if the currently opened scanner needs calibration.

|  |
| --- |
| ReturnCode IsCalibrateNeeded(); |

**Parameters: void**

**Return value: ReturnCode**

If the scanner needs calibration, this method returns NeedCalibrate, If the scanner doesn’t need calibration, this method returns Success. For other errors, see return code definition.

1. **Calibrate**

Calibrates the currently opened scanner.

|  |
| --- |
| ReturnCode Calibrate(); |

**Parameter: void**

**Return value: ReturnCode**

Returns Success if the process ended normally, or NeedCalibrate if failed. For other errors, see return code definition.

**Remarks:**

This method blocks until calibrate process returns. This method does not have any calibrating UI.

1. **StartScan**

Start a scan procedure.

|  |
| --- |
| ReturnCode StartScan(int dpi, ImageType type, string outputFileName); |

**Parameter:**

**int dpi**

[In] Specify the scanning DPI. Scanner ID A6\_73F has minimum value 300, and A6\_7P3 has minimum value 200.

**ImageType type**

[In] Specify the output image file’s format.

**string outputFileName**

[In] Specify the output image file’s full path. If this file already exists, it will be overwritten. If outputFileName is null, the SDK returns the result image in-memory instead of on-disk.

**Return value: ReturnCode**

See error code definition.

**Remarks:**

This method returns immediately. If this method succeeds, result of the scanning procedure would be raised through [ScanCompleted](#ScanCompleted) event.

1. **StartScanBlocking**

Start a scan procedure.

|  |
| --- |
| ScanCompletedEventArgs StartScanBlocking(int dpi, ImageType type, string outputFileName); |

**Parameter:**

**int dpi**

[In] Specify the scanning DPI. Scanner ID A6\_73F has minimum value 300, and A6\_7P3 has minimum value 200.

**ImageType type**

[In] Specify the output image file’s format.

**string outputFileName**

[In] Specify the output image file’s full path. If this file already exists, it will be overwritten. If outputFileName is null, the SDK returns the result image in-memory instead of on-disk.

**Return value: ScanCompletedEventArgs**

See [ScanCompletedEventArgs](#ScanCompletedEventArgs_Class) class definition.

**Remarks:**

This method blocks until scan procedure is completed. It will not raise [ScanCompleted](#ScanCompleted) event.

1. **Close**

Close any opened scanner, abort scanning procedure. All the resources would be released, and the connected device would be uninitialized.

|  |
| --- |
| void Close(); |

**Parameter: void**

**Return value: void**

1. **ScanCompleted Event**

Event for a scanning procedure completed.

|  |
| --- |
| event EventHandler<ScanCompletedEventArgs> ScanCompleted; |

**Remarks:**

This event is raised after an image is scanned, or an error occurred during scanning/image processing. See definition of [ScanCompletedEventArgs](#ScanCompletedEventArgs_Class) class.

1. **ButtonPressed Event**

Event for a hardware button press.

|  |
| --- |
| event EventHandler<ButtonEventArgs> ButtonPressed; |

**Remarks:**

This event is raised after a button on A6\_7P3 scanner is pressed, other scanners will not raise this event. See definition of [ButtonEventArgs](#ButtonEventArgs_Class) class.

**ScanCompletedEventArgs Class**

**Fields**

**ReturnCode Result**

See error code definition.

**string FileName**

The output image file name.

If Result is Success and outputFileName is specified when start scanning, this field contains the same full file path as outputFileName of the corresponding [StartScan](#StartScan) method. Otherwise, this filed is null.

**string ImageData**

The output in-memory image data, as a base-64 encoded string.

If Result is Success and outputFileName is null when start scanning, this field contains the image data. Otherwise, this filed is null.

**ButtonEventArgs Class**

**Fields**

**int ButtonId**

The identifier of pressed buttons on A6\_7P3 scanner. [Scan] button is 1, and [Custom] button is 2.

**Return code definition**

public enum ReturnCode

{

Success = 0,

Busy,

InvalidParameter,

IOException,

ScannerNotFound,

ScannerOpenFailed,

NoPaper,

PaperJam,

NeedCalibrate,

DllNotFound,

DriverError,

Uninitalized,

OtherError

}

|  |  |  |
| --- | --- | --- |
| Error code | Value | Explanation |
| Success | 0 | Operation succeeds. |
| Busy | 1 | Another operation is occupying the SDK. |
| InvalidParameter | 2 | Given scanning parameter is invalid. |
| IOException | 3 | IO errors occurred during scanning procedure. |
| ScannerNotFound | 4 | Given scanner ID is not connected or invalid. |
| ScannerOpenFailed | 5 | Cannot open the scanner. |
| NoPaper | 6 | The scanner has no paper inside feeder. |
| PaperJam | 7 | The scanner has paper jammed inside feeder. |
| NeedCalibrate | 8 | The scanner needs calibrate. |
| DllNotFound | 9 | An underlying assembly cannot be loaded. |
| DriverError | 10 | An error detected in the scanner driver. |
| Uninitalized | 11 | This method required [Open](#Open) being called first. |
| OtherError | 12 | Other uncategorized errors. |

**ScannerType definition**

Specify the scanner ID. A6\_7P3 is the 2022 new model (black color). A6\_73F is the old model (white color).

public enum ScannerType

{

None = 0,

A6\_7P3,

A6\_73F

}

**ImageType definition**

public enum ImageType

{

BMP = 0,

JPG,

PNG

}