

BizComX.dll

SDK Library Description

July 2013

Table of Context

Table of Context	2
Introduction.....	3
Retrieving information from Business Cards	3
Functions and properties summary	4
Licensing	5
Distribution	6
Library Methods.....	6
InitBizLibrary	6
CalibrateScanner	7
Clean	8
ScanToFile.....	8
ScanToFileEx.....	9
ProcessBizCard	10
RotatImage	11
ConvertImage	13
ReformatImage	15
GetBizData.....	16
GetRawData	19
GetRawDataEx	20
GetBwlImage	20
Library Properties	22
IsNeedCalibration.....	22
IsScannerValid.....	22
PaperInTray	22
ScanHeight	23
ScanWidth.....	24
JpgQuality	24
ScannerType.....	24
ScanProg	25
IsScannerValid.....	25
PressedButton.....	25
FullName.....	26
FirstName	26
MiddleName	27
LastName.....	27
Title	27
Company.....	28
Address1st.....	28
State1st.....	28
Country1st.....	29
City1st	29
ZipCode1st.....	29
Tel1st	30
Fax1st	30

Email1st	30
WebSite1st.....	31
Appendix A – SDK installation and Registration.....	31
Installing the SDK package	31
Appendix B – Constant Definitions.....	33

Introduction

BizComX.Dll library is an ActiveX wrapper that wraps the BizCom com dll. It eases the integration between VC++ source libraries and the web script languages. There is one component in the BizComX ActiveX. It is called BizLib. It contains all the exposed methods and properties of the BizCom class.

“*BizLib*” class controls all the operations around processing the Business cards

- Controls the scanner activity and contains the last scanned image in an internal memory. It controls the scanning settings such as scan size. The scanned image is loaded to an internal memory and can be saved to an external file in a bitmap format.
- Parse and refine the textual data extracted by the OCR. The data is kept in internal variables ready to be exported to the application.

Perform internal image manipulation such as rotation, color scheme modification, dpi modification and image export to a file in various formats. This library can also be used for external image files manipulation

Retrieving information from Business Cards

Retrieving the data from a business card involves 4 steps:

- Loading the internal image. Scanning the business card does this. The size can be set to the common check size 3.6 x 2.2, but you can let the software auto detect the size by setting the Height and width to -1.
- Processing the internal image – use *ProcessBizCard* function to activate the OCR on the scanner’s internal image.
- Retrieve the data – use *GetBizData* function to retrieve the detected data
- Export the internal image to a file in the proper dpi and color scheme.

IMPORTANT:

The internal image is overwritten in every new scan. It is the application responsibility to make sure that the scan is performed in the proper setting. Failing to do so may result with bad data retrieval.

Functions and properties summary

Name	Type	Functionality
InitBizLibrary	Function	Activate and enables the library functionality
CalibrateScanner	Function	Calibrate the scanner color sensor
Clean	Function	Clean the scanner color sensor
ScanToFile	Function	Scan document to a bitmap file
ScanToFileEx	Function	Scan document to a bitmap file, while presenting a window with the progress of the scanning.
ProcessBizCard	Function	Initialize the image. Process and data extraction from the business card image file.
ConvertImage	Function	Convert image format (internal or external image)
RotateImage	Function	Rotate image clock-wise in 90,180 and 270 degrees (internal or external image)
ReformatImage	Function	Modify the image dpi, color scheme and save scheme (internal or external image)
GetBizData	Function	Get the value of one OCR field by its id.
GetRawData	Function	Get the all OCR text, as it appears on the business card.
GetRawDataEx	Function	Get the all OCR text, as it appears on the business card. This function has a return code, which indicates if it failed or not.
GetBwImage	Function	Get the image in B/W color.
IsNeedCalibration	Property	Check if the scanner needs to be calibrated
IsScannerValid	Property	Verify the scanner functionality
PaperInTray	Property	Check is a document is in the scanner tray
ScanHeight	Property	Set/retrieve the image height
ScanWidth	Property	Set/retrieve the image width
ScannerType	Property	Get the Acuant's scanner type, which is plugged to the computer
JpgQuality	Property	A number which represents the Quality of the Jpeg to be scanned.
ScanProg	Property	Get the scan program

Name	Type	Functionality
FullName	Propert	OCR Property
FirstName	Propert	OCR Property
MiddleName	Propert	OCR Property
LastName	Propert	OCR Property
Title	Propert	OCR Property
Company	Propert	OCR Property
Address1st	Propert	OCR Property
State1st	Propert	OCR Property
Country1st	Propert	OCR Property
City1st	Propert	OCR Property
ZipCode1st	Propert	OCR Property
Tel1st	Propert	OCR Property
Fax1st	Propert	OCR Property
Email1st	Propert	OCR Property
WebSite1st	Propert	OCR Property

Licensing

Each library *must* to be initialized with a license number prior to usage using the *Init()* function of each library. Failing to initialize the library will block the access to the library functionality. There are two types of licenses:

Temporary License: Using this type of license common for library evaluation. This type of license normally expires after 60 days. A temporary SDK license can be obtained from <http://id-scan.com/developer>

- **Permanent License:** Unique key that is used to activate the library with no time limit and also used to identify the customer when calling Acuant for customer support.

The software dealer supplies the permanent license key

Distribution

To install the SDK files at destination computer you just need to copy all the SDK files that are in the SDK installation folder to the destination computer.

There are some files that will need to be register on the destination computer such as COM\ActiveX objects, install these files at the end of the SDK files installations since it will need the none COM\ActiveX files to exist before register.

Here is a list of the files that need to be register:

- BizCom.dll (Com object)
- BizComX.dll (ActiveX object) mostly used for VB scripts

Note:

If you don't use the COM interface in your application and use the SDK files directly like in VC++ then you don't need to install these files on the destination computer.

Library Properties and Methods

Important: Some scanner functions may take several seconds to execute (such as **Scan** or **Clean**). While executing lengthily function one should not try to execute a second scanner function until the previous function has returned. Failing to follow this rule may cause software\hardware exception.

Library Methods

InitBizLibrary

Format

```
InitBizLibrarv (License As Strina) As Long
```

Parameters

[in] **License** – Null terminated string that holds license key value.

Return

SLIB_ERR_SCANNER_BUSSY: The scanner is still busy executing the previous scanner command.

LICENSE_VALID: License is valid and the library is ready to be used.

LICENSE_INVALID: The license is invalid. All scanner operations are disabled.

LICENSE_EXPIRED: License has expired. All scanner operations are disabled.

SLIB_ERR_DRIVER_NOT_FOUND: The scanner driver was not found. To fix this error re-install the scanner's driver. All scanner operations are disabled.

SLIB_ERR_SCANNER_NOT_FOUND: The scanner is not connected to the PC. To fix this error make sure the scanner is connected and re-starts the. All scanner operations are disabled.

Remarks

Use this function to initialize the scanner library. This function loads the scanner driver and initializes the internal image structure. This function must be called before calling any other function in the library.

CalibrateScanner

Format

```
calibrateScanner ()
```

Return value

Void.

Remarks

This function calibrates the scanner using the calibration card. The calibration results are stored in a file inside the windows directory. The operation result can be tested for good completion by reading the return value. This property may store one of the following values:

SLIB_ERR_SCANNER_BUSSY: The scanner is still busy executing the previous scanner command.

LICENSE_INVALID – Library was not initialized with proper license.

SLIB_ERR_SCANNER_NOT_FOUND – No attached scanner was found.
SLIB_ERR_INVALID_SCANNER – The attached scanner is invalid.
SLIB_FALSE – The operation failed (Mostly because no calibration card was found)
SLIB_TRUE – Operation succeeded.

Clean

Format

```
Clean ()
```

Return value

SLIB_ERR_SCANNER_BUSSY: The scanner is still busy executing the previous scanner command.

Remarks

This function clean the scanner lens by running the cleaning pad (supplied in the scanner kit) back and fourth. This function applies only to scanner modules ScanShell800/N.

ScanToFile

Format

```
ScanToFile (FileName As Strind) As Long
```

Parameters

[in] **FileName** – Null terminated string that holds the full path of the scanned image.

Return

If function succeeds, the return value is **SLIB_ERR_NONE**

If function fails, the return number is may be one of the following:

SLIB_ERR_SCANNER_BUSSY: The scanner is still busy executing the previous scanner command.

LICENSE_INVALID – Library was not initialized with proper license.

SLIB_ERR_SCANNER_NOT_FOUND – No attached scanner was found.

SLIB_ERR_SCANNER_GENERAL_FAIL

SLIB_ERR_SCANNER_NOT_FOUND

SLIB_ERR_HARDWARE_ERROR

SLIB_ERR_PAPER_FED_ERROR

SLIB_ERR_SCANABORT

SLIB_ERR_NO_PAPER

SLIB_ERR_PAPER_JAM

SLIB_ERR_FILE_IO_ERROR

SLIB_ERR_PRINTER_PORT_USED

SLIB_ERR_OUT_OF_MEMORY

Remarks

Scan document to the internal image buffer and, in the same time, export it to a bitmap file named “File Name” in the local disk under. The operation result can be tested for good completion by reading the return value.

After the scan, the internal image can be further manipulated and exported using separate commands such as:

- **Rotation** – use [RotatImage\(\)](#) to rotate the internal image in 90,180 or 270 degrees.
- **Color Scheme** – Modify the internal image color to Gray or black and white images using [ReformatImage\(\)](#).
- **Resolution** – Modify the internal image resolution to any resolution using [ReformatImage\(\)](#).
- **Saving format** – Save the internal image to external file in one of 7 popular file formats using either [ConvertImage\(\)](#) or [ReformatImage\(\)](#) or [RotatImage\(\)](#)

ScanToFileEx

Format

ScanToFileEx (**FileName** As Strind) As

Parameters

[in] **FileName** – Null terminated string that holds the full path of the scanned image.

Return

If function succeeds, the return value is **SLIB_ERR_NONE**

If function fails, the return number is may be one of the following:

SLIB_ERR_SCANNER_BUSSY: The scanner is still busy executing the previous scanner command.

LICENSE_INVALID – Library was not initialized with proper license.

SLIB_ERR_SCANNER_NOT_FOUND – No attached scanner was found.

SLIB_ERR_SCANNER_GENERAL_FAIL

SLIB_ERR_SCANNER_NOT_FOUND

SLIB_ERR_HARDWARE_ERROR

SLIB_ERR_PAPER_FED_ERROR

SLIB_ERR_SCANABORT

SLIB_ERR_NO_PAPER

SLIB_ERR_PAPER_JAM

SLIB_ERR_FILE_IO_ERROR

SLIB_ERR_PRINTER_PORT_USED

SLIB_ERR_OUT_OF_MEMORY

Remarks

Scan document to the internal image buffer and, in the same time, export it to a bitmap file named “File Name” in the local disk under. In addition, the function automatically invoke a dialog box with a scanning progress indication. The operation result can be tested for good completion by reading the return value.

After the scan, the internal image can be further manipulated and exported using separate commands such as:

- **Rotation** – use [RotatImage\(\)](#) to rotate the internal image in 90,180 or 270 degrees.
- **Color Scheme** – Modify the internal image color to Gray or black and white images using [ReformatImage\(\)](#).
- **Resolution** – Modify the internal image resolution to any resolution using [ReformatImage\(\)](#).
- **Saving format** – Save the internal image to external file in one of 7 popular file formats using either [ConvertImage\(\)](#) or [ReformatImage\(\)](#) or [RotatImage\(\)](#)

ProcessBizCard

Format

```
ProcessBizCard (fileName As String) As Long
```

Parameters

[in] **fileName** – The full path of the image. If an empty string is supplied then it will process the internal image (the image of the last scanned card).

Return

If function succeeds, the return value is **ID_TRUE**.

If function fails, the return value is one of the following:

- **LICENSE_INVALID** – Library was not initialized with proper license.
- **SLIB_ERR_SCANNER_NOT_FOUND** – No attached scanner was found.
- **SLIB_ERR_INVALID_SCANNER** – No scanner was found attached to the PC.
- **INVALID_INTERNAL_IMAGE** – No internal image is loaded. This value return when attempting to use this function without scanning an image first.

Remarks

Use this function to process the internal image acquired in the last scan. The function de skew and cleans the image and then pass to the OCR for analysis. The resultant textual data is kept in internal structure ready for retrieval by *GetBizData()* function. Processing the image does not modify the image content.

Rotatelmage

Format

```
Rotatelmage ( _
    SourceImage As String, _
    Angle As Long, _
    DestType As Long, _
    DestImage As String _
)
```

Parameters

[in] **SourceImage** – Full path name of the original image file. If this string is empty the rotation is performed on the internal image.

[in] **Angle** – The angle to rotate the original image. This value can be one of the following values:

ANGLE_0 : No rotation

ANGLE_90: 90 degrees rotation

ANGLE_180: 180 degrees rotation

ANGLE_270: 270 degrees rotation

[in] **DestType** – The destination of the rotated image. This parameter may be one of two values:

SAVE_TO_FILE : Save the image to a file. The file name should be given in *DestImage* parameter.

SAVE_TO_CLIPBOARD : Copy the rotated image the image to the clipboard.

[in] **DestImage** – Full path name of the destination file. This parameter is ignored if the parameter **DestType** is set to **SAVE_TO_CLIPBOARD**. If this value is an empty string no save will be performed.

Return

If function succeeds, the return the value **IMG_ERR_SUCCESS**.

If the function fails it returns one of the following values:

- **LICENSE_INVALID** – Library was not initialized with proper license.
- **IMG_ERR_BAD_ANGLE_0** – bad rotation parameter.
- **IMG_ERR_BAD_DESTINATION** – Bad destination parameter (the destination parameter is neither file or clipboard)
- **IMG_ERR_FILE_OPEN** – Cannot open input file. This value is returned if the *SourceImage* string is not empty but it points to a missing or invalid source image file.
- **INVALID_INTERNAL_IMAGE** – This value is returned if the *SourceImage* string is empty but no document was scanned so there is no internal image in the memory.
- **IMG_ERR_FILE_SAVE_TO_CLIPBOARD** – Cannot save image to clipboard due to an error.
- **IMG_ERR_FILE_SAVE_TO_FILE** – Cannot save destination file due to invalid destination file or disk save error.

Remarks

Use this function to rotate an image in 0, 90, 180 or 270 degrees and save it to a file in any one of seven formats. The manipulated image may be loaded from an external file (if **SourceImage** string holds a string value equal to the source image file name) or performed on the internal image buffer (if **SourceImage** string is empty). When using a file as the image source, it is important to use the proper file extension to indicate the image format. Proper extension types are: BMP, JPG, TIFF, PCX, PNG, TGA and PSD. If an image has unrecognizable extension due to an error (e.g. TIFF instead of TIF) the function refers to the file as BITMAP.

After the image is rotated it can be exported to either the clipboard or to external image file. The destination file name may be one of the seven file formats indicated above. If an image has unrecognizable extension due to an error (e.g. TIFF instead of TIF) the function exports to the file in a BITMAP format. The destination file name may be the same as the source file name. In such case the new file, resulting with a rotated image, will overwrite the original file. If no destination image file name is given (empty string) no save is done.

Do not be misled by the name of this function. This function flexibility actually allows implicitly to do the following:

- Use the following function call to convert an image file from one type to another:
Rotatelmage ("xxx.bmp", ANGLE_0, SAVE_TO_FILE, "xxx.jpg")
- Use the following function call to copy an image file to the clipboard:
Rotatelmage ("xxx.bmp", ANGLE_0, SAVE_TO_CLIPBOARD, "")
- Use the following function call to rotate the internal image :
Rotatelmage ("", ANGLE_0, SAVE_TO_FILE, "")
- Use the following function call to save the internal image to a file:
Rotatelmage ("", ANGLE_0, SAVE_TO_FILE, "xxx.bmp")

ConvertImage

Format

```
ConvertImage ( _
SourceImage As String, _
DestImage As String _
) as Long
```

Parameters

[in] **SourceImage** – Full path name of the original image file. If this string is empty the rotation is performed on the internal image.

[in] **DestImage** – Full path name of the destination file.

Return

If function succeeds, the return the value **IMG_ERR_SUCCESS**.

If the function fails it returns one of the following values:

- **LICENSE_INVALID** – Library was not initialized with proper license.
- **IMG_ERR_BAD_DESTINATION** – Bad destination parameter (the destination parameter is neither file or clipboard)
- **IMG_ERR_FILE_OPEN** – Cannot open input file. This value is returned if the **SourceImage** string is not empty but it points to a missing or invalid source image file.
- **INVALID_INTERNAL_IMAGE** – This value is returned if the **SourceImage** string is empty but no document was scanned so there is no internal image in the memory.
- **IMG_ERR_FILE_SAVE_TO_FILE** – Cannot save destination file due to invalid destination file or disk save error.

Remarks

This function is a shorter version of the function [RotateImage\(\)](#). It takes and input file (if **SourceImage** is not empty) or use the internal image as a source (if **SourceImage** is empty) and save it to a file. Using different file extension for the source and the destination converts the image type to the desired type. Image type conversion must be done carefully since some color schemes are not supported in all file types. The following table shows the available destination types and the color scheme they as capable of storing:

Destination image extension	Destination image type			
	True color (24 bit)	256 colors (8 bit)	Gray scale (8 bit)	Black and white (1 bit)
BMP	√	√	√	√
TIF	√		√	√
JPG	√		√	
PCX	√	√	√	
TGA	√	√	√	
PNG	√	√	√	
PSD	√			

Important: This table is applicable to all the functions in this library.

ReformatImage

Format

```
ReformatImage ( _
    SourceImage As String, _
    toColor As Integer, _
    toDpi As Integer, _
    DestImage As String _
```

Parameters

[in] **SourceImage** – Full path name of the original image file. If this string is empty the rotation is performed on the internal image.

[in] **toColor** – One of five values:

- **LICENSE_INVALID** – Library was not initialized with proper license.
- **IMAGE_SAME_COLOR** – no modification in the image color scheme
- **IMAGE_BW** – Convert to black and white color scheme.
- **IMAGE_GRAY_256** – convert to 256 gray scale color scheme.
- **IMAGE_COLOR_256** – convert to 256-color scheme.
- **IMAGE_COLOR_TRUE** – convert to true color scheme.

[in] **toDpi** – Set the new Image DPI. A value of 0 indicate no DPI modification.

[in] **DestImage** – Full path name of the destination file. Is this value is an empty string no save will be performed.

Return

If function succeeds, the return the value **IMG_ERR_SUCCESS**.

If the function fails it returns one of the following values:

- **IMG_ERR_BAD_COLOR** – Bad **toColor** parameter value.
- **IMG_ERR_BAD_DPI** – Bad **toDpi** parameter value.
- **IMG_ERR_FILE_OPEN** – Cannot open input file. This value is returned if the **SourceImage** string is not empty but it points to a missing or invalid source image file.
- **INVALID_INTERNAL_IMAGE** – This value is returned if the **SourceImage** string is empty but no document was scanned so there is no internal image in the memory.
- **IMG_ERR_FILE_SAVE_TO_FILE** – Cannot save destination file.

- **IMG_ERR_FILE_SAVE_TO_FILE** – Cannot save destination file due to invalid destination file or disk save error.

Remarks

Use this function to modify the image color scheme and resolution and save it to a file in any one of seven formats. The manipulated image may be loaded from an external file (if **SourceImage** string holds a string value equal to the source image file name) or performed on the internal image buffer (if **SourceImage** string is empty). When using a file as the image source, it is important to use the proper file extension to indicate the image format. Proper extension types are: BMP, JPG, TIFF, PCX, PNG, TGA and PSD. If an image has unrecognizable extension due to an error (e.g. TIFF instead of TIF) the function refers to the file as BITMAP.

Image reformat can be done either on the image color scheme or the image dpi or both. Notice that changing the image format may lose the image color information (e.g., when converting from 24 bit true color to 256 gray scale).

Modifying and image format from 256 gray scales to 24 bit true color will (obviously) not add color to the image but it will save the image with the proper RGB format (no color map) instead of or using 256 gray scale palette.

After the image is reformatted it can be exported to external image file. The destination file name may be one of the seven file formats indicated above. If the destination file name has unrecognizable extension the function exports to the file in a BITMAP format (the default format). If no destination image file name is given (empty string) no save is done.

Important: The 256 colors scheme is NOT supported for JPG and TIF files.

GetBizData

Format

```
GetBizData ( _
OcrFieldType As Long, _
index As Long, _
Value As String, _
label As String _
    \ As Long
```

Parameters

[in] **OcrFieldType** – The id number of the OCR field, which it's value is requested.

[in] **index** – The index of the OCR field in it's value's array. For example, there may be several phone numbers on the business card (zero based).

[out] **Value** – the value of the OCR field will be placed in this parameter. If no OCR processing was performed before using this method then this parameter's value will be an empty string.

[out] **label** – the label of the OCR field will be placed in this parameter. If no OCR processing was performed before using this method then this parameter's value will be an empty string.

Return

If function succeeds, the return value is **ID_TRUE**.

If function fails, the return value is negative.

Remarks

This function retrieves the value of one OCR field from the internal structure. It takes the 1st parameter and uses it as an index to identify the requested OCR field (See table below). Next it takes the index (2nd Parameter) and direct to the specific OCR field's value. It places the value in the 3rd parameter and the field's label in the 4th parameter. If the retrieve operation was failed, the function will return a negative value.

It is important, to notice that this function is relevant only after a processing of the image was done (only after the use of *ProcessBizCard*).

OCR field	ID
Telephone	0
Fax	1
Email	2
Web site	3
Full name	4
First name	5
Middle name	6
Last name	7
Prefix name	8
Name suffix	9
Company	10
Address	11

City	12
State	13
Zip	14
Country	15
Title	16

Example

The following example will use the method to retrieve and display all the addresses from the card (Written in Visual basic format):

```
....
Dim ret,          as Long
Dim sOCRVal      as String
Dim sOCRLabel    as String
Dim nIndex       as Integer

nIndex = 0
ret = MyCBiz.GetBizData (11, 0, sOCRVal, sOCRLabel)

while ((ret = BIZ_NEXT_DATA_EXIST) Or (ret = BIZ_LAST_DATA))
    MsgBox sOCRVal
    nIndex = nIndex+1
    ret = MyCBiz.GetBizData (11, nIndex, sOCRVal, sOCRLabel)
wend
.....
```

GetRawData

Format

GetRawData (**data** As String)

Parameters

[out] **data** – All the OCR data, which was retrieved from the business card.

Return value

None.

Remarks

This subroutine retrieves all the OCR text from the business card, in the same order as on the card. It is important, to notice that this function is relevant only after a processing of the image was done (only after the use of *ProcessBizCard*).

GetRawDataEx

Format

```
GetRawDataEx (data As String) As Long
```

Parameters

[out] **data** – All the OCR data, which was retrieved from the business card.

Return value

If function succeeds, the return value is **ID_TRUE**.
If function fails, the return value is negative.

Remarks

This function retrieves all the OCR text from the business card, in the same order as on the card. It places this OCR data in the 1st and only parameter. It is important, to notice that this function is relevant only after a processing of the image was done (only after the use of *ProcessBizCard*).

GetBwImage

Format

```
GetBwImage (fileName As String) As
```

Parameters

[out] **fileName** – The image full path and name.

Return value

If function succeeds, the return value is **ID_TRUE**.

If function fails, the return value is negative.

Remarks

This function saves the internal image as a B/W image file.

The image file is placed in the user's temporary folder. Its full path (including the image file name) is placed in the 1st parameters.

This function works on the internal image so scanning a card must be done before using it.

Library Properties

IsNeedCalibration

Type

Property.

Direction

Read Only.

Remarks

Retrieve if the scanner needs to be calibrated. This should be tested before every scan. Non-calibrated scanner may generate images with incorrect colors. The property returns a non zero value if the scanner needs to be calibrated and a zero value if the scanner does not need to be calibrated.

IsScannerValid

Type

Property.

Direction

Read Only.

Remarks

Detect if the scanner version is supported by current code. This property is 0 if scanner is not supported and non-zero if it is supported. If the scanner is not connected to the PC, reading this property will try to search for the scanner in the USB port. If a scanner IS connected, reading this property will communicate with the scanner and verify its validity.

PaperInTray

Type

Property.

Direction

Read only.

Remarks

Detects if a document exists in the scanner tray. This property is equal 0 if no paper detected in tray or non-zero if paper is in tray.

ScanHeight**Type**

Property.

Direction

Read / Write.

Remarks

Set/retrieve the scan height (in 1/100 inch units). Available range 200-400.

Auto-detect scan size: When using scanner model ScanShell800\ScanShell800N the scanner may also scan a document with unknown size. In this mode the scanner scans the document until it “senses” that the entire document has been fed into the scanner. Then, the black borders of the image are automatically cropped and the resulting image is returned to the calling application. To trigger this feature you must set both properties *ScanHeight* and *ScanWidth* to -1. When the scan completes these properties will reflect the detected document size in milli-inches.

Note: If you intend to use *Auto-detect scan size* make sure to set *ScanHeight* and *ScanWidth* with -1 before each scan as after the scan these properties are overwritten automatically with the recently scanned document size.

ScanWidth

Type

Property.

Direction

Read / Write.

Remarks

Set/retrieve the scan width (in 1/100 inch units). Available range 200-600.

Auto-detect scan size: When using scanner model ScanShell800\ScanShell800N the scanner may also scan a document with unknown size. In this mode the scanner scan the document until it “sense” that the entire document has be fed into the scanner. Than, the black borders of the image is automatically cropped and the result image is returned to the calling application. To trigger this feature you must set the both properties *ScanHeight* and *ScanWidth* to -1. When the scan completes these properties will reflect the detected document size in milli- inches.

Note: If you intend to use *Auto-detect scan size* make sure to set *ScanHeight* and *ScanWidth* with -1 before each scan as after the scan these properties are overwritten automatically with the recently scanned document size.

JpgQuality

Type

Property.

Direction

Read / Write.

Remarks

Get/Set the quality of the jpeg image. A number between 1 to 100.

Pay attention that this property is relevant to only a jpeg image

ScannerType

Type

Property.

Direction

Read.

Remarks

Get the id number which represents the scanner type (e.g. ScanShell800).

ScanProg**Type**

Property.

Direction

Read.

Remarks

Get the id number which represents the scanner program.

IsScannerValid**Type**

Property.

Direction

Read / Write.

Remarks

0 = one side scan, 1 = Duplex scan.

Setting this value activate the double side scan when using scanner model ScannShell3000D\ScanShell800DX\ScanShell800DXN.

PressedButton**Type**

Property.

Direction

Read.

Remarks

Get the id of the button, which was pressed on the scanner.
It will return one of the following values.

OCR field	ID
Top button	0
Bottom button	1
Middle button	2

Pay attention that this property is relevant only to the ScanShell1000 scanner.

FullName

Type

OCR Property.

Direction

Read.

Remarks

Get the value of the full name field on the business card.
The value is set automatically, after processing a scanned business card.

FirstName

Type

OCR Property.

Direction

Read.

Remarks

Get the value of the first name field on the business card.
The value is set automatically, after processing a scanned business card.

MiddleName**Type**

OCR Property.

Direction

Read.

Remarks

Get the value of the middle name field on the business card.
The value is set automatically, after processing a scanned business card.

LastName**Type**

OCR Property.

Direction

Read.

Remarks

Get the value of the last name field on the business card.
The value is set automatically, after processing a scanned business card.

Title**Type**

OCR Property.

Direction

Read.

Remarks

Get the value of the title field on the business card.
The value is set automatically, after processing a scanned business card.

Company

Type

OCR Property.

Direction

Read.

Remarks

Get the value of the company field on the business card.
The value is set automatically, after processing a scanned business card.

Address1st

Type

OCR Property.

Direction

Read.

Remarks

Get the value of the first address on the business card.
The value is set automatically, after processing a scanned business card.

State1st

Type

OCR Property.

Direction

Read.

Remarks

Get the value of the first state on the business card.
The value is set automatically, after processing a scanned business card.

Country1st**Type**

OCR Property.

Direction

Read.

Remarks

Get the value of the first country on the business card.
The value is set automatically, after processing a scanned business card.

City1st**Type**

OCR Property.

Direction

Read.

Remarks

Get the value of the first city on the business card.
The value is set automatically, after processing a scanned business card.

ZipCode1st**Type**

OCR Property.

Direction

Read.

Remarks

Get the value of the first zip code on the business card.
The value is set automatically, after processing a scanned business card.

Tel1st**Type**

OCR Property.

Direction

Read.

Remarks

Get the value of the first phone number on the business card.
The value is set automatically, after processing a scanned business card.

Fax1st**Type**

OCR Property.

Direction

Read.

Remarks

Get the value of the first fax number on the business card.
The value is set automatically, after processing a scanned business card.

Email1st**Type**

OCR Property.

Direction

Read.

Remarks

Get the value of the first email on the business card.
The value is set automatically, after processing a scanned business card.

WebSite1st

Type

OCR Property.

Direction

Read.

Remarks

Get the value of the first web address on the business card.
The value is set automatically, after processing a scanned business card.

Appendix A – SDK installation and Registration

Installing the SDK package

The Business card SDK files are packed in a single setup file (*sdk_setup.exe*). Installing the setup file extract the following file list:

File name	Destination directory	Functionality
imgForm.DLL	Windows\system32	Image processing tools collection
ImageCtrl.dll	Sdk destination directory	Image processing tools collection
BizLib.dll	Sdk destination directory	Checks image scanning & analyzer
BizCOM.dll	Sdk destination directory	COM Wrapper library to SDK
BizCOMX.dll	Sdk destination directory	ActiveX Wrapper library to SDK

dic.dll	Sdk destination directory	Dictionary
OCR_PreProc.dll	Sdk destination directory	Driver's license image analyser
SLib.dll	Sdk destination directory	Controls the scanner activity
SOCRdll.dll	Sdk destination directory	OCR library
test5.gar	Sdk destination directory	OCR library
test5.n3s	Sdk destination directory	OCR library
test5.qnp	Sdk destination directory	OCR library
test5.teh	Sdk destination directory	OCR library
TOCRR.ini	Sdk destination directory	OCR library
TOCRRdll.dll	Sdk destination directory	OCR library
TOCRRService.exe	Sdk destination directory	OCR library
UsaIDs.bin	Sdk destination directory	USA template collections
EngRotDB.bin	Sdk destination directory	Card alignment template detection
lib0.lib	Sdk destination directory	SDK library
lib1.lib	Sdk destination directory	SDK library
lib2.lib	Sdk destination directory	SDK library
ATL.DLL	Windows\system32	Part of Windows system (Installs only if newer)
MSI.DLL	Windows\system32	Part of Windows system (Installs only if newer)

Note: The libraries imgForm.dll, ImageCtrl.dll, OCR_PreProc.dll and Slib.dll are mandatory libraries and must exist in ALL sdk configurations

After the extraction the installation program register ATL.DLL (if newer) and MSI.DLL (if newer) and eventually register BizCom.dll and than register BizComX.dll

Manual registration 1:

Another method to register the BizCOM.dll library is to use the mouse right click button. To use this option you must first merge the attached OCX file 'ocxdllreg.reg' (can be found under TOOLS directory) by clicking on it and selecting YES button when asked. This will enable the option to register BizCOM.dll using the mouse when you will click on the file name in windows explorer with the mouse right button.

Manual registration 2:

Additional method to register BizCom.dll is to open a shell command prompt and in the SDK files destination directory and type the following command:

```
REGSVR32.EXE BizCOM.dll
```

Appendix B – Constant Definitions

Constant Name	Constant Value
Scanner return values	
SLIB_FALSE	0
SLIB_TRUE	1
Scanner general error types	
SLIB_ERR_NONE	1
SLIB_ERR_INVALID_SCANNER	-1
Scanning failure definition	
ERR_SCANNER_GENERAL_FAIL	-2
SLIB_ERR_CANCELED_BY_USER	-3
SLIB_ERR_SCANNER_NOT_FOUND	-4
SLIB_ERR_HARDWARE_ERROR	-5
SLIB_ERR_PAPER_FED_ERROR	-6
SLIB_ERR_SCANABORT	-7
SLIB_ERR_NO_PAPER	-8
SLIB_ERR_PAPER_JAM	-9
SLIB_ERR_FILE_IO_ERROR	-10
SLIB_ERR_PRINTER_PORT_USED	-11
SLIB_ERR_OUT_OF_MEMORY	-12
SLIB_ERR_BAD_WIDTH_PARAM	-2
SLIB_ERR_BAD_HEIGHT_PARAM	-3
SLIB_ERR_BAD_PARAM	-2
SLIB_LIBRARY_ALREADY_INITIALIZED	-13
SLIB_ERR_DRIVER_NOT_FOUND	-14
SLIB_ERR_SCANNER_BUSY	-15

Constant Name	Constant Value
Return values – Image Manipulate methods	
IMG_ERR_SUCCESS	0
IMG_ERR_FILE_OPEN	-100
IMG_ERR_BAD_ANGLE_0	-101
IMG_ERR_BAD_ANGLE_1	-102
IMG_ERR_BAD_DESTINATION	-103
IMG_ERR_FILE_SAVE_TO_FILE	-104
IMG_ERR_FILE_SAVE_TO_CLIPBOARD	-105
IMG_ERR_FILE_OPEN_FIRST	-106
IMG_ERR_FILE_OPEN_SECOND	-107
IMG_ERR_COMB_TYPE	-108
IMG_ERR_BAD_COLOR	-130
IMG_ERR_BAD_DPI	-131
INVALID_INTERNAL_IMAGE	-132
Image saving target definition	
SAVE_TO_FILE	0
SAVE_TO_CLIPBOARD	1
Image rotation angle definitions	
ANGLE_0	0
ANGLE_90	1
ANGLE_180	2
ANGLE_270	3
Image color conversion	
IMAGE_SAME_COLOR	0
IMAGE_BW	1
IMAGE_GRAY_256	2
IMAGE_COLOR_256	3
IMAGE_COLOR_TRUE	4
Resolution types	
RES_200	200
RES_300	300
RES_400	400
RES_600	600
Button definition for ScanShell1000	
TOP_BUTTON	1
BOTTOM_BUTTON	2
MIDDLE_BUTTON	3
License related constants	

Constant Name	Constant Value
LICENSE_VALID	1
LICENSE_EXPIRED	-20
LICENSE_INVALID	-21
LICENSE_DOES_NOT_MATCH_LIBRARY	-22
GENERAL_ERR_PLUG_NOT_FOUND	-200
Specific Biz library constants	
BIZ_OK	0
BIZ_FILE_NOT_FOUND	-1
BIZ_CANNOT_LOAD_FILE	-2
BIZ_NO_LINES	-3
BIZ_IMAGE_NOT_LOADED	-4
BIZ_INVALID_FIELD	-5
BIZ_NO_DATA	-6
BIZ_LAST_DATA	-7
BIZ_NEXT_DATA_EXIST	-8
Index of OCR Fields	
BIZ_FIELD_TEL	0
BIZ_FIELD_FAX	1
BIZ_FIELD_EMAIL	2
BIZ_FIELD_WEB	3
BIZ_FIELD_NAME_FULL	4
BIZ_FIELD_NAME_FIRST	5
BIZ_FIELD_NAME_MIDDLE	6
BIZ_FIELD_NAME_LAST	7
BIZ_FIELD_NAME_PREFIX	8
BIZ_FIELD_NAME_SUFFIX	9
BIZ_FIELD_COMPANY	10
BIZ_FIELD_ADDRESS	11
BIZ_FIELD_CITY	12
BIZ_FIELD_STATE	13
BIZ_FIELD_ZIP	14
BIZ_FIELD_COUNTRY	15
BIZ_FIELD_NAME_TITLE	16