IBScanUltimate API Manual for C/C++ on Windows Mobile

Version 2.0.1 (April 27, 2018)

INTEGRATED
BIOMETRICS

Table of Contents

Revision History

Date	Author	Remarks
2015/8	Young	Added descriptions of new functions for IBScanUltimate v1.8.5 IBSU_CombineImage()
2015/3	Young	A (ASCII) version of APIs was removed for IBScanUltimate v1.8.3. According to Microsoft documentation specifies that: the .NET compact Framework only supports Unicode, and consequently only includes the CharSet.Unicode (and CharSet.Auto which equals Unicode) value, and does not support any of the clauses of the Declare statement. This means that the ExactSpelling property is also not supported. Added descriptions of new function IBSU_RedrawClientWindow()
2014/10	Young	Added descriptions of new functions for IBScanUltimate v1.8.2 IBSU_WSQEncodeMem(), IBSU_WSQEncodeToFile(), IBSU_WSQDecodeMem(), IBSU_WSQDecodeFromFile(), IBSU_FreeMemory(), IBSU_SavePngImage(), IBSU_SaveJP2Image()
2014/4	Jeff	Created document

1. Windows Mobile API Overview

The Windows Mobile API set is nearly identical to the Windows API set. This document only covers the differences between the Windows Mobile APIs and the Windows APIs. The primary reference for the Windows Mobile IBScanUltimate APIs is "IBScanUltimate API Manual for C.pdf".

The difference between the two versions is that the Windows Mobile IBScanUltimate API set is extended to add support for UNICODE.

Since v1.8.3, A version of API was removed.
According to Microsoft documentation specifies that:
the .NET compact Framework only supports Unicode, and consequently only includes the CharSet.Unicode (and CharSet.Auto which equals Unicode) value, and does not support any of the clauses of the Declare statement. This means that the ExactSpelling property is also not supported.

2. API Function Lists

2.1. C Functions

2.1.1. Summary of Interface Functions different on Windows Mobile

No	Functions
1	IBSU_GetSDKVersion split in to UNICODE versions
2	IBSU_GetDeviceDescription split in to UNICODE versions
3	IBSU_OpenDeviceEx split in to UNICODE versions
4	IBSU_GetProperty split in to UNICODE versions
5	IBSU_SetProperty split in to UNICODE versions
6	IBSU_GetClientWindowProperty split in to UNICODE versions
7	IBSU_SetClientWindowProperty split in to UNICODE versions
8	IBSU_SetClientWindowOverlayText split in to UNICODE versions
9	IBSU_SaveBitmapImage split in to UNICODE versions
10	IBSU_AddOverlayText split in to UNICODE versions
11	IBSU_ModifyOverlayText split in to UNICODE versions
12	IBSU_WSQEncodeToFile split in to UNICODE versions
13	IBSU_WSQDecodeFromFile split in to UNICODE versions
14	IBSU_SavePngImage split in to UNICODE versions
15	IBSU_SaveJP2Image split in to UNICODE versions
16	IBSU_CombineImage split in to UNICODE versions

Table 1

2.1.2. Function Details

1.1.2.1) IBSU_GetSDKVersion

- *Description*

    This function maps to IBSU_GetSDKVersionW when UNICODE is defined

1.1.2.2) IBSU_GetSDKVersionW

- *Prototype*

API DLL	int WINAPI IBSU_GetSDKVersionW (IBSU_SdkVersionW *pVerinfo)

- *Description*

    Gets a structure holding product and software version information, UNICODE version

- *Parameter*

Parameter	Description
*pVerinfo	*[out] API version information. Memory must be provided by caller.*

- *Return*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.3) IBSU_DeviceDescription

- *Description*

    This function maps to IBSU_DeviceDescriptionW when UNICODE is defined.

1.1.2.4) IBSU_ DeviceDescriptionW

- *Prototype*

API DLL	int WINAPI IBSU_GetDeviceDescriptionW (const int deviceIndex, IBSU_DeviceDescW *pDeviceDesc)

- *Description*

    Retrieves detailed device information about a particular scanner by logical index, UNICODE version.

- *Parameter*

Parameter	Description
deviceIndex	[in] Zero-based device index for device to lookup
*pDeviceDesc	*[out] Basic device information. Memory must be provided by caller.*

- *Return*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.5) IBSU_OpenDeviceEx

- *Description*

    This function maps to IBSU_OpenDeviceExW when UNICODE is defined.

1.1.2.6) IBSU_ OpenDeviceExW

- *Prototype*

API DLL	int WINAPI IBSU_OpenDeviceExW (const int deviceIndex, LPCWSTR uniformityMaskPath, const BOOL asyncOpen, int *pHandle)

- *Description*

    Extension initialize device(fast mode), given a particular by device index, UNICODE version

- *Parameter*

Parameter	Description
deviceIndex	[in] Zero-based device index for device to init
uniformityMaskPath	[in] Uniformity mask path in your computer If the file does not exist or different in path, the DLL makes a new file in path.
asyncOpen	[in] async open device(TRUE) or sync open device(FALSE)

*pHandle	[out] Function returns device handle to be used for subsequent function calls. Memory must be provided by caller

- *Return*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

- *Note*

    Any initialized device must be released before closing the host application!
    (call IBSU_CloseDevice() or IBSU_CloseAlldevice())

1.1.2.7)   IBSU_ GetProperty

- *Description*

    This function maps to IBSU_GetPropertyW when UNICODE is defined.

1.1.2.8)   IBSU_ GetPropertyW

- *Prototype*

API DLL	int IBSU_GetPropertyW(const int handle,
	const IBSU_PropertyId propertyId,
	LPWSTR propertyValue)

- *Description*

    Retrieves a particular device's property value (by handle), UNICODE version

- *Parameter*

Parameter	Description
handle	[in] Device handle obtained by IBSU_OpenDevice().
propertyId	[in] Property identifier to get value for.
propertyValue	*[out] String returning property value. Memory must be provided by caller.*

- *Return*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.9)   IBSU_ SetProperty

- *Description*

    This function maps to IBSU_SetPropertyW when UNICODE is defined.

1.1.2.10)   IBSU_ GetPropertyW

- *Prototype*

API DLL	int IBSU_SetPropertyW(const int handle, const IBSU_PropertyId propertyId, LPCWSTR propertyValue)

- *Description*

    Set a device's property value (by handle), UNICODE version

- *Parameter*

Parameter	Description
handle	[in] Device handle obtained by IBSU_OpenDevice().
propertyId	[in] Property identifier to set value for.
propertyValue	[in] String containing property value.

- *Return*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

- *Note*
    Only specific property values can be set.

1.1.2.11)   IBSU_ GetClientWindowProperty

- *Description*

    This function maps to IBSU_GetClientWindowPropertyW when UNICODE is defined.

1.1.2.12)   IBSU_ GetClientWindowPropertyW

- *Prototype*

API DLL	int IBSU_GetClientWindowPropertyW (const int handle, const IBSU_ClientWindowPropertyId propertyId, LPWSTR

	propertyValue)

- *Description*

    Get user-defined window properties, UNICODE version.

- *Parameters*

Parameter	Description
handle	[in] Device handle obtained by IBSU_OpenDevice()
propertyId	[in] Property identifier to set value
propertyValue	[out] String returning the property's value. (Memory must be provided by caller)

- *Returns*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.13)   IBSU_ SetClientDisplayProperty

- *Description*

    This function maps to IBSU_SetClientWindowPropertyW when UNICODE is defined.

1.1.2.14)   IBSU_ SetClientDisplayPropertyW

- *Prototype*

API DLL	int WINAPI IBSU_SetClientDisplayPropertyW(const int handle, const IBSU_ClientWindowPropertyId propertyId, LPCWSTR propertyValue)

- *Description*

    Set a user-define window property, UNICODE version.

- *Parameters*

Parameter	Description
handle	[in] Device handle obtained by IBSU_OpenDevice()
propertyId	[in] Property identifier to set value
propertyValue	[out] String returning property value. (Memory must be provided

	by caller)

- *Returns*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.15)　　IBSU_ SetClientWindowOverlayText (Deprecated)

- *Description*

    This function maps to IBSU_SetClientWindowOverlayTextW when UNICODE is
    defined.

1.1.2.16)　　IBSU_ SetClientWindowOverlayTextW (Deprecated)

- *Prototype*

API DLL	Int WINAPI IBSU_SetClientWindowOverlayTextW (const int handle, const WCHAR *fontName, const int fontSize, const BOOL fontBold, const WCHAR *text, const int posX, cons tint posY, const DWRD textColor)

- *Description*

    Set the text property on a user-defined window, UNICDE version.

- *Parameters*

Parameter	Description
handle	[in] Device handle obtained by IBSU_OpenDevice()
*fontName	[in] font name for display
fontsize	[in] font size for display
fontBold	[in] font bold for display
*text	[in] string for display
posX	[in] X coordinate of text for display
posY	[in] Y coordinate of text for display
textColor	[in] string color for display

- *Returns*

Return Value	Description
0	Function completed successfully.

< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.17)   IBSU_ SaveBitmapImage

- *Description*

    This function maps to IBSU_SaveBitmapImageW when UNICODE is defined.

1.1.2.18)   IBSU_ SaveBitmapImageW

- *Prototype*

API DLL	Int WINAPI IBSU_SaveBitmapImageW (LPCWSTR filepath, const BYTE *imgBuffer, const DWORD width, const DWORD height, const int pitch, const double resX, const double resY)

- *Description*

    Save fingerprint image in bitmap format, UNICODE version.

- *Parameters*

Parameter	Description
filePath	[in] File path to save bitmap
*imgBuffer	[in] Point to raw image data (background color is black)
Width	[in] Image width
Height	[in] Image height
Pitch	[in] Image line pitch (Positive value indicate top down line order, Negative value mean bottom up line order)
resX	[in] Image horizontal resolution (in PPI)
resY	[in]Image vertical resolution (in PPI)

- *Returns*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.19)   IBSU_ AddOverlayText

- *Description*

    This function maps to IBSU_AddOverlayTextW when UNICODE is defined.

1.1.2.20)   IBSU_ AddOverlayTextW

- *Prototype*

API DLL	int WINAPI IBSU_AddOverlayTextW
	const int handle,
	int *pOverlayHandle,
	const WCHAR *fontName,
	const int fontSize,
	const BOOL fontBold,
	const WCHAR *text,
	const int posX,
	const int posY,
	const DWORD textColor
	);

- *Description*

  Add an overlay text for display on window.

- *Parameters*

Parameter	Description
handle	[in] Device handle
*pOverlayHandle	[out] Function returns overlay handle to be used for client windows functions call
*fontName	[in] Name of font.
fontSize	[in] Font size.
fontBold	[in] Indicates whether font is bold.
*text	[in] Text for display on window
posX	[in] X coordinate of text for display on window
posY	[in] Y coordinate or test for display on window
textColor	[in] Text color

- *Returns*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.21)   IBSU_ ModifyOverlayText

- *Description*

    This function maps to IBSU_ModifyOverlayTextW when UNICODE is defined.

1.1.2.22)   IBSU_ ModifyOverlayTextW

- *Prototype*

API DLL	int WINAPI IBSU_ModifyOverlayTextW(
	const int handle,
	int OverlayHandle,
	const wchar_t *fontName,
	const int fontSize,
	const BOOL fontBold,
	const wchar_t *text,
	const int posX,
	const int posY,
	const DWORD textColor
	);

- *Description*

    Modify an existing overlay text for display on window, Unicode version.

- *Parameters*

Parameter	Description
handle	[in] Device handle
OverlayHandle	[in] Handle of overlay to modify.
*fontName	[in] Name of font.
fontSize	[in] Font size.
fontBold	[in] Indicates whether font is bold.
*text	[in] Text for display on window
posX	[in] X coordinate of text for display on window
posY	[in] Y coordinate or test for display on window
textColor	[in] Text color

- *Returns*

Return Value	Description

0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.23)   IBSU_ WSQEncodeToFile

- *Description*

  This function maps to IBSU_WSQEncodeToFileW when UNICODE is defined.

1.1.2.24)   IBSU_ WSQEncodeToFileW

- *Prototype*

API DLL	Int WINAPI IBSU_WSQEncodeToFileW(LPCWSTR filePath, const BYTE *image, const int width, const int height, const int pitch, const int bitsPerPixel, const int pixelPerInch, const double bitRate, const WCHAR *commentText)

- *Description*

  Save WSQ compresses grayscale fingerprint image to specific file path, UNICODE version.

- *Parameters*

Parameter	Description
filePath	[in] File path to save image which is compressed from original image by WSQ compression
*image	[in] Original image
width	[in] Width of original image (in pixels)
height	[in] Height of original image (in pixels)
pitch	[in] Image line pitch (Positive value indicate top down line order, Negative value mean bottom up line order)
bitsPerPixel	[in] Bits per pixel of original image
pixelPerInch	[in] Pixel per inch of orginal image
bitRate	[in] Determines the amount of lossy compression Suggested settings: bitRate = 2.25 yields around 5:1 compression bitRate = 0.75 yields around 15:1 compression
*commentText	[in] Comment to write compressed data

- *Returns*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.25)   IBSU_ WSQDecodeFromFile

- *Description*

  This function maps to IBSU_WSQDecodeFromFileW when UNICODE is defined.

1.1.2.26)   IBSU_ WSQDecodeFromFileW

- *Prototype*

API DLL	Int WINAPI IBSU_WSQDecodeFromFileW(LPCWSTR filePath , BYTE **decompressedImage, int *outWidth, int *outHeight, int *outpitch, int *outBitsPerPixel, int *outPixelPerInch)

- *Description*

  Decompress a WSQ-encoded grayscale fingerprint image from specific file path, UNICODE version.

- *Parameters*

Parameter	Description
filePath	[in] File path of WSQ-encoded image
**decompressedImage	[out] Pointer of image which is decompressed from WSQ-encoded image. This pointer is deallocated by IBSU_FreeMemory() after using it
*outWidth	[out] Width of decompressed image (in pixels)
*outHeight	[out] Height of decompressed image (in pixels)
*outPitch	[out] Image line pitch (Positive value indicate top down line order, Negative value mean bottom up line order)
*outBitsPerPixel	[out] Bits per pixel of decompressed image
*outPixelPerInch	[out] Pixel per inch of decompressed image

- *Returns*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.27) IBSU_ SavePngImage (Not supported due to memory issue)

- *Description*

    This function maps to IBSU_SavePngImageW when UNICODE is defined.

1.1.2.28) IBSU_ SavePngImageW (Not supported due to memory issue)

- *Prototype*

API DLL	Int WINAPI IBSU_SavePngImageW (LPCWSTR filepath, const BYTE *image, const DWORD width, const DWORD height, const int pitch, const double resX, const double resY)

- *Description*

    Save fingerprint image in png format, UNICODE version.

- *Parameters*

Parameter	Description
filePath	[in] File path to save png
*image	[in] Point to raw image data (background color is black)
Width	[in] Image width
Height	[in] Image height
Pitch	[in] Image line pitch (Positive value indicate top down line order, Negative value mean bottom up line order)
resX	[in] Image horizontal resolution (in PPI)
resY	[in]Image vertical resolution (in PPI)

- *Returns*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.29) IBSU_ SaveJP2Image (Not supported due to memory issue)

- *Description*

This function maps to IBSU_SaveJP2ImageW when UNICODE is defined.

1.1.2.30)   IBSU_ SaveJP2ImageW (Not supported due to memory issue)

- *Prototype*

API DLL	Int WINAPI IBSU_SaveJP2ImageA (LPCWSTR filepath, const BYTE *image, const DWORD width, const DWORD height, const int pitch, const double resX, const double resY, const int fQuality)

- *Description*

Save fingerprint image in JPEG-2000 format, UNICODE version.

- *Parameters*

Parameter	Description
filePath	[in] File path to save jp2
*image	[in] Point to raw image data (background color is black)
Width	[in] Image width
Height	[in] Image height
Pitch	[in] Image line pitch (Positive value indicate top down line order, Negative value mean bottom up line order)
resX	[in] Image horizontal resolution (in PPI)
resY	[in]Image vertical resolution (in PPI)
fQuality	[in] Quality level for JPEG2000, he valid range is between 0 and 100

- *Returns*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

1.1.2.31)   IBSU_CombineImage

- *Description*

This function maps to IBSU_WSQDecodeFromFileW when UNICODE is defined

1.1.2.32) IBSU_CombineImageW

- *Prototype*

API DLL	Int WINAPI IBSU_CombineImage (const IBSU_ImageData inImage1, const IBSU_ImageData inImage2 ,IBSU_CombineImageWhichHand whichHand , IBSU_ImageData *ouImage)

- *Description*

    Combine two images (2 flat fingers) into a single image (left/right hands).

- *Parameters*

Parameter	Description
inImage1	[in] Pointer to IBSU_ImageData (index and middle finger)
inImage2	[in] Pointer to IBSU_ImageData (ring and little finger)
whichHand	[in] Information of left or right hand
*ouImage	[out] Pointer to IBSU_ImageData (1600 x 1500 fixed size image)

- *Returns*

Return Value	Description
0	Function completed successfully.
< 0	The error code as defined in IBScanUltimateApi_err.h

Support Contact Information:

www.integratedbiometrics.com

Integrated Biometrics, LLC

North American Office

South Korean Office

Seongnam-si, Gyeonggi-do
Republic of Korea

For Mailings & Correspondence
PO Box 170938
Spartanburg, SC 29301

Phone
+82-31-777-2207

US & Canada
(864) 990-3711
Toll-free (888) 840-8034
Extension 1 – Company Directory
Extension 2 – Technical Support
Extension 3 – Sales Support
Extension 4 – Marketing
Extension 5 – Accounting
Extension 0 – Main Line

Sales Administration
everun@ibkr.co.kr

Sales & Pricing Inquiries
sales@integratedbiometrics.com

Terms & Conditions of a Sale

Terms & Conditions for Supplier Purchases

Sales Administration
marci.bowers@integratedbiometrics.com

Technical Support
technical@integratedbiometrics.com