

ScanW.DLL

SDK Library Description

SDK Version 9.60

January 2012

Table of Context

Table of Context	2
Introduction	4
Retrieving information from ID cards	6
Functions and properties summary	7
Licensing	8
Distribution	8
Library SlibEx: General Functionality.....	10
SlibEx Library Functions.....	10
InitLibrary	10
CalibrateScanner	12
Clean.....	12
ScanToFile	13
ScanToFileEX	13
ResetIntImage	14
SlibEx Library Properties	15
IsNeedCalibration	15
IsScannerValid	15
LastErrorStatus	15
PaperInTray	16
PressedButton	16
Resolution.....	17
ScanHeight	17
ScannerColorScheme	17
ScanWidth	18
ScannerType.....	18
Version	19
Library IdData: General Functionality.....	20
idData Library Functions	20
AutoDetectState.....	20
AutoDetectStateEx	21
InitLibrary	22
ProcState	22
GetFaceImage	23
GetFirstCountry	24
GetFirstStateByCountry	24
GetNextCountry	25
GetNextStateByCountry	25
GetSignImage	27
RefreshData.....	28
RefreshDataAu	28
State2Id	28
Id2Country	29
Id2State.....	29
StateIsSupported.....	31
GetDetectAcuracy	31
CountySupportAutoDetect	31
idData Library Properties	32
Library CBarCode: General Functionality.....	33
CBarCode Library Functions	33

InitLibrary	33
ProcImage.....	34
RefreshData.....	34
GetRawField.....	36
GetRawData.....	36
CBarCode Library Properties	37
Library CImage: Properties and Functions	38
InitLibrary	38
GetImageColor	39
RotateImage	39
ConvertImage.....	41
ReformatImage	43
ConcatenateImage	44
Library COcr: Properties and Functions	46
COcr Library Functions	46
InitLibrary	46
ExtractText	46
ExtractTextEx.....	47
COcr Library Properties	48
mText	48
Library MagLib: General Functionality	49
MagLib Library Functions.....	49
InitLibrary	49
IsReaderValid.....	50
WasCardSweaped	50
Process	51
GetRawData.....	51
MagLib Library Properties	52
Library CPassport: General Functionality.....	53
CPassport Library Functions	53
Init	53
Process	54
GetFace	54
CPassport Library Properties.....	55
Appendix A – VB Demo Program	56
Using the license key in the program	56
Applying the license key to the code	56
Appendix B – SDK installation.....	57
Installing the SDK package	57
Library SlibEx constants.....	58
Library CImage constants.....	68
Library COcr constants	69
License related constants	69
Library CBarCode constants.....	69
Library MagLib constants.....	72
Appendix D – Debugging Flags.....	73
Appendix E – Errata Information	73
Appendix F – Supported States for Detection	74

Introduction

ScanW.Dll library is a wrapper COM object that eases the integration between VC++ source libraries and VB code. The library includes four sub libraries:

SlibEx – Controls the scanner activity and contains the last scanned image in an internal memory. This library controls the scanning settings such as scan size, color scheme and resolution. The scanned image is loaded to an internal memory and can be saved to an external file in a bitmap format.

Ocr (In OCR version only) – Extract the textual data from the internal image.

IdData (In the ID version only) – Parse and refine the textual data extracted by the OCR. The data is kept in internal variables ready to be exported to the application.

Cimage – Use for 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.

Each sub library can function as individual library regardless to the other libraries.

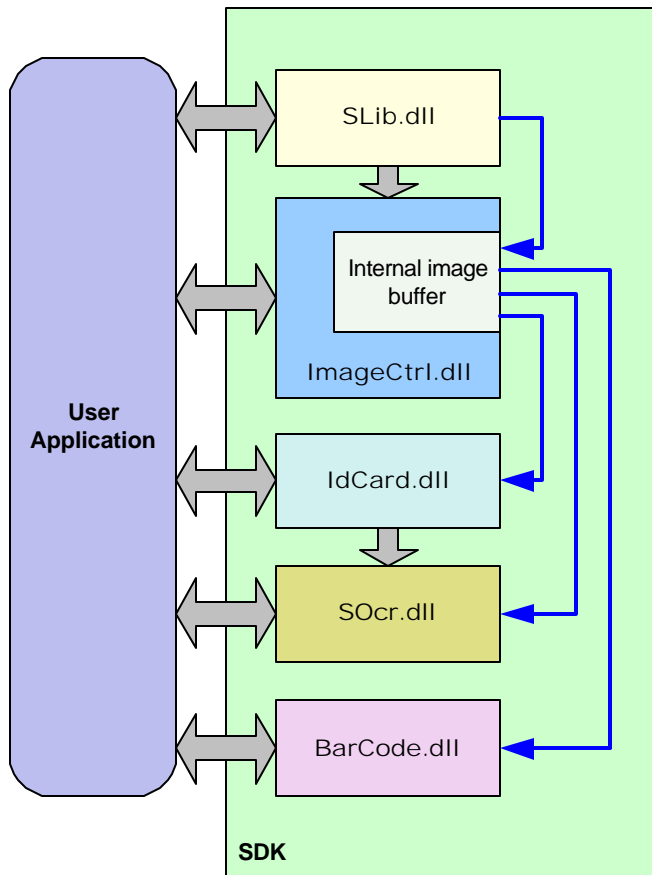


Figure1 : ScanW library internal architecture

Retrieving information from ID cards

Retrieving the data from an id card involves 4 steps:

Loading the internal image. Scanning the ID card in True color and 300 dpi, 2.2 x 3.6 inch does this.

Processing the internal image – use ProcState to activate the OCR on the scanner's internal image.

Retrieve the data – use RefreshData 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 (True color, 300 dpi, 2.2 x 3.6 inch). Failing to do so may result with bad data retrieval.

Functions and properties summary

Library: SlibEx

Name	Type	Functionality
InitLibrary	Function	Activate and enables the library functionality
CalibrateScanner	Function	Calibrate the scanner color sensor
ScanToFile	Function	Scan document to a bitmap file
IsNeedCalibration	Property	Check if the scanner needs to be calibrated
IsScannerValid	Property	Verify the scanner functionality
LastErrorStatus	Property	Retrieve the last operation error status
PaperInTray	Property	Check is a document is in the scanner tray
Resolution	Property	Set/Retrieve the scanning resolution
ScanHeight	Property	Set/retrieve the image height
ScannerColorScheme	Property	Set/retrieve the scanner to color/gray/bw color scheme
ScanWidth	Property	Set/retrieve the image width

Library: Idata

Name	Type	Functionality
ProcState	Function	Initialize the image process and data extraction from the id image file
RefreshDate	Function	Update the data collected by ProcState to the internal variables
Name	Property	
Address	Property	
City	Property	
State	Property	
Zip	Property	
DateOfBirth	Property	
ExpirationDate	Property	
IssueDate	Property	
ID	Property	
License	Property	
Class	Property	
Eyes	Property	
Hair	Property	
Height	Property	
Sex	Property	
Weight	Property	
Duplicate/Exam	Property	
County	Property	
CSC	Property	
Restriction	Property	
Type	Property	
Endorsements	Property	
SigNum	Property	
Original Date	Property	

Library: CImage

Name	Type	Functionality
ConatenatImage	Function	Concatenate two images into a single image file
ConvertImage	Function	Convert image format (internal or external image)
Rotateimage	Function	Rotate image clock-wise in 90,180and 270 degrees (internal or external image)
ReformatImage	Function	Modify the image dpi, color scheme and save scheme (internal or external image)

Library: COcr

Name	Type	Functionality
ExtractText	Function	Extract text bulks from an image file.
mText	Property	A buffer that contains the extracted text data

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 Card Scanning Solutions 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:

ScanW.dll (Com object)

ScanWEx.dll (Com object)

ScanX.dll (ActiveX object) mostly used for VB scripts

Note:



6167 Bristol Parkway Suite 330, Culver City, CA 90230
Phone: 213-867-2625, 213-867-2630 Fax: 419-735-2419

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

Library SlibEx: General Functionality

SlibEx library is used to scan documents and load their image to the internal image container. The library also saves the scanned image to an external bitmap file. The library sets and retrieves the scanners properties (such as scanning size, resolution and color scheme). The validity of all operations on properties or functions is reflected on the property [LastErrorStatus](#). This property automatically reset upon reading.

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.

SlibEx Library Functions

InitLibrary

Format

```
InitLibrary (License As String) 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.

IMPORTANT: When the application unload is should call the function **UnInit** to unload the SDK and release its memory (See the function description as section 1.1.34 on ScanWex.pdf)



6167 Bristol Parkway Suite 330, Culver City, CA 90230
Phone: 213-867-2625, 213-867-2630 Fax: 419-735-2419

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 [LastErrorStatus](#) property. 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 String) 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 [LastErrorStatus](#) property.

Notice that it is important to scan the image in True color and 300 dpi for OCR recognition.

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

Rotation – use [RotateImage\(\)](#) 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 [RotateImage\(\)](#)

ScanToFileEX

Format

```
ScanToFileEx (FileName As String) 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

This function is the same as ScasnToFile only this function will display a progress bar that will show the scanning progress.

ResetIntImage

Format

```
ResetIntImage ()
```

Remarks

Delete the internal image buffer.

SlibEx 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.

LastErrorStatus

Type

Property.

Direction

Read Only.

Remarks

Retrieve the preceding property/command setup. This property is equal **ELIB_ERR_NONE** if no error happened in the previous command execution. This property clears itself automatically to **ELIB_ERR_NONE** after read.

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.

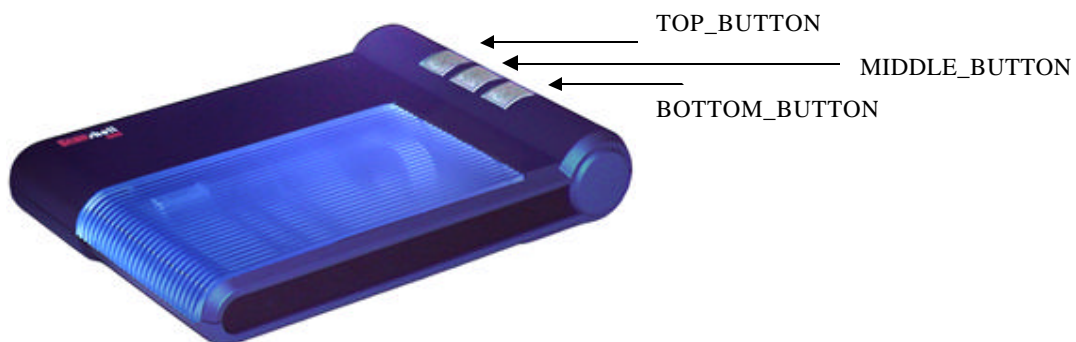
PressedButton

Type
Property.

Direction
Read only.

Remarks
Returns the button number that was pressed (valid for only when using ScanShell1000 scanner model). Read this value after reading the property *PaperInTray* (that indicate that one of the three buttons of the scanner was pressed) to retrieve which button was pressed. The return value can be one of the following:
TOP_BUTTON
MIDDLE_BUTTON
BOTTOM_BUTTON

These values correspond to the buttons shown in the following figure.



Resolution

Type
Property.

Direction
Read / Write.

Remarks

Set/retrieve the scanner resolution settings. Resolution value can be any integer in the range 50-600 (for 50-600 dpi). Trying to set value outside this will be rejected and the previous value will be used.

This property set the scanned image resolution. The internal image resolution can be modified after the scan using [ReformatImage\(\)](#) function.

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 ScanShell800xx\2000xx\3000xx the scanner may also scan a document with un-known 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.

ScannerColorScheme

Type
Property.

Direction
Read / Write.

Remarks

Set/retrieve the scanner color scheme. Available values are:

BW
GRAY
TRUECOLOR

This property set the scanned image color scheme. The internal image color scheme can be modified after the scan using *ReformatImage()* function.

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 ScanShell800xx\2000xx\3000xx the scanner may also scan a document with un-known 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.

ScannerType

Type

Property.

Direction

Read only.

Remarks

Retrieve the scanner type.

Scanner type can be one of these values:

0: No scanner
1: ScanShell 600
2: ScanShell 800
3: ScanShell 800N

- 4: ScanShell 1000
- 5: ScanShell 2000
- 6: ScanShell 2000N
- 7: ScanShell 800E
- 8: ScanShell 800EN
- 9: ScanShell 3000
- 10: ScanShell 4000
- 11: ScanShell 800G
- 12: ScanShell 5000
- 13: SnapShell (IDR)
- 14: ScanShell 800DX
- 15: ScanShell 800DXN
- 16: SnapShell (FDA)
- 17: SnapShell (WMD)
- 18: SnapShell (TWN)

Version

Type

Property.

Direction

Read Only.

Remarks

Retrieve the SlibEx version.

Library IdData: General Functionality

IdData library is where most of the image processing and data extraction is being handled. The library fetches the internal image (the last scanned image), process its graphic information and activates the OCR. The resultant text is kept in internal data structure ready to be retrieved by the application.

Notice: The library processes only the last image that was recently scanned. The library does not support external image file processing.

idData Library Functions

AutoDetectState

Format

AutoDetectState ([Reserved](#) As String) As Long

Parameters

[in] [Reserved](#) – Null terminated empty string - reserved. This parameter is not used.

Return

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

ID_ERR_USA_TEMPLATES_NOT_FOUND: The template database file for the USA states (*UsaIds.bin*) is missing. The file should be located in the SDK files location.

INVALID_INTERNAL_IMAGE – No internal image is loaded. This value return when attempting to use this function without scanning an image first.

ID_ERR_STATE_NOT_SUPPORTED : The license image doe not match any state template.

ID_ERR_STATE_NOT_RECOGNIZED : The license image doe not match any state template.

If non of the above error values is returned, the function return the state id value.

Remarks

Use this function to automatically detect the state type according to the image. If the function returns with non of the above error values t hen the return value is the state id. This value can be assigned to the input parameter *IdState* in the function [ProcState](#) for data extraction.

AutoDetectStateEx

Format

AutoDetectStateEx ([Reserved](#) As String, [angle](#) As Long) As Long

Parameters

[in] [Reserved](#) –Null terminated empty string - reserved. This parameter is not used.

[out] [angle](#) – Return the amount of clockwise 90 degrees turns that the image was rotated to be aligned horizontally.

Return

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

ID_ERR_USA_TEMPLATES_NOT_FOUND: The template database file for the USA states (*UsaIds.bin*) is missing. The file should be located in the SDK files location.

INVALID_INTERNAL_IMAGE – No internal image is loaded. This value return when attempting to use this function without scanning an image first.

ID_ERR_STATE_NOT_SUPPORTED : The license image doe not match any state template.

ID_ERR_STATE_NOT_RECOGNIZED : The license image doe not match any state template.

If non of the above error values is returned, the function return the state id value.

Remarks

Use this function to automatically align the internal image horizontally and then to detect the. If the function returns with non of the above error values then the return value is the state id. This value can be assigned to the input parameter *IdState* in the function [ProcState](#) for data extraction. The function loads in the parameter *angle* one of the following values:

ANGLE_0: The image was received in the proper alignment.

ANGLE_90: The image was rotated once in 90 degrees (clockwise).

ANGLE_180: The image was rotated twice in 90 degrees (clockwise).

ANGLE_270: The image was rotated three times in 90 degrees (clockwise).

You can dump the rotated image from the internal buffer to a file using the function [RotateImage](#)

InitLibrary

Format

```
InitLibrary (License As String) As Long
```

Parameters

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

Return

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.

LICENSE_DOES_NOT_MATCH_LIBRARY: The license is invalid for this library. All library operations are disabled.

GENERAL_ERR_PLUG_NOT_FOUND: This error returns if the attached scanner is not one of the following scanners:

ScanShell 600

ScanShell 800

ScanShell 1000

SLIB_LIBRARY_ALREADY_INITIALIZED: The *InitLibrary* function call is ignored since the library is already loaded.

Remarks

Use this function to initialize the idData library. This function must be called before calling any other function in the library.

ProcState

Format

```
ProcState (Reserved As String, idState As Long) As Long
```

Parameters

[in] **Reserved** – Null terminated empty string - reserved. This parameter is not used.

[in] **idState** – Constant numeric value of the processed state id.

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.
LICENSE_INVALID – The library was not initialized with the proper license.
SLIB_ERR_INVALID_SCANNER – No scanner was found attached to the PC.
ID_ERR_STATE_NOT_SUPPORTED – The requested state id is not supported.
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 [RefreshData\(\)](#) function. Processing the image does not modify the image content.

Successful image processing depends on the following:

The image must be scanned in 24 bit (true color) and 300 dpi.

The image must be aligned horizontally with tolerance of no more than ± 10 degrees.

GetFaceImage

Format

```
GetFaceImage (SourceFileName As String, DestFileName As String, stateId As Integer) As Long
```

Parameters

[in] **SourceFileName** – Null terminated string that holds the full path of the scanned id image. If this string is empty the internal image is used as image source.

[in] **DestFileName** – Null terminated string that holds the full name of the destination image file that will contain the face image from the id document.

[in] **stateId** – The state index value as defined in idLibDef.bas file.

Return

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

If the function fails one of the following values is returned:

LICENSE_INVALID – Library was not initialized with proper license.

ID_ERR_FILE_OPEN – **Failing to load source image (if SourceFileName is not empty)**

INVALID_INTERNAL_IMAGE – No internal image is loaded. This value return when attempting to use the internal image without scanning an image first.

ID_ERR_STATE_NOT_SUPPORTED – The requested state is not supported.

ID_BAD_DESTINATION_FILE – Bad destination path (could not create destination file).

ID_ERR_FILE_OPEN – Bad source image file (used only when using a file as the source image).

This value return if the source file is missing or cannot be accessed for reading.

INVALID_INTERNAL_IMAGE– Bad internal image (used only when extracting the face image from the image stored in the internal buffer). This value returned if there is no image in the buffer.

ID_FALSE– Internal processing error.

ID_ERR_FACE_IMAGE_NOT_FOUND– Returned when the analyzer cannot detect the face image in the driver's license image.

ID_ERR_CANNOT_DELETE_DESTINATION_IMAGE– Returned when a file with the same name as the destination file already exist and cannot be overwritten.

ID_ERR_CANNOT_COPY_TO_DESTINATION– Returned when the destination file cannot be opened for write on the disk.

Remarks

Use this function to extract the image rectangle of the person’s face from the source id image. The source image can be one of two:

Internal image: The last scanned image (stored in the internal memory). This image will be used only if the **SourceFileName** string is empty

External image file: The full file name is given in **SourceFileName** parameter. If an external file is used as the source image it must be a 24 bit image (true color) and have a resolution of 300dpi. The source and file destination can be one of the following formats: BMP, TIFF, JPG, PCX, TGA, PNG, PSD.

To set the image format use the proper file extension (*xxx.bmp* for bitmap, *xxx.jpg* for Jpeg, etc.)

GetFirstCountry

Format

```
GetFirstCountry( ) As Long
```

Return value

The constant value of the first country in the countries list.

Remarks

Use this function to get the first country in the countries list. Combining this function with the function *GetNextCountry* allow you to obtain the constant values of all supported countries. The countries constant values in the Region ARE NOT always consecutive and should be obtained using the function *GetNextCountry*.

GetFirstStateByCountry

Format

```
GetFirstStateByCountry (country As Integer) As Long
```


Parameters

[in] **country** – Constant value of the country

Return

ID_ERR_NO_MATCH: Bad country constant.

Otherwise, the function return the constant value of the first state in the country.

Remarks

Use this function to retrieve the first state constant in the country.

The states constant values in the country ARE NOT always consecutive

And should be obtained using the function *GetNextStateByCountry*.

GetNextCountry

Format

```
GetNextCountry() As Long
```

Return value

ID_ERR_COUNTRY_NOT_INIT : Returned if *GetFirstCntry* function was not called before..

ID_ERR_NO_MATCH:Returned if list has internal error.

ID_ERR_NO_NEXT_COUNTRY – Returned if this country is the last country in the list.

Return the next country constant.

Remarks

Use this function to obtain the next country in the country list. To obtain the countries list call *GetFirstCountry* once to obtain the first country. Then countiuously call *GetNextCountry* until the value **ID_ERR_NO_NEXT_COUNTRY** is returned.

GetNextStateByCountry

Format

```
GetNextStateByCountry (country As Integer) As Long
```

Parameters

[in] **country** – Constant value of the country

Return value

ID_ERR_NO_MATCH: Bad country constant.

ID_ERR_COUNTRY_NOT_INIT : Returned if *GetFirstCountry* function was not called before.

ID_ERR_NO_NEXT_STATE – Returned if this state is the last state of the country state list .

Return the next state constant.

Remarks

Use this function to obtain the next state in the list.

GetSignImage

Format

```
GetSignImage (SourceFileName As String, DestFileName As String, stateId As Integer) As Long
```

Parameters

[in] **SourceFileName** – Null terminated string that holds the full path of the scanned id image. If this string is empty the internal image is used as image source.

[in] **DestFileName** – Null terminated string that holds the full name of the destination image file that will contain the signature image from the id document.

[in] **stateId** – The state index value as defined in idLibDef.bas file.

Return

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

If the function fails one of the following values is returned:

LICENSE_INVALID – Library was not initialized with proper license.

ID_ERR_FILE_OPEN – Failing to load source image (if SourceFileName is not empty)

INVALID_INTERNAL_IMAGE – No internal image is loaded. This value return when attempting to use the internal image without scanning an image first.

ID_ERR_STATE_NOT_SUPORTED – The requested state is not supported.

ID_BAD_DESTINATION_FILE – Bad destination path (could not create destination file).

ID_ERR_FILE_OPEN – Bad source image file (used only when using a file as the source image).

This value return if the source file is missing or cannot be accessed for reading.

INVALID_INTERNAL_IMAGE– Bad internal image (used only when extracting the signature image from the image stored in the internal buffer). This value returned if there is no image in the buffer.

ID_FALSE– Internal processing error.

ID_ERR_FACE_IMAGE_NOT_FOUND– Returned when the analyzer cannot detect the face image in the driver's license image.

ID_ERR_CANNOT_DELETE_DESTINATION_IMAGE– Returned when a file with the same name as the destination file already exist and cannot be overwritten.

ID_ERR_CANNOT_COPY_TO_DESTONATION– Returned when the destination file cannot be opened for write on the disk.

Remarks

Use this function to extract the image rectangle of the person's signature from the source id image. Signature extraction is relatively new feature and is not implemented yet for all the state templates (For supported states see [Appendix F](#)). The source image can be one of two:

Internal image: The last scanned image (stored in the internal memory). This image will be used only if the **SourceFileName** string is empty

External image file: The full file name is given in **SourceFileName** parameter. If an external file is used as the source image it must be a 24 bit image (true color) and have a resolution of 300dpi. The source and file destination can be one of the following formats: BMP, TIFF, JPG, PCX, TGA, PNG, PSD.

To set the image format use the proper file extension (*xxx.bmp* for bitmap, *xxx.jpg* for Jpeg, etc.)

RefreshData

Format

```
RefreshData () As Long
```

Return Value

If the function returns non-zero value, the data was retrieved successfully.
If the function returns zero value, the data was retrieved un-successfully.

Remarks

This function is used to load library properties with the recently detected data. This function should normally be called after the execution of ProcState function.

RefreshDataAu

Type

Function.

Format

```
RefreshDataAu () As Long
```

Return Value

If the function returns non-zero value, the data was retrieved successfully.
If the function returns zero value, the data was retrieved un-successfully.

Remarks

This function is identical to [RefreshData\(\)](#) function.

State2Id

Format

```
State2Id (StateName As String) As Long
```

Parameters

[in] **StateName** – A string that holds the state name.

Return Value

The state ID if the **StateName** is recognizable state name.

ID_ERR_NO_MATCH – if the string contain unrecognizable state name.

Remarks

This helper function is used to convert a state name (string) to its equivalent id value. For example, passing StateName loaded with the string “California” will return the value 4. The function converts the string to the id even if the state is not supported by the library.

See also: [Id2State\(\)](#), LastStateIndex

Id2Country

Format

```
Id2Country (countryId As Integer, CountryName As String) As Long
```

Parameters

[in] **countryId** – An integer that holds the country id value.

[out] **CountryName** – A string that will be loaded by the function with the country name.

Return Value

ID_TRUE: The function succeeded.

ID_ERR_NO_MATCH – No country with such id was found.

Remarks

This helper function is used to convert a country id to the country name (string).

Id2State

Format

```
Id2State (stateId As Integer, StateName As String) As Long
```

Parameters

[in] **stateId** – An integer that holds the state id value.

[out] **StateName** – A string that will be loaded by the function with the state name.

Return Value

ID_TRUE: The function succeeded.

ID_ERR_NO_MATCH – No state with such id was found.

Remarks

This helper function is used to convert a state id to the state name (string). For example, passing **stateId** loaded with the value 4 loads the string **StateName** with the string “California”. The function converts the string to the id even if the state is not supported by the library.



6167 Bristol Parkway Suite 330, Culver City, CA 90230
Phone: 213-867-2625, 213-867-2630 Fax: 419-735-2419

See also: [State2Id\(\)](#), LastStateIndex

StateIsSupported

Format

```
StateIsSupported (stateId As Integer) As Long
```

Return Value

LICENSE_INVALID – Library was not initialized with proper license.

ID_TRUE: The state is supported and can be detected by the library.

ID_ERR_STATE_NOT_SUPPORTED – the library does not support the state.

Remarks

This function is used to inform the application if the id image of a given state can be processed. Use this function to detect what states the library currently supports.

GetDetectAccuracy

Format

```
GetDetectAccuracy () As Long
```

Return Value

The function returns a number in the range 0-100 that estimates the detection accuracy percentage.

Remarks

This helper function checks the detected data values and returns a value that indicates the detection accuracy (in percents). This function, combined with [RotateImage\(\)](#) function is useful when implementing auto-orientation feature in the application. This feature can be implemented by rotating the internal image and for each new position executing [ProcState\(\)](#) function. The operation success can be evaluated using [GetDetectAccuracy\(\)](#) which allow us to determine if the image is in the right orientation or a new rotation-detection-inspection is needed.

CountySupportAutoDetect

Format

```
CountySupportAutoDetect (countryId As Integer) As Long
```

Parameters

[in] [countryId](#) – countryId – Constant value of the country.

Return Value

- ID_TRUE:** The country support state auto detection.
- ID_FALSE** The country does not support state auto detection.

Remarks

The state auto-detection feature is not implemented on all the supported countries. Use this function to determine which countries can use the *AutoDetectState* function.

idData Library Properties

idData library store the detected textual information in an internal data structure. This structure is refreshed in each [RefreshData\(\)](#) function call. Some of the properties are relevant to some states and some are irrelevant. For example, the property Weight is relevant to id cards of states such as Arizona, California and Georgia but it does not exist in states such as Arkansas or Michigan. If a property is irrelevant to a particular state, or if the field is not detected, it clears to an empty string. Loading the properties is done using a simple string assignment. The *Name* field is parsed and generates four additional fields (First name, Middle name, last name and name suffix).

The supported properties are:

Name (as shown on card)	NameFirst	NameMiddle
NameLast	NameSuffix	ID
License	IssueDate	License
Address	ExperationDate	CSC
City	Eyes	Dup_Test
State	Hair	Endorsements
Zip	Height	Fee
County	Class	Restriction
DateOfBirth	Sex	SigNum
Type	Weight	Address2
Address3	Address4	Address5
Text1	Text2	Text3
Side		

An additional property is LastStateIndex. It is useful when iterating in a loop between the entire states id from the first (index 0) to the last one.

Library CBarCode: General Functionality

CBarCode library functionality is similar to idData library. It extract the data from 2D, PDF417 type bar code image. The library fetches the internal image (the last scanned image), process its graphic information and activates its internal image analyzer. The resultant text is kept in internal data structure ready to be retrieved by the application.

Notice: The library processes only the last image that was recently scanned. The library does not support external image file processing.

NOTE: For best results for 2D barcode we require 600 DPI color image (RGB).

CBarCode Library Functions

InitLibrary

Format

```
InitLibrary (License As String) As Long
```

Parameters

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

Return

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.

LICENSE_DOES_NOT_MATCH_LIBRARY: The license is invalid for this library. All library operations are disabled.

GENERAL_ERR_PLUG_NOT_FOUND: This error returns if the attached scanner is not one of the following scanners:

ScanShell 600

ScanShell 800

ScanShell 1000

SLIB_LIBRARY_ALREADY_INITIALIZED: The *InitLibrary* function call is ignored since the library is already loaded.

Remarks

Use this function to initialize the CBarCode library. This function must be called before calling any other function in the library.

ProcImage

Format

ProcImage (**Reserved** As String) As Long

Parameters

[in] **Reserved** –Null terminated empty string - reserved. This parameter is not used.

Return

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

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.

LICENSE_INVALID – The library was not initialized with the proper license.

SLIB_ERR_INVALID_SCANNER – No scanner was found attached to the PC.

ID_ERR_STATE_NOT_SUPPORTED – The requested state id is not supported.

INVALID_INTERNAL_IMAGE – No internal image is loaded. This value return when attempting to use this function without scanning an image first.

BC_ERR_NO_BC_FOUND – No bar code pattern (PDF417) was found on the image.

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 image analyzer for data extraction. The resultant textual data is kept in internal structure ready for retrieval by [RefreshData\(\)](#) function. Processing the image does not modify the image content.

Successful image processing depends on the following:

Bar code must be scanned at 600 DPI color image (RGB).

The image must be aligned in such way that the bar code image is vertical or horizontal with tolerance of no more than ± 10 degrees.

RefreshData

Format

RefreshData () As Long

Return Value

If the function returns non-zero value, the data was retrieved successfully.

If the function returns zero value, the data was retrieved un-successfully.

Remarks

This function is used to load library properties with the recently detected data. This function should normally be called after the execution of ProcImage function.

GetRawField

Format

```
GetRawField ( Index As Integer, Value As String) As Long
```

Parameters

- [in] **Index** – An Integer that holds the value of the requested field.
- [out] **Value** – A string that holds the returned value indicated by *Index*.

Return

If function succeeds, the return value is **BC_ERR_NONE**.
If function fails, the return value is **BC_ERR_BAD_PARAM** to indicate that the value passes by *Index* is not a valid field index.

Remarks

AAMVA National standard defines 51-field type that might be encoded in the PDF417 bar code image. Each one of the fields may be retrieved from the image using the field index specified in section 1.1.43. If the field is populated its value is retrieved and copied to *Value*. If the field is empty (or not implemented on the image) the parameter *Value* is set to be empty string. The maximum field size may be up to 60 characters long. Its important to understand that the fields are populated differently in different states. For example, The field NAME is populated in field index **BCF_NAME** for Alabama driver license, and the same field is populated in field index **BCF_DRIVER_LAST_NAME** for Delaware driver license. The raw field range from **BCF_NAME (0)** to **BCF_AKA_PREFIX (50)**

In addition to the raw fields, the CBarCode offers an additional set of fields (range from **BCF_EMUL_FULL_NAME** to **BCF_EMUL_SSN**). These fields are generated after the raw fields are loaded. The fields uniform the data extraction by copying the data from the raw fields and diverting it while considering its originated state type. This removes the developer the burden of knowing the state type and the way the data is handled in this particular state.

GetRawData

Format

```
GetRawData (Buffer As String)
```

Parameters

- [out] **Buffer** – A string that holds the returned value..

Return

No return value.

Remarks

This function returns the extracted text data in one bulk as detected by the bar code analyzer. The data is returned, "as is" with no parsing or additional processing. This function is useful for retrieving data from general purpose bar code images.

CBarcode Library Properties

CBarcode library store the detected textual information in an internal data structure. This structure is refreshed in each [RefreshData\(\)](#) function call. All the properties are copied from the relevant raw fields and can be retrieved directly by using GetRawFiled function while passing the proper BFC_EMUL_XXX index. (name, last name and name suffix).

The supported properties are:

Name (as shown on card)	NameFirst	NameMiddle
NameLast	NameSuffix	DateOfBirth
License	IssueDate	ExperationDate
Address	City	State
Zip	SSN	

Library CImage: Properties and Functions

CImage library is a collection of graphic functions, capable of manipulating an image object. The image object may be loaded from external file or the image object stored in ScanLib library (which is the image of the last scanned document).

The library functions are capable of doing the following:

Image rotation: Rotating an image in 90, 180 or 270 degrees.

Resolution modification: Modify the resolution to any value.

Image color conversion: Converting the image to 24 bit (true color), 256 colors (gray or color) or Black and White (1 bit).

Concatenate two image files to a single image: Attach two images (horizontally or vertically) to form a single image file of both id card sides.

The image can be exported (saved) to external image file in any one of seven popular image formats such as BMP, JPG, TIFF, PCX, PNG, TGA and PSD. Alternatively, the image object can be exported to the clipboard and from there be imported to other applications.

InitLibrary

Format

```
InitLibrary (License As String) As Long
```

Parameters

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

Return

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.

LICENSE_DOES_NOT_MATCH_LIBRARY: The license is invalid for this library. All library operations are disabled.

Remarks

Use this function to initialize the CImage library. This function must be called before calling any other function in the library.

GetImageColor

Format

```
GetImageColor ( FileName As Integer) As Long
```

Parameters

[in] **FileName** – Image file name or empty string if evaluating the internal image.

Return

IMG_ERR_FILE_OPEN: Cannot open input image file.

INVALID_INTERNAL_IMAGE: Internal image is invalid and cannot be analyzed.

IMAGE_BW – The image has Black and White colors (1 bit image).

IMAGE_GRAY_256 - The image has 256 colors of gray (8 bit image).

IMAGE_COLOR_256 - The image has 256 colors (8 bit image).

IMAGE_COLOR_TRUE - The image has 16 million (24 bit image).

Remarks

Use this function to obtain the image color scheme.

RotateImage

Format

```
RotateImage ( _  
    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 : 0 degrees 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:
RotateImage (“xxx.bmp”, ANGLE_0, SAVE_TO_FILE, “xxx.jpg”)

Use the following function call to copy an image file to the clipboard:
RotateImage (“xxx.bmp”, ANGLE_0, SAVE_TO_CLIPBOARD, “”)

Use the following function call to rotate the internal image :
RotateImage (“”, ANGLE_0, SAVE_TO_FILE, “”)

Use the following function call to save the internal image to a file:
RotateImage (“”, ANGLE_0, SAVE_TO_FILE, “xxx.bmp”)

ConvertImage

Format

```
ConvertImage ( _
    SourceImage As String, _
    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] **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 an 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 are 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	v	v	v	v
TIF	v		v	v
JPG	v		v	
PCX	v	v	v	
TGA	v	v	v	
PNG	v	v	v	
PSD	v			



6167 Bristol Parkway Suite 330, Culver City, CA 90230
Phone: 213-867-2625, 213-867-2630 Fax: 419-735-2419

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.

ConcatenateImage

Type

Function.

Format

```
ConcatenateImage( _  
    SourceImage0 As String, _  
    Angle0 As Long, _  
    SourceImage1 As String, _  
    Angle1 As Long, _  
    CombType As Long, _  
    DestType As Long, _  
    DestImage As String _  
)
```

Parameters

- [in] **SourceImage0** – Full path name of the first image.
- [in] **Angle0** – The angle to rotate SourceImage0 before the combination.
- [in] **SourceImage1** – Full path name of the second image.
- [in] **Angle1** – The angle to rotate SourceImage0 before the combination.
- [in] **CombType** – The location of the images in the result image file:

IMAGE_COMB_HORIZONTAL – SourceImage0 is located to the left of SourceImage1
IMAGE_COMB_VERTICAL - SourceImage0 is located above SourceImage1.

- [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**.

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 for Image 0.

IMG_ERR_BAD_ANGLE_1 – Bad rotation parameter for Image 1.

IMG_ERR_FILE_OPEN_FIRST – Cannot open **SourceImage0** file.

IMG_ERR_FILE_OPEN_SECOND – Cannot open **SourceImage1** file.

IMG_ERR_BAD_DESTINATION – Bad destination parameter (the destination is neither file or clipboard)

IMG_ERR_COMB_TYPE – Bad **CombType** value.

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 a bad destination path or disk error.

Remarks

Use this function to combine two image files into a single image file. The function works in the following sequence:

Imports **SourceImage0** to an image object 0.

Rotate image object0 by **Angle0**

Imports **SourceImage1** to an image object 1.

Rotate image object1 by **Angle1**

Combine Image0 and Image1 one on top of each other (if **CombType** is equal

IMAGE_COMB_VERTICAL) or one to the left of the other (if **CombType** is equal

IMAGE_COMB_HORIZONTAL).

Save the result image to an external file or to the clipboard.

Notice:

This function can work only on image files and not on the internal image.

Library COcr: Properties and Functions

COcr provides basic text extraction from an image file. The image file format must have a resolution of 300 dpi. The image may be in either color or black and white color scheme.

COcr Library Functions

InitLibrary

Format

```
InitLibrary (License As String) As Long
```

Parameters

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

Return

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.

LICENSE_DOES_NOT_MATCH_LIBRARY: The license is invalid for this library. All library operations are disabled.

GENERAL_ERR_PLUG_NOT_FOUND: This error returns if the attached scanner is not one of the following scanners:

ScanShell 600

ScanShell 800

ScanShell 1000

Remarks

Use this function to initialize the COcr library. This function must be called before calling any other function in the library.

ExtractText

Type

Function.

Format

```
ExtractText ( _  
    SourceImage As String, _  
    TextType As Long, _  
)
```

Parameters

[in] **SourceImage** – Full path name of the original image.

[in] **TextType** – Instruct the OCR what type of data is written in the image. This value increases the detection accuracy and speeds the OCR operation. This value can be one of the following values:

USE_ALPHANUM : The image contains alphanumeric characters.

USE_ALPHA_CAPS_ONLY : The image contains capital letters only.

USED_NUM_ONLY : The image contains numbers only

Return

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

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

LICENSE_INVALID – Library was not initialized with proper license.

TOCRJOBERROR – The OCR engine was not able to accomplish the detection process correctly.

TOCR_BAD_TYPE– Bad **TextType** value.

Remarks

Use this function to extract text bulks from an image. The text size is limited to 4K (4096) characters.

ExtractTextEx

Type

Function.

Format

```
ExtractTextEx ( SourceImage As String )
```

Parameters

[in] **SourceImage** – Full path name of the original image.

Return

If function succeeds, the return the value **TOCR_OK (=0)**.

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

LICENSE_INVALID – Library was not initialized with proper license.

TOCRJOBERROR – The OCR engine was not able to accomplish the detection process correctly.

TOCR_BAD_TYPE– Bad **TextType** value.

Remarks

Use this function to extract text bulks from an image. The text size is limited to 4K (4096) characters. This function process the image file in a different method than *ExtractText* function does; this result in a longer processing time but with higher accuracy. This function is recommended for use with complex document structure where processing time is not critical.

COcr Library Properties

mText

Type

Property.

Direction

Read Only.

Remarks

Contains the extracted text. This property is loaded after the execution of *ExtractText()* function.

Library MagLib: General Functionality

MagLib controls the magnetic reader, collect and analyses its data once a card was swiped. The library scans COM1-COM16 for the existence of the magnetic reader and initializes it. Once a magnetic card is swiped, the data is parsed and refresh the relevant properties of the library. The library automatically detects the data format and parses it. The following driver license formats are supported:

AAMVA
Old DMV (California)
Old DMV (Louisiana).

MagLib Library Functions

InitLibrary

Format

```
InitLibrary (License As String) As Long
```

Parameters

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

Return

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.

LICENSE_DOES_NOT_MATCH_LIBRARY: The license is invalid for this library. All library operations are disabled.

GENERAL_ERR_PLUG_NOT_FOUND: This error returns if the attached scanner is not one of the following scanners:

ScanShell 600

ScanShell 800

ScanShell 1000

MAG_ERR_NO_READER_FOUND: The magnetic reader device could not be found on any of the PC ports.

Remarks

This function scan COM1-COM16 and search for the magnetic reader device. Once found, the reader is initialized and the library loads and initialized.

IsReaderValid

Format

```
IsReaderValid () As Long
```

Return

MAG_ERR_NO_READER_FOUND: The reader is not connected to the PC.

MAG_ERR_NONE: The reader is connected to the PC and functioning correctly.

Remarks

Detect if the Magnetic Reader hardware is connected and functioning. The reader is searched in the COM port found in *InitLibrary* function.

WasCardSwept

Format

```
WasCardSwept () As Long
```

Return

MAG_ERR_NO_READER_FOUND: The reader is not connected to the PC.

SERIAL_NOT_INIT: Serial port is not initialized.

SERIAL_PORT_NOT_OPEN: Serial port could not be opened.

SERIAL_PORT_CONFIG_FAIL: COM Port configuration failed.

SERIAL_COM_TIMEOUT_FAIL: COM Port timeout failure.

MAG_ERR_CARD_NOT_DETECTED: No new card swipe was detected from the last call to this function.

MAG_ERR_CARD_DETECTED: A recent card swipe was detected and the data is available for process.

Remarks

Call this function periodically to find out if a new card swipe was performed. If no new swipe was performed, the function returns **MAG_ERR_CARD_NOT_DETECTED**. If the reader detects a new swipe it returns **MAG_ERR_CARD_DETECTED**. If the system is in error condition (due to bad initialization or disconnection of the reader from the PS) the function return one of the other values.

Process

Format

Process () As Long

Return

LONG_AAMVA: Standard AAMVA format (includes channel1, channel2 and channel3).

SHORT_AAMVA: Short AAMVA format (includes channel1 and channel3).

OLD_CA_DMV: Old DMV format (California).

OLD_LA_DMV: Old DMV format (Louisiana).

UNKNOWN_FORMAT: Unknown format. In such case no further processing is done.

Remarks

Call this function to process the recently swiped card raw data. The raw data is scanned for format detection. If a specific format is detected, the data is parsed further and load the library properties.

GetRawData

Format

GetRawData (**buffer** As String) As Long

Parameters

[in] **Buffer** – Null terminated string that receives the raw data.

Return

MAG_ERR_NONE: Data retrieve successfully.

MAG_ERR_CARD_NOT_DETECTED: Buffer is empty.

Remarks

Call this function to get the data as retrieved from the magnetic reader device without further processing.

MagLib Library Properties

MagLib library store the detected textual information in an internal data structure. This structure is refreshed in each Process() function call. Some of the properties are relevant to some magnetic card formats and some are irrelevant. For example, the property ISSUE is relevant to California Old DMV id cards and not to standard AAMVA cards. If a property is irrelevant to a particular state, or if the field is not detected, it clears to an empty string. Loading the properties is done using a simple string assignment.

The supported properties are:

Address	City	Class
DateOfBirth	Endorsements	ExpirationDate
Eyes	Hair	Height
Issue	License	NameFirst
NameLast	NameMiddle	Restriction
Sex	State	Weight
Zip		

Library CPassport: General Functionality

CPassport analyses and retrieve data from a standard passport image. The passport image is taken using ScanShell 1000 scanner in either color or gray color scheme, analysed by the library and the result data is stored the library properties. The image may be a full image of the page (3"x5") or only the lower portion of the page (1"x5").

CPassport Library Functions

Init

Format

```
Init (License As String) As Long
```

Parameters

[in] [License](#) – Null terminated string that holds license key value.

Return

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.

LICENSE_DOES_NOT_MATCH_LIBRARY: The license is invalid for this library. All library operations are disabled.

GENERAL_ERR_PLUG_NOT_FOUND: This error returns if the attached scanner is not one of the following scanners:

ScanShell 600

ScanShell 800

ScanShell 1000

MAG_ERR_NO_READER_FOUND: The magnetic reader device could not be found on any of the PC ports.

Remarks

This function initialize the library. This function must be called before any other function in the library can be used.

Process

Format

Process () As Long

Return

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

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

LICENSE_INVALID – The library was not initialized with the proper license.

INVALID_INTERNAL_IMAGE – No internal image is loaded. This value return when attempting to use this function without scanning an image first.

Remarks

Call this function to process the recently scanned passport image. When the scanning the passport page using the ScanShell1000, the opened page should be aligned to the top right corner which yields a rotated internal image. Before processing the image it needs to be rotated in 180 degrees (using the function [RotateImage](#)).

The image should have the following properties:

Color scheme: Select one of the following

24 bit (True color)

256 Gray shades

Image size: Select one of the following

3" x 5": This scan the full page of the passport

1" x 5": This scan s only the lower portion of the page.

Once the function returns **PASS_ERR_NONE** the library properties will be loaded with analyzed text. Otherwise, the library property fields will be empty.

The raw data is scanned for format detection. If a specific format is detected, the data is parsed further and load the library properties.

GetFace

Format

GetFace (**DestFile** As String) As Long

Parameters

[in] **DestFile** – Null terminated string that holds the full name of the destination image file that will contain the face image from the passport.

Return

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

If the function fails one of the following values is returned:

LICENSE_INVALID – Library was not initialized with proper license.

PASS_ERR_CANNOT_DELETE_DESTINATION_IMAGE– Returned when a file with the same name as the destination file already exist and cannot be overwritten.

PASS_ERR_CANNOT_COPY_TO_DESTINATION–Returned when the destination file cannot be opened for write on the disk.

PASS_ERR_FACE_IMAGE_NOT_FOUND – **Could not retrieve the face image from the passport image.**

Remarks

Use this function to extract the image rectangle of the person’s face from the source passport image. Remember that the original scanned image must be rotated in 180 degrees (so it will be aligned correctly) before this function is called. This function works properly only for 3”x5” images. The destination image is generated in BITMAP format.

C Passport Library Properties

C Passport library store the detected textual information in an internal data structure. This structure is refreshed in each Process() function call.

The supported properties are:

PassportNumber	NameFirst	NameMiddle
NameLast	ExpirationDate	DateOfBirth
Sex	PersonalNumber	Nationality
NationalityLong	Country	CountryLong

Notes:

The field *Country* stands for the Passport Issuing country

The fields *Country* and *Nationality* comes in two formats:

Regular format (*Nationality, Country*): The text as appears in the passport, i.e., USA for United States, SGP for Singapore, etc.

Long Format (*NationalityLong, CountryLong*): Interpreted country names: United States, Singapore, etc.

Appendix A – VB Demo Program

The attached demo VB project shows the utilization of the four libraries. The program shows in a nutshell the various aspects of the libraries:

Paper insertion detection
Document scan
Image file manipulation
OCR text extraction (using OCR related license only)
ID card data extraction (using ID CARD license only)

Using the license key in the program

To activate the proper libraries you must first apply the proper license number to the program. The license activates the libraries in the Form.Load() function.

To obtain the license key please contact your local Card Scanning Solution product dealer.

Applying the license key to the code

Open the file License.bas and type in your license value.
So instead of the original line:

```
' Setup the license value. This value is used to unlock the libraries usage  
Public Const LICENSE_VALUE = "XXXXXXXXXXXXXXXXXXXX"
```

You will type your license key value:

```
' Setup the license value. This value is used to unlock the libraries usage  
Public Const LICENSE_VALUE = "A7MIIFHXDZT8J3FU"
```

An updated temporary SDK license can be obtained from <http://www.id-scan.com/developer>

Appendix B – SDK installation

Installing the SDK package

The SDK files are packed in a single setup file (*SDK_Setup.exe*).

SDK can also be installed with the silent SDK installation process. This is typically used when you want to make SDK part of your installation process :

1. Download sdk_setup.exe (http://www.id-reader.com/ftp/applications/sdk/sdk_setup.exe)
2. Download Silent Installer zip from here: http://id-reader.com/Support/Silent_Installers/ and go through the ReadMe.txt in the zip package before initiating the silent installation
3. Open SDK Silent folder, open the SetupParams.txt -> MAINDIR=C:\ (Should be pointing to the SDK installation directory). Also in the SetupParams.txt, select the appropriate Components list that associated with APP_COMPONENTS= and COMPONENTS=select the scanner model
4. While in the SDK Silent Installer folder, right click on SilentInstaller.bat and choose Edit, this line ("C:\SDK_Silent_Installer\sdk_setup.exe") should be pointing to the location for the sdk_setup.exe that you have downloaded and also this line (C:\SDK_Silent_Installer\SetupParams.txt) should be pointing to the location of SetupParams.txt
5. Run SilentInstaller.bat and you should see the installation progress in the taskbar.

Appendix C – Constant Definitions

The following values are used as constants:

Library SlibEx constants

' Scanner color scheme types

Public Const GRAY = 1
Public Const BW = 2
Public Const HT = 3
Public Const TRUECOLOR = 4

' Scanner return values

Public Const SLIB_FALSE = 0
Public Const SLIB_TRUE = 1

' Scanner general error types

Public Const SLIB_ERR_NONE = 1
Public Const SLIB_ERR_INVALID_SCANNER = -1

' Scanning failure definition

Public Const SLIB_ERR_SCANNER_GENERAL_FAIL = -2
Public Const SLIB_ERR_CANCELED_BY_USER = -3
Public Const SLIB_ERR_SCANNER_NOT_FOUND = -4
Public Const SLIB_ERR_HARDWARE_ERROR = -5
Public Const SLIB_ERR_PAPER_FED_ERROR = -6
Public Const SLIB_ERR_SCANABORT = -7
Public Const SLIB_ERR_NO_PAPER = -8
Public Const SLIB_ERR_PAPER_JAM = -9
Public Const SLIB_ERR_FILE_IO_ERROR = -10
Public Const SLIB_ERR_PRINTER_PORT_USED = -11
Public Const SLIB_ERR_OUT_OF_MEMORY = -12

Public Const SLIB_ERR_BAD_WIDTH_PARAM = -2

Public Const SLIB_ERR_BAD_HEIGHT_PARAM = -3

Public Const SLIB_ERR_BAD_PARAM = -2

Public Const SLIB_LIBRARY_ALREADY_INITIALIZED = -13

Public Const SLIB_ERR_DRIVER_NOT_FOUND = -14

Public Const SLIB_ERR_SCANNER_BUSSY = -15

Public Const SLIB_ERR_IMAGE_CONVERSION = -16

Public Const SLIB_UNLOAD_FAILED_BAD_PARENT = -17

Public Const SLIB_NOT_INITIALIZED = -18

Public Const SLIB_LIBRARY_ALREADY_USED_BY_OTHER_APP = -19

Public Const SLIB_CONFLICT_WITH_INOUTSCAN_PARAM = -20

Public Const SLIB_CONFLICT_WITH_SCAN_SIZE_PARAM = -21

' Button definition for ScanShell1000

Public Const TOP_BUTTON = 1

Public Const MIDDLE_BUTTON = 3
Public Const BOTTOM_BUTTON = 2

'Error values for multiple devices management

Public Const SLIB_NO_SUPPORT_MULTIPLE_DEVICES = -22
Public Const SLIB_ERR_CAM_ALREADY_ASSIGNED = -23
Public Const SLIB_ERR_NO_FREE_CAM_FOUND = -24
Public Const SLIB_ERR_CAM_NOT_FOUND = -25
Public Const SLIB_ERR_CAM_NOT_ASSIGNED_TO_THIS_APP = -26
Public Const GENERAL_ERR_PLUG_NOT_FOUND = -200
Public Const ERR_SCANNER_ALREADY_IN_USE = -201
Public Const SLIB_ERR_SCANNER_ALREADY_IN_USE = -202
Public Const SLIB_ERR_CANNOT_OPEN_TWAIN_SOURCE = -203
Public Const SLIB_ERR_NO_TWAIN_INSTALLED = -204
Public Const SLIB_ERR_NO_NEXT_VALUE = -205

Library idData con stants

' Country definitions
Public Const COUNTRY_USA = 0
Public Const COUNTRY_AUSTRALIA = 1
Public Const COUNTRY_MALAYSIA = 2
Public Const COUNTRY_CANADA = 3
Public Const COUNTRY_CHILE = 4
Public Const COUNTRY_FRANCE = 5
Public Const COUNTRY_MEXICO = 6
Public Const COUNTRY_UNITED_KINGDOM = 7
Public Const COUNTRY_BRAZIL = 8
Public Const COUNTRY_ISRAEL = 9
Public Const COUNTRY_GERMANY = 10
Public Const COUNTRY_SPAIN = 11
Public Const COUNTRY_ROMANIA = 12
Public Const COUNTRY_BERMUDA = 13
Public Const COUNTRY_SINGAPORE = 14
Public Const COUNTRY_NORWAY = 15
Public Const COUNTRY_NEW_ZELAND = 16
Public Const COUNTRY_HOLAND = 17
Public Const COUNTRY_LUX = 18
Public Const COUNTRY_LITHUANIA = 19
Public Const COUNTRY_SWISS = 20
Public Const COUNTRY_BAHAMAS = 21
Public Const COUNTRY_SWEDEN = 22
Public Const COUNTRY_ITALY = 23
Public Const UNIVERSITY_USA = 24
Public Const COUNTRY_TURKEY = 25
Public Const EMPLOYMENT_CARDS = 26
Public Const COUNTRY_POLAND = 27
Public Const COUNTRY_PERU = 29
Public Const COUNTRY_PUERTO_RICO = 30
Public Const COUNTRY_PORTUGAL = 31
Public Const COUNTRY_NICARAGUA = 32
Public Const COUNTRY_GUATEMALA = 33
Public Const COUNTRY_EL_SALVADOR = 34
Public Const COUNTRY_SOUTH_AFRICA = 35
Public Const COUNTRY_PANAMA = 36
Public Const COUNTRY_INDONESIA = 37
Public Const COUNTRY_BELGIUM = 38
Public Const SERVICE_CARDS = 39
Public Const ENTERTAINMENT_CARDS = 40
Public Const COUNTRY_CROATIA = 41
Public Const USAPILOTS_CARDS = 42
Public Const COUNTRY_CHINA = 43
Public Const ACCESS_CARDS = 44
Public Const COUNTRY_BULGARIA = 45
Public Const fine EUROPE_GENERAL_CARDS = 46
Public Const COUNTRY_CZECH = 47
Public Const COUNTRY_BOSNIA = 48
Public Const COUNTRY_HUNGARY = 49
Public Const COUNTRY_SLOVAKIA = 50

Public Const COUNTRY_KOSOVO = 51
Public Const OCB_CARDS = 52
Public Const COUNTRY_SLOVENIA = 53
Public Const COUNTRY_IRELAND = 54
Public Const COUNTRY_UAE = 55
Public Const COUNTRY_BAHRAIN = 56
Public Const COUNTRY_AUSTRIA = 57
Public Const COUNTRY_RUSSIA = 58
Public Const COUNTRY_SERBIA = 59
Public Const COUNTRY_BOLIVIA = 60
Public Const SPAIN_POLICE_CARDS = 61
Public Const COUNTRY_LIECHTENSTEIN = 63
Public Const COUNTRY_FINLAND = 64
Public Const EHC_CARDS = 65
Public Const COUNTRY_ECUADOR = 67
Public Const COUNTRY_BRUNEI = 68
Public Const COUNTRY_HONDURA = 69
Public Const SCISUSAC_CARDS = 70
Public Const COUNTRY_ESTONIA = 71
Public Const COUNTRY_DENMARK = 72
Public Const COUNTRY_DOMINICAN_REPUBLIC = 73
Public Const COUNTRY_HAITI = 74
Public Const USAA_CARDS = 75
Public Const COUNTRY_CYPRUS = 76
Public Const AMPORT_CARDS = 77
Public Const COUNTRY_ISLAND = 78
Public Const COUNTRY_COLUMBIA = 79
Public Const COUNTRY_VENEZUELA = 80
Public Const COUNTRY_INDIA = 81
Public Const COUNTRY_NAMIBIA = 82
Public Const COUNTRY_ZAMBIA = 83
Public Const PH_CARDS_CARDS = 84
Public Const COUNTRY_OMAN = 85
Public Const COUNTRY_QATAR = 86
Public Const COUNTRY_SAUDI_ARABIA = 87
Public Const COUNTRY_ANDORRA = 88
Public Const COUNTRY_GUERNSEY = 89
Public Const COUNTRY_ISLE_OF_MAN = 90
Public Const COUNTRY_LATVIA = 91
Public Const COUNTRY_MALTA = 92
Public Const COUNTRY_ARGENTINA = 93
Public Const COUNTRY_ST_CHRIST_NEVIS = 94
Public Const COUNTRY_ALBANIA = 95
Public Const IRELAND_FIREARM_CARDS = 96
Public Const COUNTRY_MONTENEGRO = 97
Public Const COUNTRY_KENYA = 98
Public Const COUNTRY_NIGERIA = 99
Public Const COUNTRY_MACEDONIA = 100
Public Const COUNTRY_MOROCCO = 101
Public Const COUNTRY_PHILIPPINES = 102
Public Const COUNTRY_TURKS_CAICOS = 103
Public Const COUNTRY_THAILAND = 104
Public Const COUNTRY_MOLDOVA = 105

Public Const COUNTRY_BELIZE = 106
Public Const ISRAEL_DOCS = 107
Public Const COUNTRY_ANTIGUA = 108
Public Const TUNISIA_ELECTION_CARDS = 109
Public Const BEAUCE_CARDS 110
Public Const COUNTRY_IRAQ 111

' states definitions

Public Const ID_ALABAMA = 0
Public Const ID_ALASKA = 1
Public Const ID_ARIZONA = 2
Public Const ID_ARKANSAS = 3
Public Const ID_CALIFORNIA = 4
Public Const ID_COLORADO = 5
Public Const ID_CONNECTICUT = 6
Public Const ID_DELAWARE = 7
Public Const ID_WASHINGTON_DC = 8
Public Const ID_FLORIDA = 9
Public Const ID_GEORGIA = 10
Public Const ID_IDAHO = 11
Public Const ID_ILLINOIS = 12
Public Const ID_INDIANA = 13
Public Const ID_IOWA = 14
Public Const ID_KANSAS = 15
Public Const ID_KENTUCKY = 16
Public Const ID_LOUISIANA = 17
Public Const ID_MAINE = 18
Public Const ID_MARYLAND = 19
Public Const ID_MASSACHUSETTS = 20
Public Const ID_MICHIGAN = 21
Public Const ID_MINNESOTA = 22
Public Const ID_MISSISSIPPI = 23
Public Const ID_MISSOURI = 24
Public Const ID_MONTANA = 25
Public Const ID_NEBRASKA = 26
Public Const ID_NEVADA = 27
Public Const ID_NEW_HAMPSHIRE = 28
Public Const ID_NEW_JERSEY = 29
Public Const ID_NEW_MEXICO = 30
Public Const ID_NEW_YORK = 31
Public Const ID_NORTH_CAROLINA = 32
Public Const ID_NORTH_DAKOTA = 33
Public Const ID_OHIO = 34
Public Const ID_OKLAHOMA = 35
Public Const ID_OREGON = 36
Public Const ID_PENNSYLVANIA = 37
Public Const ID_RHODE_ISLAND = 38
Public Const ID_SOUTH_CAROLINA = 39
Public Const ID_SOUTH_DAKOTA = 40
Public Const ID_TENNESSEE = 41
Public Const ID_TEXAS = 42

Public Const ID_UTAH = 43
Public Const ID_VERMONT = 44
Public Const ID_VIRGINIA = 45
Public Const ID_WASHINGTON = 46
Public Const ID_WEST_VIRGINIA = 47
Public Const ID_WISCONSIN = 48
Public Const ID_WYOMING = 49
Public Const ID_HAWAII = 54
Public Const ID_GREEN_CARD = 81
Public Const ID_ARMY_CARD = 82
Public Const ID_SSN_CARD = 83
Public Const ID_NYPD = 84
Public Const ID_MEXICO_USA = 85
Public Const ID_GUAM = 86
Public Const ID_CEMA_COMPLIANT = 87

Public Const ID_NSW = 50
Public Const ID_ACT = 51
Public Const ID_QLD = 52
Public Const ID_VIC = 53
Public Const ID_TAS = 55
Public Const ID_WST = 56
Public Const ID_SA = 57
Public Const ID_NT = 58
Public Const ID_COOK_ISLANDS = 59

Public Const ID_MALAYSIA = 60
Public Const ID_SINGAPORE = 180
Public Const ID_NEW_ZELAND = 200

Public Const ID_ONTARIO = 70
Public Const ID_ALBERTA = 71
Public Const ID_BRITISH_COLUMBIA = 72
Public Const ID_MANITOBA = 73
Public Const ID_NEW_BRUNSWICK = 74
Public Const ID_NEW_FOUNDLAND = 75
Public Const ID_NWTERITORIES = 76
Public Const ID_NOVASCOTIA = 77
Public Const ID_SASKATCHEWAN = 78
Public Const ID_CANADA_CITIZEN_ID = 79
Public Const ID_QUEBEC = 1079

Public Const ID_CHILE = 80
Public Const ID_FRANCE = 90
Public Const ID_MEXICO = 100
Public Const ID_UNITED_KINGDOM = 110
Public Const ID_ISRAEL = 120
Public Const ID_BRAZIL = 130
Public Const ID_GERMAN_ID = 140
Public Const ID_GERMAN_LIC = 141
Public Const ID_SPAIN = 150
Public Const ID_ROMANIA = 160
Public Const ID_NORWAY = 190

Public Const ID_HOLAND = 210
Public Const ID_LUX = 220
Public Const ID_LITHUANIA = 230
Public Const ID_SWISS = 240
Public Const ID_SWEDEN = 260
Public Const ID_ITALY = 270
Public Const ID_UNIVERSITY_USA = 280
Public Const ID_TURKEY = 290
Public Const ID_EMPLOY = 300
Public Const ID_POLAND = 310
Public Const ID_COSTA_RICA = 320
Public Const ID_PERU = 330
Public Const ID_PUERTO_RICO = 340
Public Const ID_PORTUGAL = 350
Public Const ID_NICARAGUA = 360
Public Const ID_GUATEMALA = 370
Public Const ID_EL_SALVADOR = 380
Public Const ID_SOUTH_AFRICA = 390
Public Const ID_PANAMA = 400
Public Const ID_INDONESIA = 410
Public Const ID_BELGIUM = 420
Public Const ID_PROTECTIVE = 430
Public Const ID_ENTERTAINMENT = 440
Public Const ID_CROATIA = 450
Public Const ID_USAPILOTS = 460
Public Const ID_CHINA = 470
Public Const ID_ACCESS = 480
Public Const ID_BULGARIA = 490
Public Const ID_KEYPASS = 500
Public Const ID_EUROPE_GENERAL_CARDS = 510
Public Const ID_CZECH = 520
Public Const ID_BOSNIA = 530
Public Const ID_HUNGARY = 540
Public Const ID_SLOVAKIA = 550
Public Const ID_KOSOVO = 560
Public Const OCB_CARDS = 570
Public Const ID_SLOVENIA = 580
Public Const ID_MRZ_INFO = 590
Public Const ID_IRELAND = 600
Public Const ID_UAE = 610
Public Const ID_BAHRAIN = 620
Public Const ID_SPAIN_POLICE = 630
Public Const ID_AUSTRIA = 640
Public Const ID_RUSSIA = 650
Public Const ID_SERBIA = 660
Public Const ID_BOLIVIA = 670
Public Const ID_LIECHTENSTEIN = 680
Public Const ID_FINLAND = 690
Public Const ID_EHIC = 700
Public Const ID_ECUADOR = 710
Public Const ID_BRUNEI = 720
Public Const ID_HONDURAS = 730
Public Const ID_SCSIUSAC = 740

Public Const ID_ESTONIA = 750
Public Const ID_DENMARK = 760
Public Const ID_DOMINICAN_REPUBLIC = 770
Public Const ID_HAITI = 780
Public Const ID_USAA = 790
Public Const ID_CYPRUS = 800
Public Const ID_AMPART = 810
Public Const ID_ISLAND = 820
Public Const ID_COLUMBIA = 830
Public Const ID_VENEZUELA = 840
Public Const ID_INDIA = 850
Public Const ID_NAMIBIA = 860
Public Const ID_ZAMBIA = 870
Public Const ID_PRINCEEDWARD = 880
Public Const ID_PH_CARDS = 890
Public Const ID_OMAN = 900
Public Const ID_QATAR = 910
Public Const ID_POSITIVE_ACCESS_CARDS = 920
Public Const ID_SAUDI_ARABIA = 930
Public Const ID_ANDORRA = 940
Public Const ID_GUERNSEY = 950
Public Const ID_ISLE_OF_MAN = 960
Public Const ID_LATVIA = 970
Public Const ID_MALTA = 980
Public Const ID_ARGENTINA = 990
Public Const ID_ST_CHRIST_NEVIS = 1000
Public Const ID_ALBANIA = 1010
Public Const ID_IRELAND_FIREARM = 1020
Public Const ID_MONTENEGRO = 1030
Public Const ID_KENYA = 1040
Public Const ID_NIGERIA = 1050
Public Const ID_MACEDONIA = 1060
Public Const ID_MOROCCO = 1070
Public Const ID_PHILIPPINES = 1080
Public Const ID_TURKS_CAICOS = 1090
Public Const ID_THAILAND = 1100
Public Const ID_MOLDOVA = 1110
Public Const ID_BELIZE = 1120
Public Const ID_ISRAEL_APPRISELS = 1130
Public Const ID_ISRAEL_CABIDS = 1131
Public Const ID_ISRAEL_DAN = 1132
Public Const ID_ISRAEL_DL = 1133
Public Const ID_ISRAEL_EGED = 1134
Public Const ID_ISRAEL_GENERAL_CARDS = 1135
Public Const ID_ISRAEL_GOV_ENERGY = 1136
Public Const ID_ISRAEL_GOV_PM = 1137
Public Const ID_ISRAEL_GOV_TRANSPORT = 1138
Public Const ID_ISRAEL_GUNLIC = 1139
Public Const ID_ISRAEL_HASHMAL = 1140
Public Const ID_ISRAEL_ID = 1141
Public Const ID_ISRAEL_IDF = 1142
Public Const ID_ISRAEL_JUDICIARY = 1143
Public Const ID_ISRAEL_LOWERS = 1144

Public Const ID_ISRAEL_SECURITY = 1145
Public Const ID_ISRAEL_TAXI = 1146
Public Const ID_ANTIGUA = 1160
Public Const ID_TUNISIA_ELECTION = 1170
Public Const ID_BEAUCEComple = 1180
Public Const ID_IRAQ = 1190

' region definitions

Public Const REGION_USA = 0
Public Const REGION_CANADA = 1
Public Const REGION_SOUTH_AMERICA = 2
Public Const REGION_EUROPE = 3
Public Const REGION_AUSTRALIA = 4
Public Const REGION_ASIA = 5
Public Const REGION_GENERAL_DOC = 6
Public Const REGION_AFRICA = 7

Public Const FRONT_IMAGE_SIDE = 0
Public Const BACK_IMAGE_SIDE = 1
Public Const UNKNOWN_IMAGE = 2

'Date formats

Public Const EXTRACT_DATE_FORMAT_NONE = 0	'Use defaults
Public Const EXTRACT_DATE_FORMAT_MDY = 1	'mm-dd-yy
Public Const EXTRACT_DATE_FORMAT_DMY = 2	'dd-mm-yy
Public Const EXTRACT_DATE_FORMAT_YMD = 3	'yy-mm-dd
Public Const EXTRACT_DATE_FORMAT_YDM = 4	'yy-dd-mm

' function return values

Public Const ID_TRUE = 1
Public Const ID_FALSE = 0

' error enums

Public Const ID_ERR_NONE = 1
Public Const ID_ERR_STATE_NOT_SUPORTED = -2
Public Const ID_ERR_BAD_PARAM = -3
Public Const ID_ERR_NO_MATCH = -4
Public Const ID_ERR_FILE_OPEN = -5
Public Const ID_BAD_DESTINATION_FILE = -6
Public Const ID_ERR_FEATURE_NOT_SUPPORTEDED = -7

Public Const ID_ERR_COUNTRY_NOT_INIT = -8
Public Const ID_ERR_NO_NEXT_COUNTRY = -9
Public Const ID_ERR_STATE_NOT_INIT = -10
Public Const ID_ERR_NO_NEXT_STATE = -11
Public Const ID_ERR_CANNOT_DELETE_DESTINATION_IMAGE = -12
Public Const ID_ERR_CANNOT_COPY_TO_DESTONATION = -13
Public Const ID_ERR_FACE_IMAGE_NOT_FOUND = -14

' state auto-detect error

Public Const ID_ERR_STATE_NOT_RECOGNIZED = -15
Public Const ID_ERR_USA_TEMPLATES_NOT_FOUND = -16
Public Const ID_ERR_WRONG_TEMPLATE_FILE = -17
Public Const ID_ERR_REGION_NOT_INIT = -18
Public Const ID_ERR_AUTO_DETECT_NOT_SUPPORTEDED = -19
Public Const ID_ERR_COMPARE_NO_TEXT = -20
Public Const ID_ERR_COMPARE_NO_BARCODE = -21
Public Const ID_ERR_COMPARE_BC_LIB_NOT_FOUND = -22
Public Const ID_ERR_COMPARE_LICENSE_DONT_MATCH_BC_LIBRARY = -23
Public Const ID_ERR_DM_LIBRARY_NOT_FOUND = -24
Public Const ID_ERR_DM_LIBRARY_NOT_LOADED = -25
Public Const ID_ERR_DM_WM_NOT_FOUND = -26
Public Const ID_ERR_DM_WM_NOT_AUTHENTICATED = -27

'GetFaceEx imageType values (only at supported cards see function description)

Public Const FACE_IMAGE_TYPE_REGULAR = 0
Public Const FACE_IMAGE_TYPE_WITH_FRAME = 1

Library CImage constants

```
' return values
Public Const IMG_ERR_SUCCESS = 0
Public Const IMG_ERR_FILE_OPEN = -100
Public Const IMG_ERR_BAD_ANGLE_0 = -101
Public Const IMG_ERR_BAD_ANGLE_1 = -102
Public Const IMG_ERR_BAD_DESTINATION = -103
Public Const IMG_ERR_FILE_SAVE_TO_FILE = -104
Public Const IMG_ERR_FILE_SAVE_TO_CLIPBOARD = -105
Public Const IMG_ERR_FILE_OPEN_FIRST = -106
Public Const IMG_ERR_FILE_OPEN_SECOND = -107
Public Const IMG_ERR_COMB_TYPE = -108

Public Const IMG_ERR_BAD_COLOR = -130
Public Const IMG_ERR_BAD_DPI = -131
Public Const INVALID_INTERNAL_IMAGE = -132

' image saving target definition
Public Const SAVE_TO_FILE = 0
Public Const SAVE_TO_CLIPBOARD = 1

' image rotation angle definitions
Public Const ANGLE_0 = 0
Public Const ANGLE_90 = 1
Public Const ANGLE_180 = 2
Public Const ANGLE_270 = 3

' image combination options
Public Const IMAGE_COMB_VERTICAL = 0
Public Const IMAGE_COMB_HORIZONTAL = 1

' image color conversion
Public Const IMAGE_SAME_COLOR = 0
Public Const IMAGE_BW = 1
Public Const IMAGE_GRAY_256 = 2
Public Const IMAGE_COLOR_256 = 3
Public Const IMAGE_COLOR_TRUE = 4

'text stamp positions
Public Const IMAGE_TOP = 0
Public Const IMAGE_MIDDLE = 1
Public Const IMAGE_BOTTOM = 2

Public Const IMAGE_LEFT = 0
Public Const IMAGE_MID_HOR = 1
Public Const IMAGE_RIGHT = 2
```

Library COcr constants

' return values
Public Const TOCR_SUCCESS = 1
Public Const TOCRJOBERROR = -2
Public Const TOCR_BAD_TYPE = -3

' OCR text type detection
Public Const USE_ALPHANUM = 0
Public Const USED_NUM_ONLY = 2
Public Const USE_ALPHA_CAPS_ONLY = 3

License related constants

Public Const LICENSE_VALID = 1
Public Const LICENSE_EXPIRED = -20
Public Const LICENSE_INVALID = -21
Public Const LICENSE_DOES_NOT_MATCH_LIBRARY = -22
Public Const GENERAL_ERR_PLUG_NOT_FOUND = -200

Library CBarCode constants

' List of all driver license field indexes
' The values of the fields is extracted directly from the bar code analyzer.
Public Const BCF_NAME = 0
Public Const BCF_ADDRESS = 1
Public Const BCF_CITY = 2
Public Const BCF_JURISTRICTION_CODE = 3
Public Const BCF_POASTAL_CODE = 4
Public Const BCF_LICENSE_MAIN = 5
Public Const BCF_CLASS = 6
Public Const BCF_RESTRICTION = 7
Public Const BCF_ENDORSEMENT = 8
Public Const BCF_EXPIRES = 9
Public Const BCF_DOB = 10
Public Const BCF_SEX = 11
Public Const BCF_ISSUE = 12
Public Const BCF_HEIGHT = 13
Public Const BCF_WEIGHT = 14
Public Const BCF_EYES = 15
Public Const BCF_HAIR = 16
Public Const BCF_SSNUMBER = 17
Public Const BCF_PERMIT_CLASS = 18
Public Const BCF_PERMIT_EXP = 19

Public Const BCF_PERMIT_ID = 20
Public Const BCF_PERMIT_ISSUE = 21
Public Const BCF_PERMIT_REST = 22
Public Const BCF_PERMIT_ENDORSEMENT = 23
Public Const BCF_DRIVER_LAST_NAME = 24
Public Const BCF_DRIVER_FIRST_NAME = 25
Public Const BCF_DRIVER_MIDDLE_NAME = 26
Public Const BCF_DRIVER_NAME_SUFFIX = 27
Public Const BCF_DRIVER_NAME_PREFIX = 28
Public Const BCF_MAIL_STREET_ADDRESS2 = 29
Public Const BCF_RES_STREET_ADDRESS1 = 30
Public Const BCF_RES_STREET_ADDRESS2 = 31
Public Const BCF_RES_CITY = 32
Public Const BCF_RES_JURISTRICITION = 33
Public Const BCF_RES_POSTAL_CODE = 34
Public Const BCF_HEIGHT_CM = 35
Public Const BCF_WEIGHT_KG = 36
Public Const BCF_ISSUE_TIMESTAMP = 37
Public Const BCF_NUM_OF_DUPLICATES = 38
Public Const BCF_MEDICAL_IND = 39
Public Const BCF_ORGAN_DONOR = 40
Public Const BCF_NON_RESIDENT = 41
Public Const BCF_UNIWUE_CUSTOMER_ID = 42
Public Const BCF_AKA_DOB = 43
Public Const BCF_AKA_SSN = 44
Public Const BCF_AKA_NAME = 45
Public Const BCF_AKA_LAST_NAME = 46
Public Const BCF_AKA_FIRST_NAME = 47
Public Const BCF_AKA_MIDDLE_NAME = 48
Public Const BCF_AKA_SUFFIX = 49
Public Const BCF_AKA_PREFIX = 50

' emulated field indexes. Although PDF417 standard has fixed fields definition, the
' fields population depends on the type of state. The following emulated fields
' extract the data from the raw fields and organized it in unified manner regardless
' to the state type for convenient integration

Public Const BCF_EMUL_FULL_NAME = 100
Public Const BCF_EMUL_FIRST_NAME = 101
Public Const BCF_EMUL_MIDDLE_NAME = 102
Public Const BCF_EMUL_LAST_NAME = 103
Public Const BCF_EMUL_NAME_SUFFIX = 104
Public Const BCF_EMUL_DOB = 105
Public Const BCF_EMUL_ISSUE = 106
Public Const BCF_EMUL_EXP = 107
Public Const BCF_EMUL_ADDRESS = 108
Public Const BCF_EMUL_CITY = 109
Public Const BCF_EMUL_STATE = 110
Public Const BCF_EMUL_ZIP = 111
Public Const BCF_EMUL_LICENSE = 112
Public Const BCF_EMUL_SSN = 113

Public Const BCF_EMUL_END = 114
Public Const BCF_EMUL_EYES = 115

Public Const BCF_EMUL_HAIR = 116
Public Const BCF_EMUL_HEIGHT = 117
Public Const BCF_EMUL_WEIGHT = 118

' return error values
Public Const BC_ERR_NO_BC_FOUND = 0
Public Const BC_ERR_NONE = 1
Public Const BC_ERR_BAD_PARAM = -1

Library MagLib constants

```
Public Const MAG_ERR_NONE = 1
Public Const MAG_ERR_CARD_DETECTED = 2

Public Const MAG_ERR_NO_FREE_COM = -30
Public Const MAG_ERR_NO_READER_FOUND = -31
Public Const MAG_ERR_BAD_PARAM = -32
Public Const MAG_ERR_CARD_NOT_DETECTED = -33
Public Const SERIAL_NOT_INIT = -34
Public Const SERIAL_PORT_NOT_OPEN = -35
Public Const SERIAL_PORT_CONFIG_FAIL = -36
Public Const SERIAL_COM_TIMEOUT_FAIL = -37
Public Const SERIAL_FAIL_TO_TX = -38

' driver's license card formats
Public Const UNKNOWN_FORMAT = 0
Public Const LONG_AAMVA = 1
Public Const SHORT_AAMVA = 2
Public Const OLD_CA_DMV = 3
Public Const OLD_LA_DMV = 4
```


Appendix D – Debugging Flags

Developing image manipulation code may sometimes be a very hectic task due to the many image format support by the Cimage library. In some cases the notorious “General Failure Event” pops up with no reasonable explanation.

To ease the debug process a single debug property (named **DebugFlag**) was added to the libraries Cimage and idData.

The default value of these flags is 0 (non-active).

Setting these flags to 1 activate the message system inside the libraries so whenever an error occur the libraries to issue a more detailed error message that explains the failure reason.

Appendix E – Errata Information

The Cimage library currently does not supports image saving the following conditions:

Saving 256 color images in JPG or TIF formats.

Appendix F – Supported States for Detection

The following table shows the supported states by ScanW library. This list will be updated in every new version release of ScanW library.

Country Name	Region ID	Country Name	Country ID	Document Name	Document ID
USA	0	USA	0	Alabama	0
				Alaska	1
				Arizona	2
				Arkansas	3
				California	4
				Colorado	5
				Connecticut	6
				Delaware	7
				Washington D.C.	8
				Florida	9
				Georgia	10
				Idaho	11
				Illinois	12
				Indiana	13
				Iowa	14
				Kansas	15
				Kentucky	16
				Louisiana	17
				Maine	18
				Maryland	19
				Massachusetts	20
				Michigan	21
				Minnesota	22
				Mississippi	23
				Missouri	24
				Montana	25
				Nebraska	26
				Nevada	27
				New Hampshire	28
				New Jersey	29
				New Mexico	30
				New York	31
				North Carolina	32
				North Dakota	33
				Ohio	34
				Oklahoma	35
				Oregon	36
				Pennsylvania	37
				Rhode Island	38
South Carolina	39				

				South Dakota	40
				Tennessee	41
				Texas	42
				Utah	43
				Vermont	44
				Virginia	45
				Washington	46
				West Virginia	47
				Wisconsin	48
				Wyoming	49
				Hawaii	54
				Permanent Resident (Green Card)	81
				USA Army	82
				Social Security Card	83
				NY Police department	84
				Matricula consular (Mexican Id)	85
				Tribal	88
				US Virgin Islands	91
Australia	4	Australia	1	New South Wales - DL	50
				Australian Capital Territory - DL	51
				Queensland - DL	52
				Victoria - DL	53
				Tasmania - DL	55
				Western Australia - DL	56
				South Australia - DL	57
				Northern Territory - DL	58
				Cook Island - DL	59
				Fiji - DL	61
Asia	5	China	43	Hong Kong - National ID	470
		Indonesia	37	Indonesia - DL	410
		Malaysia	2	Malaysia - National ID + Driver License	60
		Singapore	14	Singapore - DL	180
		New Zealand	16	New Zealand	200
		United Arab Emirates	55	United Arab Emirates	610
		Brunei	68	Brunei	720
		Oman	85	Oman	900
		Qatar	86	Qatar	910
		Saudi Arabia	87	Saudi Arabia	930
		India	81	India	850
		Philippines	102	Philippines	1080
		Thailand	104	Thailand	1100
		Israel Hebrew Cards	107	Israel Hebrew Cards	1130-1146
		Iraq	111	ID	1190

Canada	1	Canada	3	Ontario - DL	70
				Alberta - DL	71
				British Columbia - DL	72
				Manitoba - DL	73
				New Brunswick - DL	74
				New Foundland - DL	75
				New Territories - DL	76
				Nova Scotia - DL	77
				Saskatchewan - DL	78
				Canadian Citizen ID	79
				Prince Edward	880
				Quebec - DL	1079
South America	2	Chile	4	Chile -DL + National ID	80
		Mexico	6	Mexico – DL + National ID + Voter card	100
		Brazil	8	Brazil –DL + National ID	130
		Bermuda	13	Bermuda –DL + National ID	17
		Costa Rica	28	Costa Rica – National ID	320
		El Salvador	34	El Salvador –DL + National ID	380
		Guatemala	33	Guatemala – DL + National ID	370
		Nicaragua	32	Nicaragua – National ID	360
		Panama	36	Panama – National ID	400
		Peru	29	Peru – National ID	330
		Puerto Rico	30	Puerto Rico – DL + Firearms License	340
		Bolivia	60	Passports	670
		Columbia	79	Identity card	830
		Ecuador	67	Identity card	710
		Honduras	69	Identity card	730
		Dominican Republic	73	Identity card	770
		Haiti	74	Identity card	780
		Venezuela	80	DL	840
		Argentina	93	Identity card	990
		St. Christ Nevis	94	Identity card	1000
		Turks and Caicos Islands	103	Identity card	1090
		Belize	105	Identity card	1120
		Antigua	108	Identity card	1160
Europe	3	France	5	France – National ID	90
		Belgium	38	Belgium – National ID	420
		Bosnia	48	Bosnia – DL + National	530

		ID	
Bulgaria	45	Bulgaria – National ID	490
Czech	57	Czech – DL	520
Croatia	41	Croatia – DL + National ID	450
Germany	10	Germany – DL	140
Germany	10	Germany – National ID	141
Spain	11	Spain – DL + National ID	150
Bosnia	48	Bosnia – DL + National ID	530
Hungary	49	Hungary – DL + National ID	540
Holland	17	Holland – DL + National ID	210
Italy	23	Identity card & driver license	270
Luxembourg	18	Luxembourg – National ID	220
Norway	15	Norway – DL	190
Poland	27	Poland – DL + National ID	310
Portugal	31	Portugal – National ID	350
Romania	12	Romania – DL + National ID	160
Slovakia	50	Slovakia – DL + National ID	550
Sweden	22	Sweden driver license	260
Switzerland	20	Switzerland – National ID	240
Turkey	25	Turkey – DL	290
Israel	9	Israel – DL	120
United Kingdom and Ireland	7	United Kingdom and Ireland – DL	110
Kosovo	51	Kosovo – National ID	560
Hungary	49	Identity card & driver license	540
Slovenia	53	Slovenia Identity card	580
Austria	57	Passports	640
Liechtenstein	63	Identity card	680
Finland	64	Identity card	690
Estonia	71	Identity card	750
Denmark	72	Identity card	760
Iceland	78	Identity card	820
Cyprus	76	Identity card	800
Andorra	88	Identity card	940
Guernsey	89	Identity card	950
Isle of man	90	Identity card	960
Latvia	91	Identity card	970
Malta	92	Identity card	980
Albania	95	Identity card	1010

		Montenegro	97	Identity card	1030
		Serbia	59	Identity card	660
		Macedonia	100	Identity card	1060
		Moldova	105	Identity card	1110
Africa	7	Namibia	82	Namibia	860
		South Africa	35	South Africa	390
		Zambia	83	Zambia	870
		Kenya	98	Kenya	1040
		Nigeria	99	Nigeria	1050
		Morocco	101	Morocco	1070
		ANTIGUA	108	ANTIGUA	1160
General Documents	6	University documents (USA)	24	Student Id (UMASS, Boston Un., Emerson Clg., Harward Un., NorthEastern Un., Suffolk Un.)	280
		Employment card	26	Employment card	300
		Service card	39	Service card	430
		Entertainment Card	40	Entertainment Card	440
		USA Pilots	42	USA Pilots	460
		Access Card	44	Access Card	480
		OCB Cards	52	OCB Cards	570
		Spain Police	61	Spain Police	630
		EHIC	65	EHIC	700
		SCSIUSAC	70	SCSIUSAC	740
		USAA	75	USAA_CARDS	790
		PH Cards	84	PH Cards	890
		AMPORT Cards	77	AMPORT Cards	810
		Ireland Firearm	96	Ireland Firearm	1020
		Tunisia Election	109	Tunisia Election	1170
		BEAUCE	110	BEAUCE	1180