



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

ReceiptCOM

Receipt SDK Library Description

SDK Version 9.60

January 2012

Table of Context

| | |
|---|----|
| TABLE OF CONTEXT | 2 |
| INTRODUCTION | 3 |
| RETRIEVING INFORMATION FROM RECEIPT | 3 |
| FUNCTIONS AND PROPERTIES SUMMERY | 4 |
| LICENSING | 4 |
| DISTRIBUTION | 4 |
| RECEIPTCOM LIBRARY FUNCTIONS | 6 |
| <i>InitReceiptLibrary</i> | 6 |
| <i>CalibrateScanner</i> | 7 |
| <i>Clean</i> | 7 |
| <i>ScanToFile</i> | 8 |
| <i>ScanToFileEx</i> | 9 |
| <i>ProcessReceipt</i> | 10 |
| <i>GetReceiptData</i> | 11 |
| <i>GetReceiptGetRawData</i> | 13 |
| <i>ConvertImage</i> | 13 |
| <i>ReformatImage</i> | 15 |
| <i>RotateImage</i> | 17 |
| LIBRARY PROPERTIES | 19 |
| <i>PaperInTray</i> | 19 |
| <i>PressedButton</i> | 19 |
| <i>ScannerType</i> | 19 |
| <i>IsNeedCalibration</i> | 20 |
| <i>IsScannerValid</i> | 21 |
| <i>JpegQuality</i> | 21 |
| <i>ScanProg</i> | 21 |
| <i>ScanWidth</i> | 21 |
| <i>ScanHeight</i> | 22 |
| USING RECEIPTCOM WITH SCANW | 23 |
| APPENDIX A – RECEIPTCOM SDK INSTALLATION AND REGISTRATION | 24 |
| <i>Installing the ReceiptCOM SDK package</i> | 24 |
| <i>Obtaining the full SDK package</i> | 25 |
| APPENDIX B – CONSTANT DEFINITIONS | 25 |
| <i>Library SlibEx constants</i> | 25 |
| <i>License related constants</i> | 26 |
| <i>Library CImage constants</i> | 26 |
| <i>Receipt library constants</i> | 27 |

Introduction

The ReceiptCOM library is an SDK that gives your application an easy way to extract data from Receipts. Using the library, the integrator can detect a paper insertion to the scanner, scan the receipt, process the image and get the different data fields from the card.

ReceiptCOM may work as a stand-alone SDK or integrate with ScanW SDK, thus giving a complete solution to scan receipts and driver license.

Retrieving information from receipt

Retrieving the data from a receipt involves 3 steps:

- Loading the internal image by scanning the receipt.
- Processing the internal image – use ProcessReceipt to activate the receipt analyzer on the scanner's internal image.
 1. Retrieve the data – use GetReceiptData function to retrieve the detected data

IMPORTANT:

The internal image is overwritten in every new scan.

Functions and properties summary

| Name | Type | Functionality |
|----------------------|----------|--|
| PaperInTray | Property | Check is a document is in the scanner tray |
| PressedButton | Property | Return which of the 3 scanner's button in pressed (for ScanShell1000 only) |
| ScannerType | Property | Returns the connected scanner type (Scanshell 800/800N/1000) |
| IsNeedCalibration | Property | Check if the scanner needs to be calibrated |
| IsScannerValid | Property | Verify the scanner functionality |
| JpgQuality | Property | Set the JPEG compression factor |
| ScanProg | Property | Returns the scan progress (when using ScanToFileEx) |
| InitReceiptLibrary | Function | Activate and enables the library functionality |
| CalibrateScanner | Function | Calibrate the scanner color sensor |
| Clean | Function | Clean the scanner. |
| ScanToFile | Function | Scan document to a bitmap file |
| ScanToFileEx | Function | Scan document while showing progress bar |
| ProcessReceipt | Function | Process the recently scanned receipt |
| GetReceiptData | Function | Returns a specific data item value. |
| GetReceiptGetRawData | Function | Returns the raw text data on the card |
| ConvertImage | Function | Image file conversion function |
| ReformatImage | Function | Image file conversion function |
| RotateImage | Function | Image file conversion function |

Licensing

The use of ReceiptCOM library *must* to be under two conditions:

- Running while the scanner (Scanshell scanners) is connected to the PC as the scanner also functions as a hardware plug.
- Initializing the library with the license before making any other library function calls. See more details in the section that describes the function *InitReceiptLibrary*.

Distribution

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

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

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

- ReceiptCOM.dll
- ScanW.dll (Com object)
- ScanWEx.dll (Com object)

Note:

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

Properties and Methods

ReceiptCOM Library Functions

InitReceiptLibrary

Format

```
InitReceiptLibrary (License As String) As Long
```

Parameters

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

Return

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

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

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

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

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

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

Remarks

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

CalibrateScanner

Format

```
CalibrateScanner () As Long
```

Return value

The function returns 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.

Remarks

This function calibrates the scanner using the calibration card. The calibration results are stored in a file inside the windows directory.

Clean

Format

```
Clean () As Long
```

Return value

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

Remarks

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

ScanToFile

Format

ScanToFile ([FileName](#) As String) As Long

Parameters

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

Return

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

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

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

LICENSE_INVALID – Library was not initialized with proper license.

SLIB_ERR_SCANNER_NOT_FOUND – No attached scanner was found.

SLIB_ERR_SCANNER_GENERAL_FAIL

SLIB_ERR_SCANNER_NOT_FOUND

SLIB_ERR_HARDWARE_ERROR

SLIB_ERR_PAPER_FED_ERROR

SLIB_ERR_SCANABORT

SLIB_ERR_NO_PAPER

SLIB_ERR_PAPER_JAM

SLIB_ERR_FILE_IO_ERROR

SLIB_ERR_PRINTER_PORT_USED

SLIB_ERR_OUT_OF_MEMORY

Remarks

Scan document to the internal image buffer and, in the same time, export it to a bitmap file named “File Name” in the local disk under.

Notice that the scanner is automatically set to scan the image in True color and 300 dpi for optimal OCR recognition.

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

- **Rotation** – use [RotateImage\(\)](#) to rotate the internal image in 90,180 or 270 degrees.
- **Color Scheme** – Modify the internal image color to Gray or black and white images using [ReformatImage\(\)](#).
- **Resolution** – Modify the internal image resolution to any resolution using [ReformatImage\(\)](#).

- **Saving format** – Save the internal image to external file in one of 7 popular file formats using either [ConvertImage\(\)](#) or [ReformatImage\(\)](#) or [RotateImage\(\)](#)

ScanToFileEx

Format

ScanToFileEx (FileName As String) As

Parameters

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

Return

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

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

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

LICENSE_INVALID – Library was not initialized with proper license.

SLIB_ERR_SCANNER_NOT_FOUND – No attached scanner was found.

SLIB_ERR_SCANNER_GENERAL_FAIL

SLIB_ERR_SCANNER_NOT_FOUND

SLIB_ERR_HARDWARE_ERROR

SLIB_ERR_PAPER_FED_ERROR

SLIB_ERR_SCANABORT

SLIB_ERR_NO_PAPER

SLIB_ERR_PAPER_JAM

SLIB_ERR_FILE_IO_ERROR

SLIB_ERR_PRINTER_PORT_USED

SLIB_ERR_OUT_OF_MEMORY

Remarks

Scan document to the internal image buffer and, in the same time, export it to a bitmap file named “File Name” in the local disk under. . In addition, the function automatically invokes a dialog bog with a scanning progress indication.

Notice that the scanner is automatically set to scan the image in True color and 300 dpi for optimal OCR recognition.

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

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

ProcessReceipt

Format

```
ProcessReceipt (fileName As String) As
```

Parameters

[in] [Reserved](#) – Null terminated empty string (“”).

Return

The following are error values (negative values):

LICENSE_INVALID – Library was not initialized with proper license.

INVALID_INTERNAL_IMAGE – The scanner image buffer is empty.

INVALID_INTERNAL_IMAGE – The scanner image buffer is empty.

If none of the above (negative values) is returned than the image was successfully analyzed. In such case the function returns the angle that image was rotated to be positioned in the proper alignment.

Angle_0 – The image detected to be in the proper alignment.

Angle_90 – The image was rotated in 90 degrees.

Angle_180 – The image was rotated in 180 degrees.

Angle_270 – The image was rotated in 270 degrees.

Remarks

Use this function after a business card was scanned. This function takes the image from the scanner’s internal buffer, rotate it automatically to the proper alignment and analyze the image. In return, the function returns the amount of 90 degrees rotations that the image needed to be rotated to be placed in the proper alignment. This value will help you to decide if the displayed images needs to be refreshed.

GetReceiptData

Format

```
GetReceiptData (type As Integer, index As Integer,  
                value As String, label As String ) As Long
```

Parameters

- [in] **type** – Field descriptor enum.
- [in] **index** – Data type index (default to 0. For multi-line fields us this index to retrieved the successive lines that belong to the field).
- [out] **value** – The field value.
- [out] **label** – The field label.

Return

- RECEIPT_INVALID_FIELD** – The value given in *type* is not a valid field descriptor.
- RECEIPT_NO_DATA** – No data exist for this field.
- RECEIPT_LAST_DATA** – Field data retrieved successfully and no additional data exist for this field.
- RECEIPT_NEXT_DATA_EXIST** – Field data retrieved successfully and additional data exist for this field.

Remarks

Use this function after a call to ProcessReceipt was made to retrieve the business card data. The business card data structure may vary from card to card. Some fields contain a single line of data (such as *Name*) and some may contain several lines of data (such as *Address*). In addition, data fields may contain a label in addition to the data. E.g., the field **Tel (972) 9381032** contains a label (“**Tel**”) and a value (“**(972) 9381032**”).

For example, to retrieve the Vendor name us the following call:

```
GetReceiptData(RECEIPT_FIELD_VENDOR, 0, Vname, dummy_string)
```

Note: The name will be loaded into *name*. *dummy_string* will remain empty since a person name does not have any label attached to it.

To retrieve all the Tel numbers from the data structure use the call:

```
GetReceiptData (RECEIPT_FIELD_TEL, 0, tel1, tel_label1)  
GetReceiptData (RECEIPT_FIELD_TEL, 1, tel2, tel_label2)  
GetReceiptData (RECEIPT_FIELD_TEL, 2, tel3, tel_label3)...
```



6167 Bristol Parkway Suite 330, Culver City, CA 90230

Phone: 213-867-2625, 213-867-2630 Fax: 419-735-2419

Continue calling this function until it returns **BIZ_LAST_DATA**.

GetReceiptGetRawData

Format

```
GetReceiptGetRawData (data As String)
```

Parameters

[out] **data** – A buffer to store the data.

Return

None\Ignore.

Remarks

Use this function to obtain the raw data as retrieved by the OCR. The data is copied into *data* as a single text bulk.

ConvertImage

Format

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

Parameters

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

Return

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

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

- **LICENSE_INVALID** – Library was not initialized with proper license.
- **IMG_ERR_BAD_DESTINATION** – Bad destination parameter (the destination parameter is neither file or clipboard)
- **IMG_ERR_FILE_OPEN** – Cannot open input file. This value is returned if the **SourceImage** string is not empty but it points to a missing or invalid source image file.

- **INVALID_INTERNAL_IMAGE** – This value is returned if the [SourceImage](#) string is empty but no document was scanned so there is no internal image in the memory.
- **IMG_ERR_FILE_SAVE_TO_FILE** – Cannot save destination file due to invalid destination file or disk save error.

Remarks

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

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

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 _
) As Long
```

Parameters

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

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

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

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

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

Return

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

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

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

Remarks

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

Image reformat can be done either on the image color scheme or the image dpi or both. Notice that changing the image format may lose the image color information (e.g., when converting from 24 bit true color to 256 gray scale). Modifying and image format from 256 gray scales to 24 bit true color will (obviously) not add color to the image but it will save the image with the proper RGB format (no color map) instead of or using 256 gray scale palette.

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

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

RotateImage

Format

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

Parameters

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

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

ANGLE_0 : 0 degrees rotation

ANGLE_90: 90 degrees rotation

ANGLE_180: 180 degrees rotation

ANGLE_270: 270 degrees rotation

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

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

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

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

Return

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

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

- **LICENSE_INVALID** – Library was not initialized with proper license.
- **IMG_ERR_BAD_ANGLE_0** – bad rotation parameter.
- **IMG_ERR_BAD_DESTINATION** – Bad destination parameter (the destination parameter is neither file or clipboard)

- **IMG_ERR_FILE_OPEN** – Cannot open input file. This value is returned if the SourceImage string is not empty but it points to a missing or invalid source image file.
- **INVALID_INTERNAL_IMAGE** – This value is returned if the SourceImage string is empty but no document was scanned so there is no internal image in the memory.
- **IMG_ERR_FILE_SAVE_TO_CLIPBOARD** – Cannot save image to clipboard due to an error.
- **IMG_ERR_FILE_SAVE_TO_FILE** – Cannot save destination file due to invalid destination file or disk save error.

Remarks

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

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

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

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

Library Properties

PaperInTray

Type

Property.

Direction

Read only.

Remarks

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

PressedButton

Type

Property.

Direction

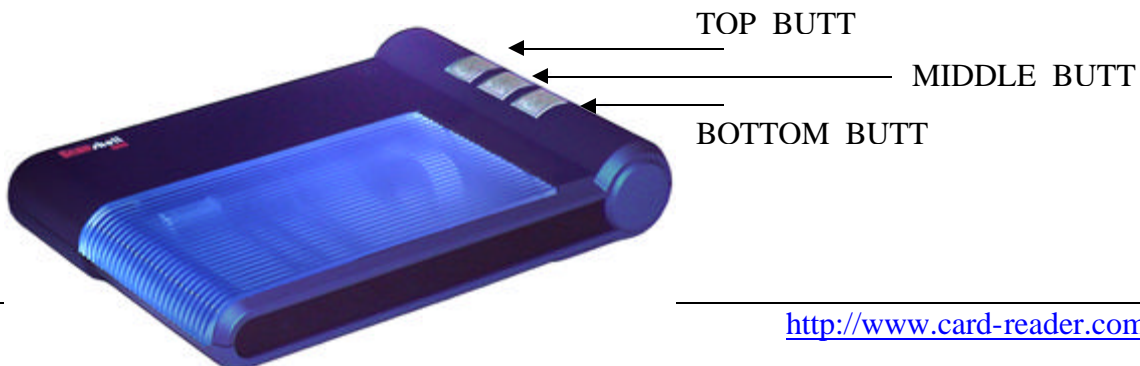
Read only.

Remarks

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

- **TOP_BUTTON**
- **MIDDLE_BUTTON**
- **BOTTOM_BUTTON**

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



Type

Property.

Direction

Read Only.

Remarks

Returns the connected scanner type. This may be one of the following values:

- CSSN_NONE
- CSSN_600
- CSSN_800
- CSSN_800N
- CSSN_1000
- CSSN_2000
- CSSN_2000N
- CSSN_800E
- CSSN_800EN
- CSSN_3000
- CSSN_4000
- CSSN_800G
- CSSN_5000
- CSSN_IDR
- CSSN_800DX
- CSSN_800DXN
- CSSN_FDA
- CSSN_TWN

IsNeedCalibration

Type

Property.

Direction

Read Only.

Remarks

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

IsScannerValid

Type

Property.

Direction

Read Only.

Remarks

Detect if the scanner version is supported by current code. This property is 0 if scanner is not supported and non-zero if it is supported.

JpegQuality

Type

Property.

Direction

Read\Write.

Remarks

This property set\Retrieve the quality factor that is used when converting image in JPEG format.

Range: 11-100 (were 11 yields the smallest image file size and 100 yields the highest image quality).

ScanProg

Type

Property.

Direction

Read.

Remarks

Use this property to retrieve the current scan progress when using the function *ScanToFileEx*. This property has values in the range 0-100.

ScanWidth

Type

Property.

Direction

Read\Write.

Remarks

Use this property to set\retrieve the scan width.

ScanHeight

Type

Property.

Direction

Read\Write.

Remarks

Use this property to set\retrieve the scan Height.

Note: Setting *ScanWidth* and *ScanHeight* both to -1 set the scanner to automatically adjust the scanning area to any document size.

Using ReceiptCOM with ScanW

ReceiptCOM SDK provides a complete and independent solution for scanning receipts. This SDK can also integrate with ScanW SDK (that analyze driver's license and passports) and expand its capabilities to cover receipt reading. In fact, both SDKs are designed to share most of their files. For example, when using both SDKs, you may control the scanner either using ReceiptCOM (using ScanToFile function) or using the scanning library SlibEx that is encapsulated in ScanW (using the function ScanToFile). Both SDKs will actually run the same code eventually. The only motivation to this redundancy is the will to keep ReceiptCOM as independent library. To work with the two libraries (or in other words, to add a business card detection capability to the ScanW library) you must install both SDKs to the same directory and register ScanW.dll and ReceiptCOM.dll.

Appendix A – ReceiptCOM SDK installation and Registration

Installing the ReceiptCOM SDK package

The ReceiptCOM SDK files are packed in a setup file. Installing the setup file extract the following file list:

| File name | Destination directory | Functionality |
|------------------|---------------------------|--|
| ReceiptCOM.dll | Sdk destination directory | Library wrapper COM object |
| ReceiptLib.dll | Sdk destination directory | Receipt analyzer. |
| dic.dll | Sdk destination directory | Receipt analyzer. |
| EngRotDB.bin | Sdk destination directory | Card alignment template detection |
| imgForm.DLL | Windows\system32 | Image processing tools collection |
| ImageCtrl.dll | Sdk destination directory | Image processing tools collection |
| Lib0.lib | Sdk destination directory | Business card analyzer. |
| Lib1.lib | Sdk destination directory | Business card analyzer. |
| Lib2.lib | Sdk destination directory | Business card analyzer. |
| SOCRdll.dll | Sdk destination directory | OCR library |
| OCR_PreProc.dll | Sdk destination directory | Driver's license image analyser |
| SLib.dll | Sdk destination directory | Controls the scanner activity |
| test5.gar | Sdk destination directory | OCR library |
| test5.n3s | Sdk destination directory | OCR library |
| test5.qnp | Sdk destination directory | OCR library |
| test5.teh | Sdk destination directory | OCR library |
| TOCRR.ini | Sdk destination directory | OCR library |
| TOCRRdll.dll | Sdk destination directory | OCR library |
| TOCRRService.exe | Sdk destination directory | OCR library |
| UsaIDs.bin | Sdk destination directory | USA template collections for auto-state detection |
| Msvbvm50.dll | Sdk destination directory | |
| ATL.DLL | Windows\system32 | Part of Windows system (Installs only if newer) |
| MSI.DLL | Windows\system32 | Part of Windows system (Installs only if newer) |

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

Manual registration 1:

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

Manual registration 2:

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

```
REGSVR32.EXE ReceiptCOM.DLL
```

Obtaining the full SDK package

The full sdk package includes the sdk files, documentation; VB and VC++ sample codes can be obtained from:

<http://www.id-scan.com/FTP/Applications/SDK/SDK.zip>

Appendix B – 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

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 CImage constants

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

Receipt library constants

Public Const BIZ_OK = 0
Public Const RECEIPT_OK = 0
Public Const RECEIPT_FILE_NOT_FOUND = -1
Public Const RECEIPT_CANNOT_LOAD_FILE = -2
Public Const RECEIPT_NO_LINES = -3
Public Const RECEIPT_IMAGE_NOT_LOADED = -4
Public Const RECEIPT_INVALID_FIELD = -5

Public Const RECEIPT_NO_DATA = -6
Public Const RECEIPT_LAST_DATA = -7
Public Const RECEIPT_NEXT_DATA_EXIST = -8

Public Const RECEIPT_IMAGE_TOP = 0 // Analyze top part of the image
Public Const RECEIPT_IMAGE_BOTTOM = 1 // Analyze bottom part of the image

Public Const RECEIPT_FIELD_DATE = 0 // Receipt date
Public Const RECEIPT_FIELD_VENDOR = 1 // Vendor name
Public Const RECEIPT_FIELD_TEL = 2 // Vendor phone number
Public Const RECEIPT_FIELD_EMAIL = 3 // Email address
Public Const RECEIPT_FIELD_WEB = 4 // Vendor web address
Public Const RECEIPT_FIELD_STREET = 5 // Street
Public Const RECEIPT_FIELD_CITY = 6 // City
Public Const RECEIPT_FIELD_STATE = 7 // State
Public Const RECEIPT_FIELD_ZIP = 8 // Zip
Public Const RECEIPT_FIELD_ADDRESS = 9 // Vendor full Address
Public Const RECEIPT_FIELD_PRODUCT_DESCRIPTION = 10 // Item name
Public Const RECEIPT_FIELD_PRODUCT_CATALOG = 11 // Item catalog number
Public Const RECEIPT_FIELD_PRODUCT_QUANTITY = 12 // Item count
Public Const RECEIPT_FIELD_PRODUCT_UNIT_PRICE = 13 // Item total
Public Const RECEIPT_FIELD_PRODUCT_FINAL_PRICE = 14 // Item total
Public Const RECEIPT_FIELD_SUB_TOTAL = 15 // Sub total
Public Const RECEIPT_FIELD_SUB_TOTAL_FORMAT = 16 // Sub total format value
Public Const RECEIPT_FIELD_TAX = 17 // Tax
Public Const RECEIPT_FIELD_TAX_FORMAT = 18 // Tax format value
Public Const RECEIPT_FIELD_TOTAL = 19 // Total price
Public Const RECEIPT_FIELD_TOTAL_FORMAT = 20 // Total price format value
Public Const RECEIPT_FIELD_PAYMENT_TYPE = 21 // Payment type
Public Const RECEIPT_FIELD_CATEGORY = 22 // Receipt category
Public Const RECEIPT_FIELD_BAR_CODE = 23 // Bar code
Public Const RECEIPT_FIELD_OTHER = 24 // Other Data