

ScanW.DLL

SDK Library Description

November 2014

Table of Contents

Table of Contents	2
Introduction	5
Retrieving information from ID cards	7
Functions and Properties Summary	8
Licensing	9
Distribution	9
Library Properties and Methods	11
Library SlibEx: General Functionality	11
SlibEx Library Functions	11
InitLibrary	11
CalibrateScanner	13
Clean	13
ScanToFile	13
ScanToFileEX	14
ResetIntImage	15
SlibEx Library Properties	16
IsNeedCalibration	16
IsScannerValid	16
LastErrorStatus	16
PaperInTray	16
PressedButton	17
Resolution	17
ScanHeight	18
ScannerColorScheme	18
ScanWidth	19
ScannerType	19
Version	20
Library IdData: General Functionality	21
idData Library Functions	21
AutoDetectState	21
AutoDetectStateEx	21
InitLibrary	22
ProcState	23
GetFacelImage	24
GetFirstCountry	25
GetFirstStateByCountry	25
GetNextCountry	25
GetNextStateByCountry	26
GetSignImage	26
RefreshData	27
RefreshDataAu	28
State2Id	28
Id2Country	29
Id2State	29
StatesSupported	30
GetDetectAccuracy	30

CountySupportAutoDetect.....	30
idData Library Properties	31
Library CBarcode: General Functionality.....	32
CBarcode Library Functions.....	32
InitLibrary	32
Proclmage.....	32
RefreshData.....	33
GetRawField	33
GetRawData	34
CBarcode Library Properties.....	35
Library CImage: Properties and Functions.....	36
InitLibrary	36
GetImageColor.....	36
Rotatelmage	37
ConvertImage	39
ReformatImage	40
Concatenatelmage.....	41
Library COcr: Properties and Functions	42
COcr Library Functions.....	42
InitLibrary	42
ExtractText.....	43
ExtractTextEx.....	44
COcr Library Properties	44
mText.....	44
Library MagLib: General Functionality.....	46
MagLib Library Functions.....	46
InitLibrary	46
IsReaderValid	47
WasCardSwept	47
Process.....	48
GetRawData	48
MagLib Library Properties.....	48
Library CPassport: General Functionality.....	49
CPassport Library Functions.....	49
Init.....	49
Process.....	50
GetFace	50
CPassport Library Properties.....	51
Appendix A – VB Demo Program.....	51
Using the license key in the program.....	52
Applying the license key to the code	52
Appendix B – SDK installation	53
Installing the SDK package	53
Appendix C – Constant Definitions	54
Library SlibEx constants.....	54
Library CImage constants	69
Library COcr constants.....	70
License related constants.....	70
Library CBarcode constants	70

Library MagLib constants 72
Appendix D – Debugging Flags 73
Appendix E – Errata Information 73
Appendix F – Supported States for Detection 74
Appendix G – Supported Passports for Detection 81

Introduction

ScanW.Dll library is a wrapper COM object that eases the integration between VC++ source libraries and the 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 the OCR version only)** – Extracts the textual data from the internal image.
- **IdData (In the ID version only)** – Parses and refines the textual data extracted by the OCR. The data is kept in internal variables ready to be exported to the application.
- **Cimage** – Used 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 file manipulation.

Each sub-library can function as an individual library regardless of the other libraries.

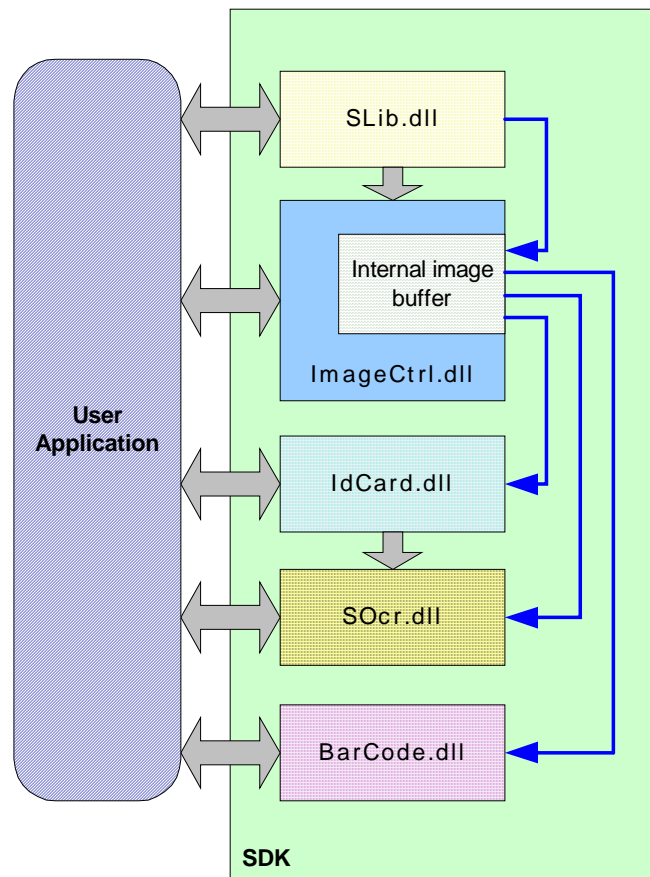


Figure1: ScanW library internal architecture

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's responsibility to ensure that the scan is performed in the proper setting (i.e. True color, 300 dpi, 2.2 x 3.6 inches). Failing to do so may result in bad data retrieval.

Functions and Properties Summary

Library: SlibEx

Name	Type	Functionality
InitLibrary	Function	Activates and enables the library functionality
CalibrateScanner	Function	Calibrates the scanner color sensor
ScanToFile	Function	Scans document to a bitmap file
IsNeedCalibration	Property	Checks if the scanner needs to be calibrated
IsScannerValid	Property	Verifies the scanner functionality
LastErrorStatus	Property	Retrieves the last operation error status
PaperInTray	Property	Checks if a document is in the scanner tray
Resolution	Property	Sets/Retrieves the scanning resolution
ScanHeight	Property	Sets/Retrieves the image height
ScannerColorScheme	Property	Sets/Retrieves the scanner to color/gray/bw color scheme
ScanWidth	Property	Sets/Retrieves the image width

Library: Idata

Name	Type	Functionality
ProcState	Function	Initializes the image process and data extraction from the id image file
RefreshDate	Function	Updates 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
Concatenate	Function	Concatenates two images into a single image file
ConvertImage	Function	Converts image format (internal or external image)
RotateImage	Function	Rotates image clock-wise in 90,180 and 270 degrees (internal or external image)
ReformatImage	Function	Modifies the image dpi, color scheme and save scheme (internal or external image)

Library: COcr

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

Licensing

Each library *must* 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 is 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:** A unique key that is used to activate the library with no time limit and is 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 the destination computer, you simply 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 registered 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 non COM\ActiveX files to exist before registration.

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

ScanW.dll (Com object)

ScanWEx.dll (Com object)

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

Note:

If you do not use the COM interface in your application and you use the SDK files directly like in VC++, then you do not 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 resets upon reading.

Important: Some scanner functions may take several seconds to execute (such as **Scan** or **Clean**). While executing a lengthy function, one should not try to execute a second scanner function until the previous function has returned. Failing to follow this rule may cause a 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 that the scanner is connected and re-start the function. 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 unloads, it should call the function UnInit to unload the SDK and release its memory (See the function description in Section 1.1.34 in the ScanWex.pdf)

CalibrateScanner

Format

```
CalibrateScanner ()
```

Return value

Void.

Remarks

This function calibrates the scanner using the calibration card. The calibration results are stored in a file inside the windows directory. The operation result can be tested for good completion by reading the [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 cleans the scanner lens by running the cleaning pad (Supplied in the scanner kit) back and forth. This function applies only to scanner modules ScanShell® 800/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 the function succeeds, the return value is **SLIB_ERR_NONE**.

If the function fails, the return 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 a 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 a document to the internal image buffer and, at the same time, export it to a bitmap file named “File Name” in the local disk. The operation result can be tested for good completion by reading the [LastErrorStatus](#) property.

Note 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 by 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 an external file in one of the seven popular file formats using either [ConvertImage\(\)](#) or [ReformatImage\(\)](#) or [RotateImage\(\)](#)

ScanToFileEX

Format

```
ScanToFileEx (FileName As String) As
```

Parameters

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

Return

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

If the function fails, the return 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 ScansToFile, only this function will display a progress bar that will show the scanning progress.

ResetIntImage

Format

```
ResetIntImage ()
```

Remarks

Deletes the internal image buffer.

SlibEx Library Properties

IsNeedCalibration

Type
Property.

Direction
Read Only.

Remarks
Retrieves if the scanner needs to be calibrated. This should be tested before every scan. A 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
Detects if the scanner version is supported by the current code. This property is 0 if the scanner is not supported and non-zero if it is supported.

LastErrorStatus

Type
Property.

Direction
Read Only.

Remarks
Retrieves the preceding property/command setup. This property is equal to **ELIB_ERR_NONE** when no error happened in the previous command execution. This property clears itself automatically to **ELIB_ERR_NONE** after reading.

PaperInTray

Type
Property.

Direction
Read only.

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

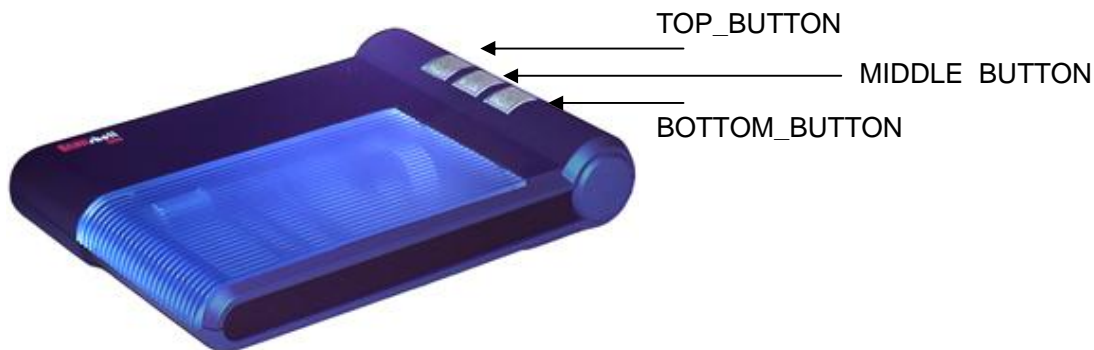
PressedButton

Type
Property.

Direction
Read only.

Remarks
Returns the button number that was pressed (Valid only when using ScanShell® 1000 scanner model). Reads this value after reading the property *PaperInTray* (that indicates 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

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

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

ScanHeight**Type**

Property.

Direction

Read / Write.

Remarks

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

Auto-detect Scan Size: When using scanner models ScanShell® 800xx\2000xx\3000xx, the scanner may also scan a document with an un-known size. In this mode, the scanner scans the document until it 'senses' that the entire document has been fed into the scanner. Then the black borders of the image are automatically cropped and the result image is returned to the calling application. To trigger this feature, you must set both properties *ScanHeight* and *ScanWidth* to -1. When the scan completes, these properties will reflect the detected document size in milli-inches.

Note: If you intend to use the *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

Sets/Retrieves the scanner color scheme. Available values are:

- BW
- GRAY
- TRUECOLOR

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

ScanWidth

Type

Property.

Direction

Read / Write.

Remarks

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

Auto-detect Scan Size: When using scanner models ScanShell® 800xx\2000xx\3000xx, the scanner may also scan a document with an un-known size. In this mode, the scanner scans the document until it “senses” that the entire document has been fed into the scanner. Then the black borders of the image are automatically cropped and the result image is returned to the calling application. To trigger this feature, you must set both properties *ScanHeight* and *ScanWidth* to -1. When the scan completes, these properties will reflect the detected document size in milli-inches.

Note: If you intend to use the *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

Retrieves the scanner type.

The 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

Retrieves 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), processes its graphic information and activates the OCR. The resultant text is kept in internal data structure ready to be retrieved by the application.

Note: 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

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 (*UsalDs.bin*) is missing. The file should be located in the SDK files location.

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

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

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

If none of the above error values is returned, the function returns the state ID value.

Remarks

Use this function to automatically detect the state type according to the image. If the function returns with none 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.

AutoDetectStateEx

Format

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

Parameters

[in] [Reserved](#) – Null terminated empty string - reserved. This parameter is not used.
 [out] **angle** – Returns the amount of clockwise 90 degree 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 (*UsalDs.bin*) is missing. The file should be located in the SDK files location.
INVALID_INTERNAL_IMAGE: No internal image is loaded. This value returns when attempting to use this function without scanning an image first.
ID_ERR_STATE_NOT_SUPPORTED: The license image does not match any state template.
ID_ERR_STATE_NOT_RECOGNIZED: The license image does not match any state template.

If none of the above error values is returned, the function returns the state ID value.

Remarks

Use this function to automatically align the internal image horizontally and then to detect the angle of rotation.

If the function returns with none 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 an *angle* of one of the following values:

- ANGLE_0: The image was received in the proper alignment.
- ANGLE_90: The image was rotated once by 90 degrees (clockwise).
- ANGLE_180: The image was rotated twice by 90 degrees (clockwise).
- ANGLE_270: The image was rotated three times by 90 degrees (clockwise).

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

InitLibrary

Format

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

Parameters

[in] [License](#) – Null terminated string that holds the 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 the function succeeds, the return value is **ID_TRUE**.

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

LICENSE_INVALID – Library was not initialized with a 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 returns 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 deskews and cleans the image and then passes it 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:

1. The image must be scanned in 24 bit (true color) and 300 dpi.
2. The image must be aligned horizontally with a tolerance of no more than ±10 degrees.

GetFacelImage

Format

```
GetFacelImage (SourceFileName As String, DestFileName As String,
statelD 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 the 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] [statelD](#) – The state index value as defined in the idLibDef.bas file.

Return

If the 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 returns 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 the destination file).
- ID_ERR_FILE_OPEN** – Bad source image file (used only when using a file as the source image). This value returns 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 returns if there is no image in the buffer.
- ID_FALSE** – Internal processing error.
- ID_ERR_FACE_IMAGE_NOT_FOUND**– Returns when the analyzer cannot detect the face image in the driver's license image.
- ID_ERR_CANNOT_DELETE_DESTINATION_IMAGE**– Returns when a file with the same name as the destination file already exists and cannot be overwritten.
- ID_ERR_CANNOT_COPY_TO_DESTONATION**– Returns 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* allows 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 returns 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.

Returns 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 continuously 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.

Returns 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] [statelid](#) – The state index value as defined in the idLibDef.bas file.

Return

If the 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 returns 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 the destination file).

ID_ERR_FILE_OPEN – Bad source image file (used only when using a file as the source image). This value returns 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 returns if there is no image in the buffer.

ID_FALSE– Internal processing error.

ID_ERR_FACE_IMAGE_NOT_FOUND– Returns when the analyzer cannot detect the face image in the driver’s license image.

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

ID_ERR_CANNOT_COPY_TO_DESTINATION– Returns 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 a 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 the [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 a non-zero value, the data was retrieved successfully.
If the function returns a 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 the ProcState function.

RefreshDataAu**Type**

Function.

Format

```
RefreshDataAu () As Long
```

Return Value

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

Remarks

This function is identical to the [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 a recognizable state name.

ID_ERR_NO_MATCH – If the string contains an unrecognizable state name.

Remarks

This helper function is used to convert a state name (string) to its equivalent ID value. For example, passing the 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 the [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.

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

StatelsSupported

Format

```
StatelsSupported (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.

GetDetectAcuracy

Format

```
GetDetectAcuracy () 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 percentage). This function, combined with the [RotatImage\(\)](#) function is useful when implementing the auto-orientation feature in the application. This feature can be implemented by rotating the internal image and for each new position it executes the [ProcState\(\)](#) function. The operation success can be evaluated using [GetDetetAcuracy\(\)](#) which allows us to determine if the image is in the right orientation or if 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 supports 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 stores 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	ExpirationDate	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 IDs from the first (index 0) to the last one.

Library CBarCode: General Functionality

CBarCode library functionality is similar to idData library. It extracts the data from 2D, PDF417 type bar code images. The library fetches the internal image (the last scanned image), processes 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.

Note: 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 barcodes, we require a 600dpi color image (RGB).

CBarCode Library Functions

InitLibrary

Format

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

Parameters

[in] **License** – Null terminated string that holds the 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.

ProclImage

Format

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

Parameters

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

Return

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

If the 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 returns when attempting to use this function without scanning an image first.

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

Remarks

Use this function to process the internal image acquired in the last scan. The function deskews and cleans the image and then passes it to the image analyzer for data extraction. The resultant textual data is kept in internal structure ready for retrieval by the [RefreshData\(\)](#) function. Processing the image does not modify the image content.

Successful image processing depends on the following:

Bar code must be scanned at 600dpi 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 a non-zero value, the data was retrieved successfully.

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

Remarks

AAMVA National standard defines 51-field types 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 an empty string. The maximum field size may be up to 60 characters long. It's 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 an Alabama driver license, and the same field is populated in field index BCF_DRIVER_LAST_NAME for a Delaware driver license. The raw field ranges from BCF_NAME (0) to BCF_AKA_PREFIX (50).

In addition to the raw fields, the CBarCode offers an additional set of fields (ranging 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 burden from the developer 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 stores 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	ExpirationDate
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 an external file or the image object stored in the 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 by 90, 180 or 270 degrees.

Resolution modification: Modifying 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: Attaching two images (horizontally or vertically) to form a single image file of both ID card sides.

The image can be exported (saved) to an 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 the 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 (Image As Image) As Color
```

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 colors (24 bit image).

Remarks

Use this function to obtain the image color scheme.

RotatImage

Format

```
RotatImage ( _
    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 to the clipboard.

SAVE_TO_IR : Saves the IR image. (If available)

SAVE_TO_UV: Saves the UV Image. (If available)

[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 the function succeeds, it returns 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 nor 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 the clipboard due to an error.

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

Remarks

Use this function to rotate an image by 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 an 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 an external image file. The destination file name may be one of the seven file formats indicated above. If an image has an unrecognizable extension due to an error (e.g. TIFF instead of TIF), the function exports the file in a BITMAP format. The destination file name may be the same as the source file name. In such a 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's flexibility actually allows you to do the following implicitly:

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:

RotatImage (“”, 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 the function succeeds, the return the value is **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 nor 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 an invalid destination file or disk save error.

Remarks

This function is a shorter version of the function [RotatImage\(\)](#). It takes an input file (if [SourceImage](#) is not empty) or uses the internal image as a source (if [SourceImage](#) is empty) and saves it to a file. Using different file extensions 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 schemes 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				
TIF				
JPG				
PCX				
TGA				

PNG				
PSD				

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

ReformatImage

Format

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

Parameters

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

[in] [toColor](#) – One of five values:

LICENSE_INVALID – Library was not initialized with proper license.

IMAGE_SAME_COLOR – No modification in the image color scheme

IMAGE_BW – Convert to black and white color scheme.

IMAGE_GRAY_256 – Convert to 256 gray scale color scheme.

IMAGE_COLOR_256 – Convert to 256-color scheme.

IMAGE_COLOR_TRUE – Convert to true color scheme.

[in] [toDpi](#) – Set the new Image DPI. A value of 0 indicates no DPI modification.

[in] [DestImage](#) – Full path name of the destination file. If this value is an empty string, no save will be performed.

Return

If the function succeeds, it returns 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 an 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 an unrecognizable extension due to an error (e.g. TIFF instead of TIF), the function refers to the file as BITMAP.

Image reformatting 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 an 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 using the 256 gray scale palette.

After the image is reformatted, it can be exported to an external image file. The destination file name may be one of the seven file formats indicated above. If the destination file name has an 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 the SourceImage0 before the combination.

[in] [SourceImage1](#) – Full path name of the second image.

[in] [Angle1](#) – The angle to rotate the 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 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 the function succeeds, it returns 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 nor clipboard)

IMG_ERR_COMB_TYPE – Bad [CombType](#) value.

IMG_ERR_FILE_SAVE_TO_CLIPBOARD – Cannot save image to the 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.

Rotates image object0 by [Angle0](#)

Imports [SourceImage1](#) to an image object 1.

Rotates image object1 by [Angle1](#)

Combines Image0 and Image1 one on top of each other (if [CombType](#) is equal to **IMAGE_COMB_VERTICAL**) or one to the left of the other (if [CombType](#) is equal to **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 a black and white color scheme.

COcr Library Functions

InitLibrary

Format

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

Parameters

[in] **License** – Null terminated string that holds the 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** – Instructs the OCR what type of data is written in the image. This value increases the detection accuracy and speeds up 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 the function succeeds, it returns 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 the function succeeds, it returns 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 processes the image file in a different method than the *ExtractText* function does; this results in a longer processing time but with higher accuracy. This function is recommended for use with a 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 the [ExtractText\(\)](#) function.

Library MagLib: General Functionality

MagLib controls the magnetic reader, collects and analyses its data once a card is 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 refreshes 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 the 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 scans COM1-COM16 and searches for the magnetic reader device. Once found, the reader is initialized and the library loads and initializes.

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

Detects if the Magnetic Reader hardware is connected and functioning. The reader is searched for in the COM port which is found in the *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 processing.

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 PC), the function returns 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 a 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 loads 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 retrieved 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

The MagLib library stores 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 the 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 retrieves data from a standard passport image. The passport image is taken using the ScanShell@1000 scanner in either color or gray color scheme, analyzed by the library and the result data is stored the library properties. The image may be a full image of the page (3"x 5") or only the lower portion of the page (1"x 5").

CPassport Library Functions

Init

Format

Init ([License](#) As String) As Long

Parameters

[in] [License](#) – Null terminated string that holds the 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 initializes the library. This function must be called before any other function in the library can be used.

Process

Format

```
Process () As Long
```

Return

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

If the 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 returns when attempting to use this function without scanning an image first.

Remarks

Call this function to process the recently scanned passport image. When scanning the passport page using the ScanShell®1000, 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 by 180 degrees (using the function [RotatImage](#)).

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 scans the full page of the passport

1" x 5": This scans 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 loads 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 the function succeeds, it returns the value **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– Returns when a file with the same name as the destination file already exists and cannot be overwritten.

PASS_ERR_CANNOT_COPY_TO_DESTINATION– Returns 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 by 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.

CPassport Library Properties

CPassport library stores 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* come in two formats:

Regular Format (*Nationality, Country*): The text as it 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 for 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 Solutions 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 = "XXXXXXXXXXXXXXXXXX"
```

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 is 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") which 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) which 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 constants

' 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 EHIC_CARDS = 65
 Public Const COUNTRY_ECUADOR = 67
 Public Const COUNTRY_BRUNEI = 68
 Public Const COUNTRY_HONDURA = 69
 Public Const SCSIUSAC_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
Public Const COUNTRY_CURACAO	112
Public Const COUNTRY_AZERBAIJAN	113
Public Const COUNTRY_IVORY_COAST	114
Public Const COUNTRY_VIRGINISLANDS	115
Public Const COUNTRY_KUWAIT	116
Public Const INTERPOL_CARDS	117
Public Const COUNTRY_ARUBA	118
Public Const T_MOBILE_CARDS	119
Public Const COUNTRY_TRINIDAD	120
Public Const COUNTRY_GREECE	121
Public Const EASYPAY_CARDS	122
Public Const COUNTRY_MAURITANIA	123
Public Const COUNTRY_CAMEROON	124
Public Const COUNTRY_BELARUS	125
Public Const COUNTRY_EUR_GEORGIA	126
Public Const COUNTRY_UZBEKISTAN	127
Public Const COUNTRY_ARMENIA	128
Public Const COUNTRY_MONACO	129
Public Const COUNTRY_SAN_MARINO	130
Public Const COUNTRY_UKRAINE	131
Public Const COUNTRY_VATICAN	132
Public Const COUNTRY_BARBADOS	133
Public Const COUNTRY_CAYMAN_ISLANDS	134
Public Const COUNTRY_CUBA	135
Public Const COUNTRY_FRENCH_GUIANA	136
Public Const COUNTRY_GREENLAND	137
Public Const COUNTRY_GRENADA	138
Public Const COUNTRY_GUYANA	139
Public Const COUNTRY_JAMAICA	140
Public Const COUNTRY_PARAGUAY	150
Public Const COUNTRY_SAINTE_KITTS_NEVIS	160
Public Const COUNTRY_SAINTE_LUCIA	170
Public Const COUNTRY_SAINTE_VINCENT_GRENADINES	180
Public Const COUNTRY_SURINAME	190
Public Const COUNTRY_URUGUAY	200
Public Const COUNTRY_ALGERIA	210
Public Const COUNTRY_ANGOLA	220
Public Const COUNTRY_BENIN	230
Public Const COUNTRY_BOTSWANA	240
Public Const COUNTRY_BURKINA_FASO	250

Public Const	COUNTRY_BURUNDI	260	
Public Const	COUNTRY_CAPE_VERDE	270	
Public Const	COUNTRY_CENTRAL_AFRICAN_REPUBLIC	280	
Public Const	COUNTRY_CHAD	290	
Public Const	COUNTRY_COMOROS	300	
Public Const	COUNTRY_REPUBLIC_OF_THE_CONGO	310	
Public Const	COUNTRY_DEMOCRATIC_REPUBLIC_OF_THE_CONGO		320
Public Const	COUNTRY_DJIBOUTI	330	
Public Const	COUNTRY_EGYPT	340	
Public Const	COUNTRY_EQUATORIAL_GUINEA	350	
Public Const	COUNTRY_ERITREA		360
Public Const	COUNTRY_ETHIOPIA	370	
Public Const	COUNTRY_GABON	380	
Public Const	COUNTRY_THEGAMBIA	390	
Public Const	COUNTRY_GHANA	400	
Public Const	COUNTRY_GUINEA		410
Public Const	COUNTRY_GUINEA_BISSAU	420	
Public Const	COUNTRY_LESOTHO		430
Public Const	COUNTRY_LIBERIA		440
Public Const	COUNTRY_LIBYA	450	
Public Const	COUNTRY_MADAGASCAR	460	
Public Const	COUNTRY_MALAWI		470
Public Const	COUNTRY_MALI	480	
Public Const	COUNTRY_MAURITIUS	490	
Public Const	COUNTRY_MOZAMBIQUE	500	
Public Const	COUNTRY_NIGER	510	
Public Const	COUNTRY_RWANDA		520
Public Const	COUNTRY_SAO_TOME_AND_PRINCIPE	530	
Public Const	COUNTRY_SENEGAL		540
Public Const	COUNTRY_SEYCHELLES	550	
Public Const	COUNTRY_SIERRA_LEONE	560	
Public Const	COUNTRY_SOMALIA		570
Public Const	COUNTRY_SUDAN	580	
Public Const	COUNTRY_SWAZILAND	590	
Public Const	COUNTRY_TANZANIA	600	
Public Const	COUNTRY_TOGO	610	
Public Const	COUNTRY_TUNISIA		620
Public Const	COUNTRY_UGANDA		630
Public Const	COUNTRY_WESTERN_SAHARA	640	
Public Const	COUNTRY_ZAIRE	650	
Public Const	COUNTRY_ZIMBABWE	660	
Public Const	COUNTRY_AFGHANISTAN	670	
Public Const	COUNTRY_BANGLADESH	680	
Public Const	COUNTRY_BHUTAN		690
Public Const	COUNTRY_CAMBODIA	700	
Public Const	COUNTRY_EAST_TIMOR	710	
Public Const	COUNTRY_IRAN	720	
Public Const	COUNTRY_JAPAN	730	
Public Const	COUNTRY_JORDAN		740
Public Const	COUNTRY_KAZAKHSTAN	750	
Public Const	COUNTRY_KYRGYZSTAN	760	
Public Const	COUNTRY_LAOS	770	
Public Const	COUNTRY_LEBANON		780
Public Const	COUNTRY_MALDIVES	790	
Public Const	COUNTRY_MONGOLIA	800	

Public Const	COUNTRY_MYANMAR		810
Public Const	COUNTRY_NEPAL		820
Public Const	COUNTRY_NORTH_KOREA		830
Public Const	COUNTRY_PAKISTAN		840
Public Const	COUNTRY_SOUTH_KOREA		850
Public Const	COUNTRY_SRI_LANKA		860
Public Const	COUNTRY_SYRIA		870
Public Const	COUNTRY_TAJIKISTAN		880
Public Const	COUNTRY_TURKMENISTAN	890	
Public Const	COUNTRY_VIETNAM		900
Public Const	COUNTRY_YEMEN		910

' 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_AMPORT = 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
 Public Const ID_CURACAO = 1200
 Public Const ID_ISRAEL_ASHDOD_PORT = 1210
 Public Const ID_ISRAEL_ELAL = 1220
 Public Const ID_ISRAEL_HAIFA_PORT = 1230
 Public Const ID_AZERBAIJAN = 1240
 Public Const ID_IVORY_COAST = 1250
 Public Const ID_VIRGINISLANDS = 1260
 Public Const ID_KUWAIT = 1270
 Public Const ID_INTERPOL = 1280
 Public Const ID_ARUBA = 1290
 Public Const ID_T_MOBILE = 1300
 Public Const ID_TRINIDAD = 1310
 Public Const ID_GREECE = 1320
 Public Const ID_EASYPAY = 1330
 Public Const ID_MAURITANIA = 1340
 Public Const ID_CAMEROON = 1350
 Public Const ID_BELARUS = 1360
 Public Const ID_EUR_GEORGIA = 1370
 Public Const ID_UZBEKISTAN = 1380
 Public Const ID_ARMENIA = 1390
 Public Const ID_MONACO = 1400
 Public Const ID_SAN_MARINO = 1410

Public Const ID_UKRAINE = 1420
 Public Const ID_VATICAN = 1430
 Public Const ID_BARBADOS = 1440
 Public Const ID_CAYMAN_ISLANDS = 1450
 Public Const ID_CUBA = 1460
 Public Const ID_FRENCH_GUIANA = 1470
 Public Const ID_GREENLAND = 1480
 Public Const ID_GRENADA = 1490
 Public Const ID_GUYANA = 1500
 Public Const ID_JAMAICA = 1510
 Public Const ID_PARAGUAY = 1520
 Public Const ID_SAINTE_KITTS_NEVIS = 1530
 Public Const ID_SAINTE_LUCIA = 1540
 Public Const ID_SAINTE_VINCENT_GRENADINES = 1550
 Public Const ID_SURINAME = 1560
 Public Const ID_URUGUAY = 1570
 Public Const ID_ALGERIA = 1580
 Public Const ID_ANGOLA = 1590
 Public Const ID_BENIN = 1600
 Public Const ID_BOTSWANA = 1610
 Public Const ID_BURKINA_FASO = 1620
 Public Const ID_BURUNDI = 1630
 Public Const ID_CAPE_VERDE = 1640
 Public Const ID_CENTRAL_AFRICAN_REPUBLIC = 1650
 Public Const ID_CHAD = 1660
 Public Const ID_COMOROS = 1670
 Public Const ID_REPUBLIC_OF_THE_CONGO = 1680
 Public Const ID_DEMOCRATIC_REPUBLIC_OF_THE_CONGO = 1690
 Public Const ID_DJIBOUTI = 1700
 Public Const ID_EGYPT = 1710
 Public Const ID_EQUATORIAL_GUIANA = 1720
 Public Const ID_ERITREA = 1730
 Public Const ID_ETHIOPIA = 1740
 Public Const ID_GABON = 1750
 Public Const ID_THE_GAMBIA = 1760
 Public Const ID_GHANA = 1770
 Public Const ID_GUIANA = 1780
 Public Const ID_GUIANA_BISSAU = 1790
 Public Const ID_LESOTHO = 1800
 Public Const ID_LIBERIA = 1810
 Public Const ID_LIBYA = 1820
 Public Const ID_MADAGASCAR = 1830
 Public Const ID_MALAWI = 1840
 Public Const ID_MALI = 1850
 Public Const ID_MAURITIUS = 1860
 Public Const ID_MOZAMBIQUE = 1870
 Public Const ID_NIGER = 1880
 Public Const ID_RWANDA = 1890
 Public Const ID_SAO_TOME_AND_PRINCIPE = 1900

Public Const ID_SENEGAL = 1910
 Public Const ID_SEYCHELLES = 1920
 Public Const ID_SIERRA_LEONE = 1930
 Public Const ID_SOMALIA = 1940
 Public Const ID_SUDAN = 1950
 Public Const ID_SWAZILAND = 1960
 Public Const ID_TANZANIA = 1970
 Public Const ID_TOGO = 1980
 Public Const ID_TUNISIA = 1990
 Public Const ID_UGANDA = 2000
 Public Const ID_WESTERN_SAHARA = 2010
 Public Const ID_ZAIRE = 2020
 Public Const ID_ZIMBABWE = 2030
 Public Const ID_AFGHANISTAN = 2040
 Public Const ID_BANGLADESH = 2050
 Public Const ID_BHUTAN = 2060
 Public Const ID_CAMBODIA = 2070
 Public Const ID_EAST_TIMOR = 2080
 Public Const ID_IRAN = 2090
 Public Const ID_JAPAN = 2100
 Public Const ID_JORDAN = 2110
 Public Const ID_KAZAKHSTAN = 2120
 Public Const ID_KYRGYZSTAN = 2130
 Public Const ID_LAOS = 2140
 Public Const ID_LEBANON = 2150
 Public Const ID_MALDIVES = 2160
 Public Const ID_MONGOLIA = 2170
 Public Const ID_MYANMAR = 2180
 Public Const ID_NEPAL = 2190
 Public Const ID_NORTH_KOREA = 2200
 Public Const ID_PAKISTAN = 2210
 Public Const ID_SOUTH_KOREA = 2220
 Public Const ID_SRI_LANKA = 2230
 Public Const ID_SYRIA = 2240
 Public Const ID_TAJIKISTAN = 2250
 Public Const ID_TURKMENISTAN = 2260
 Public Const ID_VIETNAM = 2270
 Public Const ID_YEMEN = 2280
 Public Const ID_AUSTRALIA_NEWZEALAND = 2290

' 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_SUPPORTED = -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_SUPPORTED = -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_DESTINATION = -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_SUPPORTED = -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_JURISTRDICTION_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 a 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
 ' of 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

Public Const BCF_EMUL_RANK = 119

Public Const BCF_EMUL_GENEVA_CODE = 120

Public Const BCF_EMUL_SECURITY_CODE = 121

Public Const BCF_EMUL_CHAMPUS_EFFECTIVE_DATE = 122

Public Const BCF_EMUL_CHAMPUS_EXPIRATION_DATE = 123

Public Const BCF_EMUL_SPONSER_PERSON_PID = 124

Public Const BCF_EMUL_SPONSER_FULL_ = 125

Public Const BCF_EMUL_SPONSER_FIRST_NAME = 126

Public Const BCF_EMUL_SPONSER_MIDDLE_NAME = 127

Public Const BCF_EMUL_SPONSER_LAST_NAME = 128

Public Const BCF_EMUL_SPONSER_NAME_SUFFIX = 129

' 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 codes may sometimes be a very hectic task due to the many image format supports 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 activates the message system inside the libraries so whenever an error occurs, the libraries issue a more detailed error message that explains the failure reason.

Appendix E – Errata Information

The Cimage library currently does not support image saving in 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.

Region Name	Region ID	Country Name	Country ID	Document\State Name	Document \State ID
USA	0	USA	0		
				ALABAMA	0
				ALASKA	1
				ARIZONA	2
				ARKANSAS	3
				CALIFORNIA	4
				COLORADO	5
				CONNECTICUT	6
				DELAWARE	7
				FLORIDA	9
				GEORGIA	10
				HAWAII	54
				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
				US VIRGINISLANDS	91
				UTAH	43
				VERMONT	44
				VIRGINIA	45
				WASHINGTON	46
				WASHINGTON DC	8
				WEST VIRGINIA	47
				WISCONSIN	48
				WYOMING	49
				GREEN CARD	81
				ARMY CARD	82
				SSN CARD	83
				NYPD	84
				GUAM	86
				MEXICO USA	85
				TRIBAL	88
				FIPS ID	89
				PASSPORT CARD	92
CANADA	1	Canada	3		
				ALBERTA	71
				BRITISH COLUMBIA	72
				ONTARIO	70
				PRINCE EDWARD	880
				MANITOBA	73
				NEW BRUNSWICK	74
				NEW FOUNDLAND	75
				NW TERRITORIES	76
				NOVASCOTIA	77
				QUEBEC	1079
				SASKATCHEWAN	78
				CANADA CITIZEN ID	79
AMERICA	2				
		ANTIGUA	108	ANTIGUA	1160
		ARGENTINA	93	ARGENTINA	990

ARUBA	118	ARUBA	1290
BAHAMAS	21	BAHAMAS	250
BARBADOS	133	BARBADOS	1440
BELIZE	106	BELIZE	1120
BERMUDA	13	BERMUDA	170
BOLIVIA	60	BOLIVIA	670
BRAZIL	8	BRAZIL	130
CAYMAN ISLANDS	134	CAYMAN ISLANDS	1450
CHILE	4	CHILE	80
COLUMBIA	79	COLUMBIA	830
COSTA RICA	28	COSTA RICA	320
CUBA	135	CUBA	1460
CURACAO	112	CURACAO	1200
DOMINICAN_REPUBLIC	73	DOMINICAN_REPUBLIC	770
ECUADOR	67	ECUADOR	710
EL_SALVADOR	34	EL_SALVADOR	380
FRENCH_GUIANA	136	FRENCH_GUIANA	1470
GREENLAND	137	GREENLAND	1480
GRENADA	138	GRENADA	1490
GUYANA	139	GUYANA	1500
GUATEMALA	33	GUATEMALA	370
HAITI	74	HAITI	780
HONDURAS	69	HONDURAS	730
JAMAICA	140	JAMAICA	1510
PARAGUAY	150	PARAGUAY	1520
MEXICO	6	MEXICO	100
NICARAGUA	32	NICARAGUA	360
PANAMA	36	PANAMA	400
PERU	29	PERU	330
PUERTO_RICO	30	PUERTO_RICO	340
ST_CHRIST_NEVIS	94	ST_CHRIST_NEVIS	1000
SAINT_KITTS_NEVIS	160	SAINT_KITTS_NEVIS	1530
SAINT_LUCIA	170	SAINT_LUCIA	1540
SAINT_VINCENT_GRENADINES	180	SAINT_VINCENT_GRENADINES	1550
SURINAME	190	SURINAME	1560
TRINIDAD	120	TRINIDAD	1310
TURKS_CAICOS	103	TURKS_CAICOS	1090
URUGUAY	200	URUGUAY	1570
VENEZUELA	80	VENEZUELA	840
VIRGINISLANDS	115	VIRGINISLANDS	1260
EUROPE	3		
ALBANIA	95	ALBANIA	1010
ANDORRA	88	ANDORRA	940

ARMENIA	128	ARMENIA	1390
AUSTRIA	57	AUSTRIA	640
AZERBAIJAN	113	AZERBAIJAN	1240
BELARUS	125	BELARUS	1360
BELGIUM	38	BELGIUM	420
BOSNIA	48	BOSNIA	530
BULGARIA	45	BULGARIA	490
CROATIA	41	CROATIA	450
CYPRUS	76	CYPRUS	800
CZECH	47	CZECH	520
DENMARK	72	DENMARK	760
ESTONIA	71	ESTONIA	750
FINLAND	64	FINLAND	690
FRANCE	5	FRANCE	90
EUROPE_GENERAL_CARDS	46	EUROPE_GENERAL_CARDS	510
GERMANY	10	GERMAN_ID	140
		GERMAN_LIC	141
EUR_GEORGIA	126	EUR_GEORGIA	1370
GREECE	121	GREECE	1320
GUERNSEY	89	GUERNSEY	950
HOLAND	17	HOLAND	210
HUNGARY	49	HUNGARY	540
ISLAND	78	ISLAND	820
IRELAND	54	IRELAND	600
ISLE_OF_MAN	90	ISLE_OF_MAN	960
ISRAEL	9	ISRAEL	120
ITALY	23	ITALY	270
KOSOVO	51	KOSOVO	560
LATVIA	91	LATVIA	970
LIECHTENSTEIN	63	LIECHTENSTEIN	680
LITHUANIA	19	LITHUANIA	230
LUX	18	LUX	220
MACEDONIA	100	MACEDONIA	1060
MALTA	92	MALTA	980
MOLDOVA	105	MOLDOVA	1110
MONACO	129	MONACO	1400
MONTENEGRO	97	MONTENEGRO	1030
NORWAY	15	NORWAY	190
POLAND	27	POLAND	310
PORTUGAL	31	PORTUGAL	350
ROMANIA	12	ROMANIA	160
RUSSIA	58	RUSSIA	650
SAN_MARINO	130	SAN_MARINO	1410
SERBIA	59	SERBIA	660
SLOVAKIA	50	SLOVAKIA	550

	SLOVENIA	53	SLOVENIA	580
	SPAIN	11	SPAIN	150
	SWEDEN	22	SWEDEN	260
	SWISS	20	SWISS	240
	TURKEY	25	TURKEY	290
	UKRAINE	131	UKRAINE	1420
	UNITED_KINGDOM	7	UNITED_KINGDOM AND IRELAND	110
	UZBEKISTAN	127	UZBEKISTAN	1380
	VATICAN	132	VATICAN	1430
AUSTRALIA	4			
			NSW	50
			ACT	51
			QLD	52
			TAS	55
			VIC	53
			WST	56
			SA	57
			NT	58
			COOK ISLANDS	59
			FIJI	61
			KEYPASS	500
			NEWZEALAND (AU)	2290
ASIA	5			
	AFGHANISTAN	970	AFGHANISTAN	2040
	BAHRAIN	56	BAHRAIN	620
	BANGLADESH	680	BANGLADESH	2050
	BHUTAN	690	BHUTAN	2060
	BRUNEI	68	BRUNEI	720
	CAMBODIA	700	CAMBODIA	2070
	CHINA	43	CHINA	470
	EAST_TIMOR	710	EAST_TIMOR	2080
	ISRAEL_DOCS	107	ISRAEL_DOCS	1130-1146
	INDIA	81	INDIA	850
	INDONESIA	37	INDONESIA	410
	IRAN	720	IRAN	2090
	IRAQ	111	IRAQ	1190
	JAPAN	730	JAPAN	2100
	JORDAN	740	JORDAN	2110
	KAZAKHSTAN	750	KAZAKHSTAN	2120
	KUWAIT	116	KUWAIT	1270
	KYRGYZSTAN	760	KYRGYZSTAN	2130
	LAOS	770	LAOS	2140

LEBANON	780	LEBANON	2150
MALAYSIA	2	MALAYSIA	60
MALDIVES	790	MALDIVES	2160
MONGOLIA	800	MONGOLIA	2170
MYANMAR	810	MYANMAR	2180
NEPAL	820	NEPAL	2190
New ZELAND	16	New ZELAND	200
NORTH_KOREA	830	NORTH_KOREA	2200
OMAN	85	OMAN	900
PAKISTAN	840	PAKISTAN	2210
PHILIPPINES	102	PHILIPPINES	1080
QATAR	86	QATAR	910
SAUDI_ARABIA	87	SAUDI_ARABIA	930
SINGAPORE	14	SINGAPORE	180
SOUTH_KOREA	850	SOUTH_KOREA	2220
SRI_LANKA	860	SRI_LANKA	2230
SYRIA	870	SYRIA	2240
TAJIKISTAN	880	TAJIKISTAN	2250
THAILAND	104	THAILAND	1100
TURKMENISTAN	890	TURKMENISTAN	2260
UAE	55	UAE	610
VIETNAM	900	VIETNAM	2270
YEMEN	910	YEMEN	2280
AFRICA	6		
ALGERIA	210	ALGERIA	1580
ANGOLA	220	ANGOLA	1590
BENIN	230	BENIN	1600
BOTSWANA	240	BOTSWANA	1610
BURKINA_FASO	250	BURKINA_FASO	1620
BURUNDI	260	BURUNDI	1630
CAMEROON	124	CAMEROON	1350
CAPE_VERDE	270	CAPE_VERDE	1640
CENTRAL_AFRICAN_REPUBLIC	280	CENTRAL_AFRICAN_REPUBLIC	1650
CHAD	290	CHAD	1660
COMOROS	300	COMOROS	1670
REPUBLIC_OF_THE_CONGO	310	REPUBLIC_OF_THE_CONGO	1680
DEMOCRATIC_REPUBLIC_OF_THE_CONGO	320	DEMOCRATIC_REPUBLIC_OF_THE_CONGO	1690
DJIBOUTI	330	DJIBOUTI	1700
EGYPT	340	EGYPT	1710
EQUATORIAL_GUINEA	350	EQUATORIAL_GUINEA	1720
ERITREA	360	ERITREA	1730

ETHIOPIA	370	ETHIOPIA	1740
IVORY_COAST	114	IVORY_COAST	1250
GABON	380	GABON	1750
THEGAMBIA	390	THEGAMBIA	1760
GHANA	400	GHANA	1770
GUINEA	410	GUINEA	1780
GUINEA_BISSAU	420	GUINEA_BISSAU	1790
KENYA	98	KENYA	1040
LESOTHO	430	LESOTHO	1800
LIBERIA	440	LIBERIA	1810
LIBYA	450	LIBYA	1820
MADAGASCAR	460	MADAGASCAR	1830
MALAWI	470	MALAWI	1840
MALI	480	MALI	1850
MAURITANIA	123	MAURITANIA	1340
MAURITIUS	490	MAURITIUS	1860
MOROCCO	101	MOROCCO	1070
MOZAMBIQUE	500	MOZAMBIQUE	1870
NAMIBIA	82	NAMIBIA	860
NIGER	510	NIGER	1880
NIGERIA	99	NIGERIA	1050
RWANDA	520	RWANDA	1890
SAO_TOME_AND_PRINCIPE	530	SAO_TOME_AND_PRINCIPE	1900
SENEGAL	540	SENEGAL	1910
SEYCHELLES	550	SEYCHELLES	1920
SIERRA_LEONE	560	SIERRA_LEONE	1930
SOMALIA	570	SOMALIA	1940
SOUTH_AFRICA	35	SOUTH_AFRICA	390
SUDAN	580	SUDAN	1950
SWAZILAND	590	SWAZILAND	1960
TANZANIA	600	TANZANIA	1970
TOGO	610	TOGO	1980
TUNISIA	620	TUNISIA	1990
UGANDA	630	UGANDA	2000
WESTERN_SAHARA	640	WESTERN_SAHARA	2010
ZAIRE	650	ZAIRE	2020
ZAMBIA	83	ZAMBIA	870
ZIMBABWE	660	ZIMBABWE	2030
GENERAL_DOC 7			
UNIVERSITY_USA	24	Student Id (UMASS, Boston Un., Emerson Clg., Harvard Un., Northeastern Un., Suffolk Un.)	280

EMPLOYMENT_CARD S	26	EMPLOYMENT_CARDS	300
SERVICE_CARDS	39	SERVICE_CARDS	430
ENTERTAINMENT_CARDS	40	ENTERTAINMENT_CARDS	440
USAPILOTS_CARDS	42	USAPILOTS_CARDS	460
ACCESS_CARDS	44	ACCESS_CARDS	450
OCB_CARDS	52	OCB_CARDS	570
SPAIN_POLICE_CARDS	61	SPAIN_POLICE_CARDS	630
EHIC_CARDS	65	EHIC_CARDS	700
SCSIUSAC_CARDS	70	SCSIUSAC_CARDS	740
USAA_CARDS	75	USAA_CARDS	790
AMPORT_CARDS	77	AMPORT_CARDS	810
PH_CARDS_CARDS	84	PH_CARDS_CARDS	890
IRELAND_FIREARM_CARDS	96	IRELAND_FIREARM_CARDS	1020
TUNISIA_ELECTION_CARDS	109	TUNISIA_ELECTION_CARDS	1170
BEAUCE_CARDS	110	BEAUCE_CARDS	1180
INTERPOL_CARDS	117	INTERPOL_CARDS	1280
T_MOBILE_CARDS	119	T_MOBILE_CARDS	1300
EASYPAY_CARDS	122	EASYPAY_CARDS	1330

Appendix G – Supported Passports 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	Supported Device
Albania	Scanner
Australia	SnapShell, Scanner
Austria	SnapShell, Scanner
Angola	Scanner
Argentina	Scanner
Azerbaijan	SnapShell, Scanner
Belarus	SnapShell, Scanner
Belgium	SnapShell, Scanner
Bolivia	SnapShell
Bosnian	SnapShell, Scanner
Brazil	SnapShell, Scanner
Bulgaria	SnapShell, Scanner
Canada	SnapShell, Scanner
Chile	SnapShell, Scanner
China	SnapShell, Scanner
Croatia	SnapShell, Scanner

Czech	SnapShell, Scanner
Denmark	SnapShell, Scanner
ECUADOR	Scanner
Egypt	SnapShell, Scanner
England	SnapShell, Scanner
France	SnapShell, Scanner
Georgia	SnapShell, Scanner
Germany	SnapShell, Scanner
HID	SnapShell
Holland	SnapShell, Scanner
Hungary	SnapShell, Scanner
India	SnapShell
Indonesia	SnapShell
Ireland	SnapShell, Scanner
Israel	SnapShell, Scanner
Italy	SnapShell, Scanner
Japan	SnapShell, Scanner
Kazakhstan	SnapShell, Scanner
Korea	SnapShell, Scanner
Kuwait	SnapShell
Latvia	SnapShell, Scanner
Lebanon	SnapShell, Scanner
Lithuania	SnapShell, Scanner
Malaysia	SnapShell, Scanner
Mexico	SnapShell, Scanner
Moldova	SnapShell, Scanner
New Zealand	SnapShell
Nigeria	SnapShell, Scanner
Scanner	
Norway	SnapShell, Scanner
Oman	SnapShell
Pakistan	SnapShell
Philippines	SnapShell
Poland	SnapShell, Scanner
Portugal	SnapShell, Scanner
Qatar	SnapShell, Scanner
Romania	SnapShell, Scanner
Russia	SnapShell, Scanner
S_Africa	SnapShell, Scanner
Slovenia	SnapShell, Scanner
Spain	SnapShell, Scanner
SriLanka	SnapShell, Scanner
Suisse	SnapShell, Scanner
Sweden	SnapShell, Scanner
Tunisia	SnapShell, Scanner
Turkey	SnapShell, Scanner
Ukraine	SnapShell, Scanner

