Scanquire

4

Generated by Doxygen 1.8.9.1

Fri Jun 26 2015 14:20:31

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 Package DebenuPDFLibraryDLL0915

Classes

- class DLL
- class PDFLibrary

4.2 Package Scanquire

Namespaces

- package Properties
- package Public

Classes

- class HelpDialog
- class MainForm
- class **Program**

4.3 Package Scanquire.Properties

Classes

- class **Resources**

  A strongly-typed resource class, for looking up localized strings, etc.
- class Scanquire

4.4 Package Scanquire.Public

Namespaces

- package Extensions
- package Properties

- package Sharepoint
- package ThirdParty
- package UserControls

Classes

- class BarcodeReaderEx

  Wrapper around DTK's barcode reader to apply the license key.
- class BarcodeRenderers
- class Compression

  GZip Compression/Decompression.
- class **Encryption**

  Wrappers for built in ProtectedData encryption methods.
- class Icons

  A strongly-typed resource class, for looking up localized strings, etc.
- interface ISQArchiver

  Defines the interface between the Scanquire application and
- interface ISQBarcodeEncoder

  Interface for encoding and decoding barcode values
- interface ISQBarcodeReader
- interface ISQCommand

  Empty interface definition of some command to apply to a document, file, image, page, etc.
- interface ISQCommand_Document

  Interface for defining document oriented commands.
- interface ISQCommand_File

  Interface for defining file oriented commands.
- interface ISQCommand_Image

  Interface for defining image oriented commands.
- interface ISQCommand_Page

  Interface for defining page oriented commands.
- interface ISQCommandImageBuilder

  Definition of an object that can create an command based SQImage.
- interface ISQCommandReader
- interface ISQFileReader

  Definition of an object that can read a file to produce a series of SQImages.
- interface ISQFileWriter

  Definition of an object capable to converting SQDocuments into SQFiles
- interface ISQImageProcessor

  Defines an arbitrary way to modify images
- class SerializedObjectDictionary

  Serializes a directory of object to a local directory.
- class SerializedObjectDictionaryCollection

  Serializes a dictionary of SerializedObjectDictionary to a directory.
- class **Serializer**

  Handles serialization and deserialization of objects.
- class SQArchiverBase

  A base implementation of an ISQArchiver, that handles the majority of common cases.
- class SQArchivers

  Singleton collection of available SQArchivers.
- class SQBarcodeEncoder

The default barcode encoder. Barcode commands are encoded to SQBC{Numeric CommandId}{Value Separator}{Command Text}

- class SQBarcodeEncoders

  Singleton collection of available SQBarcodeEncoders
- class SQBarcodeReader

  Default implementation of an ISQBarcodeReader
- class SQBarcodeReaderBase
- class SQBarcodeReaders

  Singleton collection of all avilable ISQBarcodeReaders
- class SQCommand_Document_IndexField

  A command to specify the value of a named metadata field to apply to a document.
- class SQCommand_Document_OwnerPassword

  A command to specify the owner password of a document.
- class SQCommand_Document_Password

  A command to specify the password of a document.
- class SQCommand_Document_UserPassword

  A command to specify the user password of a document.
- class SQCommand_Image_DrawBarcode

  A command requesting a barcode to be drawn on the specified image.
- class SQCommand_Image_DrawImage

  A command to specify that an image be overlain on the specified image.
- class SQCommand_Image_DrawText

  A command specifying a text value to draw onto a specified image.
- class SQCommand_Image_Size

  A command used to specify an image's drawn size.
- class SQCommand_Page_Bookmark

  For document types that support bookmarks, specify a bookmark be applied to the specified page.
- class SQCommand_Page_Exclude

  Command specifying that the current page should be excluded from the final document.
- class SQCommand_TerminateDocument

  Command specifying that the end of the current document has been reached.
- class SQCommandImageBuilder

  SQCommandImageBuilder that prompts the user to enter a header, footer, comments and bookmarks and applys them to the page.
- class SQCommandImageBuilderBase
- class SQCommandImageBuilders
- class SQCommandReader

  An empty command reader
- class SQCommandReader_Barcode

  ISQCommandReader used for processing barcode base command images.
- class SQCommandReaders

  Singleton collection of available ISQCommandReaders
- class SQDocument

  A collection of SQPages and ISQCommands that apply to the document level.
- class SQFile

  Description of SQFile.
- class SQFile_Pdf

  Specialized type of SQFile for PDF files.
- class SQFileReader_Image

  SQFileReader for reading single or multi-page image formats (jpeg, tiff, etc).
- class SQFileReader_Pdf

Specialized ISQFileReader for reading PDF files.

- class SQFileReaderBase
- class SQFileReaders

  Singleton collection of available ISQFileReaders
- class SQFilesystemArchiver

  A basic archiver for accessing a local or NAS file system.
- class SQFilesystemConnector
- class SQFilesystemConnectors
- class SQFileWriter_Image
- class SQFileWriter_Pdf
- class SQFileWriter_Tiff

  <credit>`http://stackoverflow.com/questions/398388/convert-bitmaps-to-one-multipage-tiff-imag`
- class SQFileWriterBase
- class SQFileWriters

  Manages all configured ISQFileWriters
- class SQImage

  A single SQImage with revision control.
- class SQImageEditLock

  Used to lock an SQImage to allow multiple edit operations.
- class SQImageProcessor_ConvertColorDepth

  Specialized ISQImageProcessor to convert the color depth of an image.
- class SQImageProcessor_RotateFlip

  Specialized ISQImageProcessor to Rotate & Flip an SQImage
- class SQImageProcessors

  Manages all configured ISQImageProcessors.
- class **SQImageRegionExtensions**
- class SQImageRevision

  Represents a single revision of an image.
- class SQPage

  A single logical page containing a single image in a document.
- class SQPhotoAlbumFilesystemArchiver

  Simple archiver for storing photographs to the local filesystem.
- class SQTwain
- class SQTwainSetting
- class SQTwainSettings

  All available SQTwainSettings.
- class SQTwainSettingsDialog

  Dialog window to allow users to select and configure a twain device.
- class **TemporaryFile**

  Class for managing temporary files

Enumerations

- enum SQAcquireIntent { **New**, **Append**, **Insert** }

  The purpose of an acquire command.
- enum SQAcquireSource { **Scanner**, **File**, **Command**, **Custom** }

  The requested image source for an acquire command.
- enum SQImageRegion {
  All, **BottomHalf**, **BottomLeft**, **BottomRight**,
  BottomThird, **TopHalf**, **TopLeft**, **TopRight**,
  TopThird, **MiddleThird** }

  A region of an image.

4.4.1 Enumeration Type Documentation

4.4.1.1 enum Scanquire.Public.SQAcquireIntent

The purpose of an acquire command.

4.4.1.2 enum Scanquire.Public.SQAcquireSource

The requested image source for an acquire command.

4.4.1.3 enum Scanquire.Public.SQImageRegion

A region of an image.

4.5 Package Scanquire.Public.Extensions

Classes

- class **FreeImageBitmapExtensions**

4.6 Package Scanquire.Public.Properties

Classes

- class **Encryption**
- class **SQArchiverSelector**
- class SQFilesystemArchiver
- class SQImage
- class SQImageListViewer
- class SQImageRevision
- class SQImageViewer
- class SQTwain
- class TemporaryFile

4.7 Package Scanquire.Public.Sharepoint

Classes

- class **CAML**
- class SharepointConnector

  Description of SharepointConnector.
- class SharepointCredentials
- class SharepointCredentialsDialog
- class SharepointCredentialsInputDialog
- class SharepointCredentialsManager
- class **SPHelper**

4.8 Package Scanquire.Public.ThirdParty

Classes

- class OAuth2Token
- class ShareFileV3

4.8.1 Detailed Description

Copyright (c) 2014 Citrix Systems, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. The methods in this file will make use of the ShareFile API v3 to show some of the basic operations using GET, POST, PATCH, DELETE HTTP verbs. See api.sharefile.com for more information.

Requirements:

Json.NET library. see `http://json.codeplex.com`

JSON Deserialization:

The sample methods here simply use the JObject data accessors, rather than deserializing to a ShareFile Class representation.

Authentication:

OAuth2 password grant is used for authentication. After the token is acquired it is sent an an authorization header with subsequent API requests.

Exception / Error Checking:

For simplicity, exception handling has not been added. Code should not be used in a production environment.

4.9 Package Scanquire.Public.UserControls

Classes

- class ProgressDialog
- class ProgressMonitor

  Progress bar to monitor progress from a Progress and respond to CancelationToken requests.
- class ProgressMonitorToolstripItem

  Wrapper to allow a ProgressMonitor to be placed on a ToolStrip.
- class SessionLogViewer

  Dialog to view the Trace history for a selected TextWriterTraceListener.
- class SQArchiverSelector

  User dropdown control to select a single SQArchiver from all configured SQArchivers.
- class SQImageListViewer

Custom control for navigating SQImages.

- class SQImageListViewerItem

  A warpper for an SQImage to be added to an SQImageViewer
- class SQSplitContainer

  A standard SplitContainer with a rectangular grabber
- class SQThumbnailPanel

  Panel for displaying thumnbail images.
- class SQThumbnailScrollBar

  Unselectable scrollbar for use with the SQThumbnailPanel
- class SQToolStrip

  A non-selectable ToolStrip that defaults to a borderless Professional render mode.

4.10 Package Testing

Namespaces

- package ImageListViewer
- package Properties

Classes

- class Form1
- class **Program**

4.11 Package Testing.ImageListViewer

Classes

- class SQImageListViewer
- class **SQImageListViewer_SplitContainer**

  A split container with a more intuitive splitter bar.

4.12 Package Testing.Properties

Classes

- class **Resources**

  A strongly-typed resource class, for looking up localized strings, etc.
- class **Settings**

Chapter 5

Class Documentation

5.1 Scanquire.Public.BarcodeReaderEx Class Reference

Wrapper around DTK's barcode reader to apply the license key.

Inheritance diagram for Scanquire.Public.BarcodeReaderEx:

5.1.1 Detailed Description

Wrapper around DTK's barcode reader to apply the license key.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/ThirdParty/BarcodeReaderEx.cs

5.2 Scanquire.Public.BarcodeRenderers Class Reference

Inheritance diagram for Scanquire.Public.BarcodeRenderers:

Properties

- override string **DirectoryPath** `[get]`
- static BarcodeRenderers **Instance** `[get]`

Additional Inherited Members

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/BarcodeRenderer/BarcodeRenderers.cs

5.3 Scanquire.Public.Compression Class Reference

GZip Compression/Decompression.

Static Public Member Functions

- static string **Compress** (string text)
- static string **Decompress** (string compressedText)

5.3.1 Detailed Description

GZip Compression/Decompression.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Compression.cs

5.4 DebenuPDFLibraryDLL0915.DLL Class Reference

Public Member Functions

- void **RegisterForShutdown** (int instanceID)
- **DLL** (string dllFileName)
- void **Release** ()
- void **RegisterForShutdown** (int instanceID)
- **DLL** (string dllFileName)
- void **Release** ()
- void **RegisterForShutdown** (int instanceID)
- **DLL** (string dllFileName)
- void **Release** ()
- void **RegisterForShutdown** (int instanceID)
- **DLL** (string dllFileName)
- void **Release** ()
- void **RegisterForShutdown** (int instanceID)
- **DLL** (string dllFileName)
- void **Release** ()

Public Attributes

- IntPtr **dllHandle** = IntPtr.Zero
- DelegateIIDDD **DebenuPDFLibraryAddArcToPath**
- DelegateIIDDDD **DebenuPDFLibraryAddBoxToPath**
- DelegateIII **DebenuPDFLibraryAddCJKFont**
- DelegateIIDDDDDD **DebenuPDFLibraryAddCurveToPath**
- DelegateIIWW **DebenuPDFLibraryAddEmbeddedFile**
- DelegateIIWI **DebenuPDFLibraryAddFileAttachment**
- DelegateIIIWW **DebenuPDFLibraryAddFormFieldChoiceSub**
- DelegateIIIW **DebenuPDFLibraryAddFormFieldSub**
- DelegateIII **DebenuPDFLibraryAddFormFont**
- DelegateIIWW **DebenuPDFLibraryAddGlobalJavaScript**
- DelegateIIWI **DebenuPDFLibraryAddImageFromFile**
- DelegateIIWIII **DebenuPDFLibraryAddImageFromFileOffset**
- DelegateIIBI **DebenuPDFLibraryAddImageFromString**
- DelegateIIW **DebenuPDFLibraryAddLGIDictToPage**
- DelegateIIDD **DebenuPDFLibraryAddLineToPath**
- DelegateIIDDDDII **DebenuPDFLibraryAddLinkToDestination**
- DelegateIIDDDDIWWII **DebenuPDFLibraryAddLinkToEmbeddedFile**

- DelegateIIDDDDWIDII **DebenuPDFLibraryAddLinkToFile**
- DelegateIIDDDDWI **DebenuPDFLibraryAddLinkToJavaScript**
- DelegateIIDDDDWI **DebenuPDFLibraryAddLinkToLocalFile**
- DelegateIIDDDDIDI **DebenuPDFLibraryAddLinkToPage**
- DelegateIIDDDDWI **DebenuPDFLibraryAddLinkToWeb**
- DelegateIIDDIDDDDWWDDDI **DebenuPDFLibraryAddNoteAnnotation**
- DelegateIIWI **DebenuPDFLibraryAddOpenTypeFontFromFile**
- DelegateIIIIIW **DebenuPDFLibraryAddPageLabels**
- DelegateIIDDDDWI **DebenuPDFLibraryAddSVGAnnotationFromFile**
- DelegateIIDDDDWWI **DebenuPDFLibraryAddSWFAnnotationFromFile**
- DelegateIIWDDDDI **DebenuPDFLibraryAddSeparationColor**
- DelegateIIDDDDIWWDDDI **DebenuPDFLibraryAddStampAnnotation**
- DelegateIII **DebenuPDFLibraryAddStandardFont**
- DelegateIIWIW **DebenuPDFLibraryAddSubsettedFont**
- DelegateIIIDDDD **DebenuPDFLibraryAddTextMarkupAnnotation**
- DelegateIIBBI **DebenuPDFLibraryAddToBuffer**
- DelegateIIWW **DebenuPDFLibraryAddToFileList**
- DelegateIIWI **DebenuPDFLibraryAddTrueTypeFont**
- DelegateIIW **DebenuPDFLibraryAddTrueTypeFontFromFile**
- DelegateIIWWI **DebenuPDFLibraryAddTrueTypeSubsettedFont**
- DelegateIIW **DebenuPDFLibraryAddType1Font**
- DelegateIIDDDDWI **DebenuPDFLibraryAddU3DAnnotationFromFile**
- DelegateIIWW **DebenuPDFLibraryAnalyseFile**
- DelegateII **DebenuPDFLibraryAnnotationCount**
- DelegateII **DebenuPDFLibraryAnsiStringResultLength**
- DelegateIID **DebenuPDFLibraryAppendSpace**
- DelegateIIII **DebenuPDFLibraryAppendTableColumns**
- DelegateIIII **DebenuPDFLibraryAppendTableRows**
- DelegateIIW **DebenuPDFLibraryAppendText**
- DelegateIIW **DebenuPDFLibraryApplyStyle**
- DelegateIII **DebenuPDFLibraryAttachAnnotToForm**
- DelegateII **DebenuPDFLibraryBalanceContentStream**
- DelegateIII **DebenuPDFLibraryBalancePageTree**
- DelegateII **DebenuPDFLibraryBeginPageUpdate**
- DelegateIII **DebenuPDFLibraryCapturePage**
- DelegateIIII **DebenuPDFLibraryCapturePageEx**
- DelegateIII **DebenuPDFLibraryCharWidth**
- DelegateIIWWII **DebenuPDFLibraryCheckFileCompliance**
- DelegateII **DebenuPDFLibraryCheckObjects**
- DelegateII **DebenuPDFLibraryCheckPageAnnots**
- DelegateIIW **DebenuPDFLibraryClearFileList**
- DelegateIII **DebenuPDFLibraryClearImage**
- DelegateII **DebenuPDFLibraryClearPageLabels**
- DelegateII **DebenuPDFLibraryClearTextFormatting**
- DelegateIII **DebenuPDFLibraryCloneOutlineAction**
- DelegateIIIII **DebenuPDFLibraryClonePages**
- DelegateIII **DebenuPDFLibraryCloseOutline**
- DelegateII **DebenuPDFLibraryClosePath**
- DelegateII **DebenuPDFLibraryCombineContentStreams**
- DelegateIIII **DebenuPDFLibraryCompareOutlines**
- DelegateII **DebenuPDFLibraryCompressContent**
- DelegateIII **DebenuPDFLibraryCompressFonts**
- DelegateIII **DebenuPDFLibraryCompressImages**
- DelegateII **DebenuPDFLibraryCompressPage**
- DelegateII **DebenuPDFLibraryContentStreamCount**

- DelegateII **DebenuPDFLibraryContentStreamSafe**
- DelegateIIIW **DebenuPDFLibraryCopyPageRanges**
- DelegateIIIWI **DebenuPDFLibraryCopyPageRangesEx**
- DelegateBII **DebenuPDFLibraryCreateBuffer**
- DelegateI **DebenuPDFLibraryCreateLibrary**
- DelegateII **DebenuPDFLibraryCreateNewObject**
- DelegateIIII **DebenuPDFLibraryCreateTable**
- DelegateIII **DebenuPDFLibraryDAAppendFile**
- DelegateIIII **DebenuPDFLibraryDACapturePage**
- DelegateIIIII **DebenuPDFLibraryDACapturePageEx**
- DelegateIII **DebenuPDFLibraryDACloseFile**
- DelegateIIIIIDDDD **DebenuPDFLibraryDADrawCapturedPage**
- DelegateIIIIIDDDDD **DebenuPDFLibraryDADrawRotatedCapturedPage**
- DelegateIIIW **DebenuPDFLibraryDAEmbedFileStreams**
- DelegateWIIII **DebenuPDFLibraryDAExtractPageText**
- DelegateIIIII **DebenuPDFLibraryDAExtractPageTextBlocks**
- DelegateIIII **DebenuPDFLibraryDAFindPage**
- DelegateIIII **DebenuPDFLibraryDAGetAnnotationCount**
- DelegateIII **DebenuPDFLibraryDAGetFormFieldCount**
- DelegateWIII **DebenuPDFLibraryDAGetFormFieldTitle**
- DelegateWIII **DebenuPDFLibraryDAGetFormFieldValue**
- DelegateBIIII **DebenuPDFLibraryDAGetImageDataToString**
- DelegateDIIIII **DebenuPDFLibraryDAGetImageDblProperty**
- DelegateIIIIII **DebenuPDFLibraryDAGetImageIntProperty**
- DelegateIIII **DebenuPDFLibraryDAGetImageListCount**
- DelegateWIIW **DebenuPDFLibraryDAGetInformation**
- DelegateIII **DebenuPDFLibraryDAGetObjectCount**
- DelegateBIII **DebenuPDFLibraryDAGetObjectToString**
- DelegateDIIIII **DebenuPDFLibraryDAGetPageBox**
- DelegateBIII **DebenuPDFLibraryDAGetPageContentToString**
- DelegateIII **DebenuPDFLibraryDAGetPageCount**
- DelegateDIII **DebenuPDFLibraryDAGetPageHeight**
- DelegateIIII **DebenuPDFLibraryDAGetPageImageList**
- DelegateDIII **DebenuPDFLibraryDAGetPageWidth**
- DelegateDIIII **DebenuPDFLibraryDAGetTextBlockBound**
- DelegateDIIII **DebenuPDFLibraryDAGetTextBlockCharWidth**
- DelegateDIIII **DebenuPDFLibraryDAGetTextBlockColor**
- DelegateIIII **DebenuPDFLibraryDAGetTextBlockColorType**
- DelegateIII **DebenuPDFLibraryDAGetTextBlockCount**
- DelegateWIII **DebenuPDFLibraryDAGetTextBlockFontName**
- DelegateDIII **DebenuPDFLibraryDAGetTextBlockFontSize**
- DelegateWIII **DebenuPDFLibraryDAGetTextBlockText**
- DelegateIIIII **DebenuPDFLibraryDAHasPageBox**
- DelegateIIII **DebenuPDFLibraryDAHidePage**
- DelegateIIIIII **DebenuPDFLibraryDAMovePage**
- DelegateIII **DebenuPDFLibraryDANewPage**
- DelegateIIII **DebenuPDFLibraryDANewPages**
- DelegateIIWW **DebenuPDFLibraryDAOpenFile**
- DelegateIIWW **DebenuPDFLibraryDAOpenFileReadOnly**
- DelegateIIII **DebenuPDFLibraryDAPageRotation**
- DelegateIIII **DebenuPDFLibraryDAReleaseImageList**
- DelegateIII **DebenuPDFLibraryDAReleaseTextBlocks**
- DelegateIII **DebenuPDFLibraryDARemoveUsageRights**
- DelegateIIIIIC **DebenuPDFLibraryDARenderPageToDC**
- DelegateIIIIIIW **DebenuPDFLibraryDARenderPageToFile**

- DelegateBIIIII **DebenuPDFLibraryDARenderPageToString**
- DelegateIIIIII **DebenuPDFLibraryDARotatePage**
- DelegateIIIW **DebenuPDFLibraryDASaveAsFile**
- DelegateIIIIW **DebenuPDFLibraryDASaveImageDataToFile**
- DelegateIIIWW **DebenuPDFLibraryDASetInformation**
- DelegateIIIIDDDD **DebenuPDFLibraryDASetPageBox**
- DelegateIIIIDD **DebenuPDFLibraryDASetPageSize**
- DelegateIIDDDD **DebenuPDFLibraryDASetTextExtractionArea**
- DelegateIIII **DebenuPDFLibraryDASetTextExtractionOptions**
- DelegateIIIDD **DebenuPDFLibraryDASetTextExtractionScaling**
- DelegateIID **DebenuPDFLibraryDASetTextExtractionWordGap**
- DelegateIII **DebenuPDFLibraryDAShiftedHeader**
- DelegateII **DebenuPDFLibraryDecrypt**
- DelegateIIWWW **DebenuPDFLibraryDecryptFile**
- DelegateIII **DebenuPDFLibraryDeleteAnalysis**
- DelegateIII **DebenuPDFLibraryDeleteAnnotation**
- DelegateII **DebenuPDFLibraryDeleteContentStream**
- DelegateIII **DebenuPDFLibraryDeleteFormField**
- DelegateIII **DebenuPDFLibraryDeleteOptionalContentGroup**
- DelegateIII **DebenuPDFLibraryDeletePageLGIDict**
- DelegateIIII **DebenuPDFLibraryDeletePages**
- DelegateIIWW **DebenuPDFLibraryDocJavaScriptAction**
- DelegateII **DebenuPDFLibraryDocumentCount**
- DelegateIIDDDDDII **DebenuPDFLibraryDrawArc**
- DelegateIIDDDDWII **DebenuPDFLibraryDrawBarcode**
- DelegateIIDDDDI **DebenuPDFLibraryDrawBox**
- DelegateIIIDDDD **DebenuPDFLibraryDrawCapturedPage**
- DelegateIIIDDDDDD **DebenuPDFLibraryDrawCapturedPageMatrix**
- DelegateIIDDDI **DebenuPDFLibraryDrawCircle**
- DelegateIIDDDWIII **DebenuPDFLibraryDrawDataMatrixSymbol**
- DelegateIIDDDDI **DebenuPDFLibraryDrawEllipse**
- DelegateIIDDDDDDII **DebenuPDFLibraryDrawEllipticArc**
- DelegateIIDDDW **DebenuPDFLibraryDrawHTMLText**
- DelegateWIDDDDW **DebenuPDFLibraryDrawHTMLTextBox**
- DelegateWIDDWDDDDDD **DebenuPDFLibraryDrawHTMLTextBoxMatrix**
- DelegateIIDDDD **DebenuPDFLibraryDrawImage**
- DelegateIIDDDDDD **DebenuPDFLibraryDrawImageMatrix**
- DelegateIIDDDDDDWI **DebenuPDFLibraryDrawIntelligentMailBarcode**
- DelegateIIDDDD **DebenuPDFLibraryDrawLine**
- DelegateIIDDWW **DebenuPDFLibraryDrawMultiLineText**
- DelegateIIDDWI **DebenuPDFLibraryDrawPDF417Symbol**
- DelegateIIDDWIIIIDD **DebenuPDFLibraryDrawPDF417SymbolEx**
- DelegateIII **DebenuPDFLibraryDrawPath**
- DelegateIII **DebenuPDFLibraryDrawPathEvenOdd**
- DelegateIII **DebenuPDFLibraryDrawPostScriptXObject**
- DelegateIIDDDDDI **DebenuPDFLibraryDrawRotatedBox**
- DelegateIIIDDDDD **DebenuPDFLibraryDrawRotatedCapturedPage**
- DelegateIIDDDDD **DebenuPDFLibraryDrawRotatedImage**
- DelegateIIDDDWW **DebenuPDFLibraryDrawRotatedMultiLineText**
- DelegateIIDDDW **DebenuPDFLibraryDrawRotatedText**
- DelegateIIDDDDDWI **DebenuPDFLibraryDrawRotatedTextBox**
- DelegateIIDDDDDI **DebenuPDFLibraryDrawRoundedBox**
- DelegateIIDDD **DebenuPDFLibraryDrawScaledImage**
- DelegateIIDDDW **DebenuPDFLibraryDrawSpacedText**
- DelegateDIIDDDII **DebenuPDFLibraryDrawTableRows**

- DelegateIIDDW **DebenuPDFLibraryDrawText**
- DelegateIIDDDDWI **DebenuPDFLibraryDrawTextArc**
- DelegateIIDDDDWI **DebenuPDFLibraryDrawTextBox**
- DelegateIIDDWIDDDDDD **DebenuPDFLibraryDrawTextBoxMatrix**
- DelegateIIDDDW **DebenuPDFLibraryDrawWrappedText**
- DelegateII **DebenuPDFLibraryEditableContentStream**
- DelegateIIWWW **DebenuPDFLibraryEmbedFile**
- DelegateII **DebenuPDFLibraryEmbeddedFileCount**
- DelegateII **DebenuPDFLibraryEncapsulateContentStream**
- DelegateIIIIIIIIII **DebenuPDFLibraryEncodePermissions**
- DelegateIIWWII **DebenuPDFLibraryEncrypt**
- DelegateIIWWWWII **DebenuPDFLibraryEncryptFile**
- DelegateIIW **DebenuPDFLibraryEncryptWithFingerprint**
- DelegateII **DebenuPDFLibraryEncryptionAlgorithm**
- DelegateII **DebenuPDFLibraryEncryptionStatus**
- DelegateII **DebenuPDFLibraryEncryptionStrength**
- DelegateII **DebenuPDFLibraryEndPageUpdate**
- DelegateIIIW **DebenuPDFLibraryEndSignProcessToFile**
- DelegateBII **DebenuPDFLibraryEndSignProcessToString**
- DelegateBIWWI **DebenuPDFLibraryExtractFilePageContentToString**
- DelegateWIWWII **DebenuPDFLibraryExtractFilePageText**
- DelegateIIWWII **DebenuPDFLibraryExtractFilePageTextBlocks**
- DelegateIIWWWW **DebenuPDFLibraryExtractFilePages**
- DelegateIIWWWWI **DebenuPDFLibraryExtractFilePagesEx**
- DelegateIIW **DebenuPDFLibraryExtractPageRanges**
- DelegateIII **DebenuPDFLibraryExtractPageTextBlocks**
- DelegateIIII **DebenuPDFLibraryExtractPages**
- DelegateIIW **DebenuPDFLibraryFileListCount**
- DelegateWIWI **DebenuPDFLibraryFileListItem**
- DelegateII **DebenuPDFLibraryFindFonts**
- DelegateIIW **DebenuPDFLibraryFindFormFieldByTitle**
- DelegateII **DebenuPDFLibraryFindImages**
- DelegateIIDDDDIII **DebenuPDFLibraryFitImage**
- DelegateIIDDDDDWI **DebenuPDFLibraryFitRotatedTextBox**
- DelegateIIDDDDWI **DebenuPDFLibraryFitTextBox**
- DelegateIIII **DebenuPDFLibraryFlattenAnnot**
- DelegateIII **DebenuPDFLibraryFlattenFormField**
- DelegateII **DebenuPDFLibraryFontCount**
- DelegateWI **DebenuPDFLibraryFontFamily**
- DelegateII **DebenuPDFLibraryFontHasKerning**
- DelegateWI **DebenuPDFLibraryFontName**
- DelegateWI **DebenuPDFLibraryFontReference**
- DelegateII **DebenuPDFLibraryFontSize**
- DelegateII **DebenuPDFLibraryFontType**
- DelegateII **DebenuPDFLibraryFormFieldCount**
- DelegateIII **DebenuPDFLibraryFormFieldHasParent**
- DelegateIIIWW **DebenuPDFLibraryFormFieldJavaScriptAction**
- DelegateIIIWW **DebenuPDFLibraryFormFieldWebLinkAction**
- DelegateIII **DebenuPDFLibraryGetActionDest**
- DelegateWII **DebenuPDFLibraryGetActionType**
- DelegateWII **DebenuPDFLibraryGetActionURL**
- DelegateWIII **DebenuPDFLibraryGetAnalysisInfo**
- DelegateIII **DebenuPDFLibraryGetAnnotActionID**
- DelegateDIII **DebenuPDFLibraryGetAnnotDblProperty**
- DelegateIII **DebenuPDFLibraryGetAnnotDest**

- DelegateIIII **DebenuPDFLibraryGetAnnotIntProperty**
- DelegateIII **DebenuPDFLibraryGetAnnotQuadCount**
- DelegateDIIII **DebenuPDFLibraryGetAnnotQuadPoints**
- DelegateIIIIW **DebenuPDFLibraryGetAnnotSoundToFile**
- DelegateBIII **DebenuPDFLibraryGetAnnotSoundToString**
- DelegateWIII **DebenuPDFLibraryGetAnnotStrProperty**
- DelegateDIDWI **DebenuPDFLibraryGetBarcodeWidth**
- DelegateWI **DebenuPDFLibraryGetBaseURL**
- DelegateIII **DebenuPDFLibraryGetCSDictEPSG**
- DelegateIII **DebenuPDFLibraryGetCSDictType**
- DelegateWII **DebenuPDFLibraryGetCSDictWKT**
- DelegateCIII **DebenuPDFLibraryGetCanvasDC**
- DelegateWIW **DebenuPDFLibraryGetCatalogInformation**
- DelegateBI **DebenuPDFLibraryGetContentStreamToString**
- DelegateWIW **DebenuPDFLibraryGetCustomInformation**
- DelegateWII **DebenuPDFLibraryGetCustomKeys**
- DelegateWI **DebenuPDFLibraryGetDefaultPrinterName**
- DelegateWII **DebenuPDFLibraryGetDestName**
- DelegateIII **DebenuPDFLibraryGetDestPage**
- DelegateIII **DebenuPDFLibraryGetDestType**
- DelegateDIII **DebenuPDFLibraryGetDestValue**
- DelegateWIW **DebenuPDFLibraryGetDocJavaScript**
- DelegateWI **DebenuPDFLibraryGetDocumentFileName**
- DelegateII **DebenuPDFLibraryGetDocumentFileSize**
- DelegateIII **DebenuPDFLibraryGetDocumentID**
- DelegateWIII **DebenuPDFLibraryGetDocumentIdentifier**
- DelegateWI **DebenuPDFLibraryGetDocumentMetadata**
- DelegateII **DebenuPDFLibraryGetDocumentRepaired**
- DelegateWI **DebenuPDFLibraryGetDocumentResourceList**
- DelegateIIIW **DebenuPDFLibraryGetEmbeddedFileContentToFile**
- DelegateBII **DebenuPDFLibraryGetEmbeddedFileContentToString**
- DelegateIII **DebenuPDFLibraryGetEmbeddedFileID**
- DelegateIIII **DebenuPDFLibraryGetEmbeddedFileIntProperty**
- DelegateWIII **DebenuPDFLibraryGetEmbeddedFileStrProperty**
- DelegateWI **DebenuPDFLibraryGetEncryptionFingerprint**
- DelegateWIWW **DebenuPDFLibraryGetFileMetadata**
- DelegateIII **DebenuPDFLibraryGetFirstChildOutline**
- DelegateII **DebenuPDFLibraryGetFirstOutline**
- DelegateII **DebenuPDFLibraryGetFontEncoding**
- DelegateIII **DebenuPDFLibraryGetFontFlags**
- DelegateIII **DebenuPDFLibraryGetFontID**
- DelegateII **DebenuPDFLibraryGetFontObjectNumber**
- DelegateIIIW **DebenuPDFLibraryGetFormFieldActionID**
- DelegateIII **DebenuPDFLibraryGetFormFieldAlignment**
- DelegateIII **DebenuPDFLibraryGetFormFieldAnnotFlags**
- DelegateDIII **DebenuPDFLibraryGetFormFieldBackgroundColor**
- DelegateIII **DebenuPDFLibraryGetFormFieldBackgroundColorType**
- DelegateDIII **DebenuPDFLibraryGetFormFieldBorderColor**
- DelegateIII **DebenuPDFLibraryGetFormFieldBorderColorType**
- DelegateDIII **DebenuPDFLibraryGetFormFieldBorderProperty**
- DelegateIII **DebenuPDFLibraryGetFormFieldBorderStyle**
- DelegateDIII **DebenuPDFLibraryGetFormFieldBound**
- DelegateWII **DebenuPDFLibraryGetFormFieldCaption**
- DelegateIII **DebenuPDFLibraryGetFormFieldCheckStyle**
- DelegateWII **DebenuPDFLibraryGetFormFieldChildTitle**

- DelegateIII **DebenuPDFLibraryGetFormFieldChoiceType**
- DelegateDIII **DebenuPDFLibraryGetFormFieldColor**
- DelegateIII **DebenuPDFLibraryGetFormFieldComb**
- DelegateWII **DebenuPDFLibraryGetFormFieldDefaultValue**
- DelegateWII **DebenuPDFLibraryGetFormFieldDescription**
- DelegateIII **DebenuPDFLibraryGetFormFieldFlags**
- DelegateWII **DebenuPDFLibraryGetFormFieldFontName**
- DelegateWIIW **DebenuPDFLibraryGetFormFieldJavaScript**
- DelegateIII **DebenuPDFLibraryGetFormFieldKidCount**
- DelegateIII **DebenuPDFLibraryGetFormFieldMaxLen**
- DelegateIII **DebenuPDFLibraryGetFormFieldNoExport**
- DelegateIII **DebenuPDFLibraryGetFormFieldPage**
- DelegateIII **DebenuPDFLibraryGetFormFieldPrintable**
- DelegateIII **DebenuPDFLibraryGetFormFieldReadOnly**
- DelegateIII **DebenuPDFLibraryGetFormFieldRequired**
- DelegateWIIW **DebenuPDFLibraryGetFormFieldRichTextString**
- DelegateIII **DebenuPDFLibraryGetFormFieldRotation**
- DelegateIII **DebenuPDFLibraryGetFormFieldSubCount**
- DelegateWIII **DebenuPDFLibraryGetFormFieldSubDisplayName**
- DelegateWIII **DebenuPDFLibraryGetFormFieldSubName**
- DelegateIIII **DebenuPDFLibraryGetFormFieldSubTempIndex**
- DelegateIII **DebenuPDFLibraryGetFormFieldTabOrder**
- DelegateIIII **DebenuPDFLibraryGetFormFieldTabOrderEx**
- DelegateIIII **DebenuPDFLibraryGetFormFieldTextFlags**
- DelegateDII **DebenuPDFLibraryGetFormFieldTextSize**
- DelegateWII **DebenuPDFLibraryGetFormFieldTitle**
- DelegateIII **DebenuPDFLibraryGetFormFieldType**
- DelegateWII **DebenuPDFLibraryGetFormFieldValue**
- DelegateWIW **DebenuPDFLibraryGetFormFieldValueByTitle**
- DelegateIII **DebenuPDFLibraryGetFormFieldVisible**
- DelegateWIIW **DebenuPDFLibraryGetFormFieldWebLink**
- DelegateII **DebenuPDFLibraryGetFormFontCount**
- DelegateWII **DebenuPDFLibraryGetFormFontName**
- DelegateWIW **DebenuPDFLibraryGetGlobalJavaScript**
- DelegateDIDW **DebenuPDFLibraryGetHTMLTextHeight**
- DelegateIIDW **DebenuPDFLibraryGetHTMLTextLineCount**
- DelegateDIDW **DebenuPDFLibraryGetHTMLTextWidth**
- DelegateIII **DebenuPDFLibraryGetImageID**
- DelegateIII **DebenuPDFLibraryGetImageListCount**
- DelegateBIIII **DebenuPDFLibraryGetImageListItemDataToString**
- DelegateDIIII **DebenuPDFLibraryGetImageListItemDblProperty**
- DelegateIIIII **DebenuPDFLibraryGetImageListItemIntProperty**
- DelegateII **DebenuPDFLibraryGetImageMeasureDict**
- DelegateIIW **DebenuPDFLibraryGetImagePageCount**
- DelegateIIB **DebenuPDFLibraryGetImagePageCountFromString**
- DelegateII **DebenuPDFLibraryGetImagePtDataDict**
- DelegateWII **DebenuPDFLibraryGetInformation**
- DelegateWIII **DebenuPDFLibraryGetInstalledFontsByCharset**
- DelegateWIII **DebenuPDFLibraryGetInstalledFontsByCodePage**
- DelegateIIW **DebenuPDFLibraryGetKerning**
- DelegateWI **DebenuPDFLibraryGetLatestPrinterNames**
- DelegateII **DebenuPDFLibraryGetMaxObjectNumber**
- DelegateIII **DebenuPDFLibraryGetMeasureDictBoundsCount**
- DelegateDIII **DebenuPDFLibraryGetMeasureDictBoundsItem**
- DelegateIII **DebenuPDFLibraryGetMeasureDictCoordinateSystem**

- DelegateIII **DebenuPDFLibraryGetMeasureDictDCSDict**
- DelegateIII **DebenuPDFLibraryGetMeasureDictGCSDict**
- DelegateIII **DebenuPDFLibraryGetMeasureDictGPTSCount**
- DelegateDIII **DebenuPDFLibraryGetMeasureDictGPTSItem**
- DelegateIII **DebenuPDFLibraryGetMeasureDictLPTSCount**
- DelegateDIII **DebenuPDFLibraryGetMeasureDictLPTSItem**
- DelegateIIII **DebenuPDFLibraryGetMeasureDictPDU**
- DelegateIIW **DebenuPDFLibraryGetNamedDestination**
- DelegateIII **DebenuPDFLibraryGetNextOutline**
- DelegateII **DebenuPDFLibraryGetObjectCount**
- DelegateIII **DebenuPDFLibraryGetObjectDecodeError**
- DelegateBII **DebenuPDFLibraryGetObjectToString**
- DelegateII **DebenuPDFLibraryGetOpenActionDestination**
- DelegateWI **DebenuPDFLibraryGetOpenActionJavaScript**
- DelegateII **DebenuPDFLibraryGetOptionalContentConfigCount**
- DelegateIIII **DebenuPDFLibraryGetOptionalContentConfigLocked**
- DelegateIII **DebenuPDFLibraryGetOptionalContentConfigOrderCount**
- DelegateIIII **DebenuPDFLibraryGetOptionalContentConfigOrderItemID**
- DelegateWIII **DebenuPDFLibraryGetOptionalContentConfigOrderItemLabel**
- DelegateIIII **DebenuPDFLibraryGetOptionalContentConfigOrderItemLevel**
- DelegateIIII **DebenuPDFLibraryGetOptionalContentConfigOrderItemType**
- DelegateIIII **DebenuPDFLibraryGetOptionalContentConfigState**
- DelegateIII **DebenuPDFLibraryGetOptionalContentGroupID**
- DelegateWII **DebenuPDFLibraryGetOptionalContentGroupName**
- DelegateIII **DebenuPDFLibraryGetOptionalContentGroupPrintable**
- DelegateIII **DebenuPDFLibraryGetOptionalContentGroupVisible**
- DelegateII **DebenuPDFLibraryGetOrigin**
- DelegateIII **DebenuPDFLibraryGetOutlineActionID**
- DelegateDIII **DebenuPDFLibraryGetOutlineColor**
- DelegateIII **DebenuPDFLibraryGetOutlineDest**
- DelegateIII **DebenuPDFLibraryGetOutlineID**
- DelegateWII **DebenuPDFLibraryGetOutlineJavaScript**
- DelegateIII **DebenuPDFLibraryGetOutlineObjectNumber**
- DelegateWII **DebenuPDFLibraryGetOutlineOpenFile**
- DelegateIII **DebenuPDFLibraryGetOutlinePage**
- DelegateIII **DebenuPDFLibraryGetOutlineStyle**
- DelegateWII **DebenuPDFLibraryGetOutlineWebLink**
- DelegateDIII **DebenuPDFLibraryGetPageBox**
- DelegateWII **DebenuPDFLibraryGetPageColorSpaces**
- DelegateBI **DebenuPDFLibraryGetPageContentToString**
- DelegateIII **DebenuPDFLibraryGetPageImageList**
- DelegateWIW **DebenuPDFLibraryGetPageJavaScript**
- DelegateWII **DebenuPDFLibraryGetPageLGIDictContent**
- DelegateII **DebenuPDFLibraryGetPageLGIDictCount**
- DelegateWII **DebenuPDFLibraryGetPageLabel**
- DelegateII **DebenuPDFLibraryGetPageLayout**
- DelegateBIIII **DebenuPDFLibraryGetPageMetricsToString**
- DelegateII **DebenuPDFLibraryGetPageMode**
- DelegateWII **DebenuPDFLibraryGetPageText**
- DelegateII **DebenuPDFLibraryGetPageViewPortCount**
- DelegateIII **DebenuPDFLibraryGetPageViewPortID**
- DelegateIII **DebenuPDFLibraryGetParentOutline**
- DelegateIII **DebenuPDFLibraryGetPrevOutline**
- DelegateWIW **DebenuPDFLibraryGetPrinterBins**
- DelegateBIW **DebenuPDFLibraryGetPrinterDevModeToString**

- DelegateWIW **DebenuPDFLibraryGetPrinterMediaTypes**
- DelegateWI **DebenuPDFLibraryGetPrinterNames**
- DelegateDI **DebenuPDFLibraryGetRenderScale**
- DelegateIIII **DebenuPDFLibraryGetSignProcessByteRange**
- DelegateIII **DebenuPDFLibraryGetSignProcessResult**
- DelegateIII **DebenuPDFLibraryGetStringListCount**
- DelegateWIII **DebenuPDFLibraryGetStringListItem**
- DelegateDIIIII **DebenuPDFLibraryGetTableCellDblProperty**
- DelegateIIIIII **DebenuPDFLibraryGetTableCellIntProperty**
- DelegateWIIIII **DebenuPDFLibraryGetTableCellStrProperty**
- DelegateIII **DebenuPDFLibraryGetTableColumnCount**
- DelegateIII **DebenuPDFLibraryGetTableLastDrawnRow**
- DelegateIII **DebenuPDFLibraryGetTableRowCount**
- DelegateWI **DebenuPDFLibraryGetTempPath**
- DelegateDI **DebenuPDFLibraryGetTextAscent**
- DelegateDIIII **DebenuPDFLibraryGetTextBlockBound**
- DelegateDIIII **DebenuPDFLibraryGetTextBlockCharWidth**
- DelegateDIIII **DebenuPDFLibraryGetTextBlockColor**
- DelegateIIII **DebenuPDFLibraryGetTextBlockColorType**
- DelegateIII **DebenuPDFLibraryGetTextBlockCount**
- DelegateWIII **DebenuPDFLibraryGetTextBlockFontName**
- DelegateDIII **DebenuPDFLibraryGetTextBlockFontSize**
- DelegateWIII **DebenuPDFLibraryGetTextBlockText**
- DelegateDII **DebenuPDFLibraryGetTextBound**
- DelegateDI **DebenuPDFLibraryGetTextDescent**
- DelegateDI **DebenuPDFLibraryGetTextHeight**
- DelegateDI **DebenuPDFLibraryGetTextSize**
- DelegateDIW **DebenuPDFLibraryGetTextWidth**
- DelegateWII **DebenuPDFLibraryGetUnicodeCharactersFromEncoding**
- DelegateDIII **DebenuPDFLibraryGetViewPortBBox**
- DelegateIII **DebenuPDFLibraryGetViewPortMeasureDict**
- DelegateWII **DebenuPDFLibraryGetViewPortName**
- DelegateIII **DebenuPDFLibraryGetViewPortPtDataDict**
- DelegateIII **DebenuPDFLibraryGetViewerPreferences**
- DelegateWIDWW **DebenuPDFLibraryGetWrappedText**
- DelegateWIDWW **DebenuPDFLibraryGetWrappedTextBreakString**
- DelegateDIDW **DebenuPDFLibraryGetWrappedTextHeight**
- DelegateIIDW **DebenuPDFLibraryGetWrappedTextLineCount**
- DelegateII **DebenuPDFLibraryGetXFAFormFieldCount**
- DelegateWII **DebenuPDFLibraryGetXFAFormFieldName**
- DelegateWIW **DebenuPDFLibraryGetXFAFormFieldNames**
- DelegateWIW **DebenuPDFLibraryGetXFAFormFieldValue**
- DelegateBII **DebenuPDFLibraryGetXFAToString**
- DelegateII **DebenuPDFLibraryGlobalJavaScriptCount**
- DelegateWII **DebenuPDFLibraryGlobalJavaScriptPackageName**
- DelegateII **DebenuPDFLibraryHasFontResources**
- DelegateIII **DebenuPDFLibraryHasPageBox**
- DelegateII **DebenuPDFLibraryHidePage**
- DelegateII **DebenuPDFLibraryImageCount**
- DelegateII **DebenuPDFLibraryImageFillColor**
- DelegateII **DebenuPDFLibraryImageHeight**
- DelegateII **DebenuPDFLibraryImageHorizontalResolution**
- DelegateII **DebenuPDFLibraryImageResolutionUnits**
- DelegateII **DebenuPDFLibraryImageType**
- DelegateII **DebenuPDFLibraryImageVerticalResolution**

- DelegateII **DebenuPDFLibraryImageWidth**
- DelegateIIWII **DebenuPDFLibraryImportEMFFromFile**
- DelegateIIII **DebenuPDFLibraryInsertPages**
- DelegateIIIII **DebenuPDFLibraryInsertTableColumns**
- DelegateIIIII **DebenuPDFLibraryInsertTableRows**
- DelegateIII **DebenuPDFLibraryIsAnnotFormField**
- DelegateII **DebenuPDFLibraryIsTaggedPDF**
- DelegateII **DebenuPDFLibraryLastErrorCode**
- DelegateWI **DebenuPDFLibraryLastRenderError**
- DelegateWI **DebenuPDFLibraryLibraryVersion**
- DelegateWI **DebenuPDFLibraryLicenseInfo**
- DelegateII **DebenuPDFLibraryLinearized**
- DelegateIIDI **DebenuPDFLibraryLoadFromCanvasDC**
- DelegateIIWW **DebenuPDFLibraryLoadFromFile**
- DelegateIIBW **DebenuPDFLibraryLoadFromString**
- DelegateII **DebenuPDFLibraryLoadState**
- DelegateIII **DebenuPDFLibraryMergeDocument**
- DelegateIIWW **DebenuPDFLibraryMergeFileList**
- DelegateIIWW **DebenuPDFLibraryMergeFileListFast**
- DelegateIIWWW **DebenuPDFLibraryMergeFiles**
- DelegateIIIIIII **DebenuPDFLibraryMergeTableCells**
- DelegateIIII **DebenuPDFLibraryMoveContentStream**
- DelegateIIII **DebenuPDFLibraryMoveOutlineAfter**
- DelegateIIII **DebenuPDFLibraryMoveOutlineBefore**
- DelegateIII **DebenuPDFLibraryMovePage**
- DelegateIIDD **DebenuPDFLibraryMovePath**
- DelegateIID **DebenuPDFLibraryMultiplyScale**
- DelegateIIIWI **DebenuPDFLibraryNewChildFormField**
- DelegateII **DebenuPDFLibraryNewContentStream**
- DelegateWIW **DebenuPDFLibraryNewCustomPrinter**
- DelegateIIIIIDDDD **DebenuPDFLibraryNewDestination**
- DelegateII **DebenuPDFLibraryNewDocument**
- DelegateIIWI **DebenuPDFLibraryNewFormField**
- DelegateIIWI **DebenuPDFLibraryNewNamedDestination**
- DelegateIIW **DebenuPDFLibraryNewOptionalContentGroup**
- DelegateIIIWID **DebenuPDFLibraryNewOutline**
- DelegateII **DebenuPDFLibraryNewPage**
- DelegateIIII **DebenuPDFLibraryNewPageFromCanvasDC**
- DelegateIII **DebenuPDFLibraryNewPages**
- DelegateIIW **DebenuPDFLibraryNewPostScriptXObject**
- DelegateIIWDDDDDDDDDDI **DebenuPDFLibraryNewRGBAxialShader**
- DelegateIIWW **DebenuPDFLibraryNewSignProcessFromFile**
- DelegateIIBW **DebenuPDFLibraryNewSignProcessFromString**
- DelegateIIIW **DebenuPDFLibraryNewStaticOutline**
- DelegateIIWI **DebenuPDFLibraryNewTilingPatternFromCapturedPage**
- DelegateIIW **DebenuPDFLibraryNoEmbedFontListAdd**
- DelegateII **DebenuPDFLibraryNoEmbedFontListCount**
- DelegateWII **DebenuPDFLibraryNoEmbedFontListGet**
- DelegateII **DebenuPDFLibraryNoEmbedFontListRemoveAll**
- DelegateIII **DebenuPDFLibraryNoEmbedFontListRemoveIndex**
- DelegateIIW **DebenuPDFLibraryNoEmbedFontListRemoveName**
- DelegateIII **DebenuPDFLibraryNormalizePage**
- DelegateIII **DebenuPDFLibraryOpenOutline**
- DelegateII **DebenuPDFLibraryOptionalContentGroupCount**
- DelegateII **DebenuPDFLibraryOutlineCount**

- DelegateWII **DebenuPDFLibraryOutlineTitle**
- DelegateII **DebenuPDFLibraryPageCount**
- DelegateDI **DebenuPDFLibraryPageHeight**
- DelegateIIWW **DebenuPDFLibraryPageJavaScriptAction**
- DelegateII **DebenuPDFLibraryPageRotation**
- DelegateDI **DebenuPDFLibraryPageWidth**
- DelegateIIWIII **DebenuPDFLibraryPrintDocument**
- DelegateIIWIIIW **DebenuPDFLibraryPrintDocumentToFile**
- DelegateIIIIW **DebenuPDFLibraryPrintOptions**
- DelegateIIWWI **DebenuPDFLibraryPrintPages**
- DelegateIIWWIW **DebenuPDFLibraryPrintPagesToFile**
- DelegateIIB **DebenuPDFLibraryReleaseBuffer**
- DelegateIII **DebenuPDFLibraryReleaseImage**
- DelegateIII **DebenuPDFLibraryReleaseImageList**
- DelegateII **DebenuPDFLibraryReleaseLibrary**
- DelegateIII **DebenuPDFLibraryReleaseSignProcess**
- DelegateIII **DebenuPDFLibraryReleaseStringList**
- DelegateIII **DebenuPDFLibraryReleaseTextBlocks**
- DelegateIII **DebenuPDFLibraryRemoveAppearanceStream**
- DelegateIIW **DebenuPDFLibraryRemoveCustomInformation**
- DelegateIII **DebenuPDFLibraryRemoveDocument**
- DelegateIII **DebenuPDFLibraryRemoveEmbeddedFile**
- DelegateIII **DebenuPDFLibraryRemoveFormFieldBackgroundColor**
- DelegateIII **DebenuPDFLibraryRemoveFormFieldBorderColor**
- DelegateIIW **DebenuPDFLibraryRemoveGlobalJavaScript**
- DelegateII **DebenuPDFLibraryRemoveOpenAction**
- DelegateIII **DebenuPDFLibraryRemoveOutline**
- DelegateII **DebenuPDFLibraryRemoveSharedContentStreams**
- DelegateIIW **DebenuPDFLibraryRemoveStyle**
- DelegateII **DebenuPDFLibraryRemoveUsageRights**
- DelegateIII **DebenuPDFLibraryRemoveXFAEntries**
- DelegateIIIIIIW **DebenuPDFLibraryRenderDocumentToFile**
- DelegateIIIIC **DebenuPDFLibraryRenderPageToDC**
- DelegateIIIIIW **DebenuPDFLibraryRenderPageToFile**
- DelegateBIIII **DebenuPDFLibraryRenderPageToString**
- DelegateII **DebenuPDFLibraryReplaceFonts**
- DelegateIIII **DebenuPDFLibraryReplaceImage**
- DelegateIIWW **DebenuPDFLibraryReplaceTag**
- DelegateIII **DebenuPDFLibraryRequestPrinterStatus**
- DelegateIIWWI **DebenuPDFLibraryRetrieveCustomDataToFile**
- DelegateBISI **DebenuPDFLibraryRetrieveCustomDataToString**
- DelegateIII **DebenuPDFLibraryReverseImage**
- DelegateIII **DebenuPDFLibraryRotatePage**
- DelegateIIW **DebenuPDFLibrarySaveFontToFile**
- DelegateIIIIIW **DebenuPDFLibrarySaveImageListItemDataToFile**
- DelegateIIW **DebenuPDFLibrarySaveImageToFile**
- DelegateBI **DebenuPDFLibrarySaveImageToString**
- DelegateII **DebenuPDFLibrarySaveState**
- DelegateIIW **DebenuPDFLibrarySaveStyle**
- DelegateIIW **DebenuPDFLibrarySaveToFile**
- DelegateBI **DebenuPDFLibrarySaveToString**
- DelegateIII **DebenuPDFLibrarySecurityInfo**
- DelegateIII **DebenuPDFLibrarySelectContentStream**
- DelegateIII **DebenuPDFLibrarySelectDocument**
- DelegateIII **DebenuPDFLibrarySelectFont**

- DelegateIII **DebenuPDFLibrarySelectImage**
- DelegateIII **DebenuPDFLibrarySelectPage**
- DelegateIII **DebenuPDFLibrarySelectRenderer**
- DelegateII **DebenuPDFLibrarySelectedDocument**
- DelegateII **DebenuPDFLibrarySelectedFont**
- DelegateII **DebenuPDFLibrarySelectedImage**
- DelegateII **DebenuPDFLibrarySelectedPage**
- DelegateIIIW **DebenuPDFLibrarySetActionURL**
- DelegateIIIDDD **DebenuPDFLibrarySetAnnotBorderColor**
- DelegateIIIDIDD **DebenuPDFLibrarySetAnnotBorderStyle**
- DelegateIIIW **DebenuPDFLibrarySetAnnotContents**
- DelegateIIIID **DebenuPDFLibrarySetAnnotDblProperty**
- DelegateIIIII **DebenuPDFLibrarySetAnnotIntProperty**
- DelegateIIIIDDDDDDDD **DebenuPDFLibrarySetAnnotQuadPoints**
- DelegateIIIDDDD **DebenuPDFLibrarySetAnnotRect**
- DelegateIIIIW **DebenuPDFLibrarySetAnnotStrProperty**
- DelegateIII **DebenuPDFLibrarySetAnsiMode**
- DelegateIIW **DebenuPDFLibrarySetBaseURL**
- DelegateIII **DebenuPDFLibrarySetBlendMode**
- DelegateIIW **DebenuPDFLibrarySetBreakString**
- DelegateIIII **DebenuPDFLibrarySetCSDictEPSG**
- DelegateIIII **DebenuPDFLibrarySetCSDictType**
- DelegateIIIW **DebenuPDFLibrarySetCSDictWKT**
- DelegateIIW **DebenuPDFLibrarySetCairoFileName**
- DelegateIIII **DebenuPDFLibrarySetCapturedPageOptional**
- DelegateIIIIII **DebenuPDFLibrarySetCapturedPageTransparencyGroup**
- DelegateIIWW **DebenuPDFLibrarySetCatalogInformation**
- DelegateIIII **DebenuPDFLibrarySetCharWidth**
- DelegateII **DebenuPDFLibrarySetClippingPath**
- DelegateII **DebenuPDFLibrarySetClippingPathEvenOdd**
- DelegateIIII **DebenuPDFLibrarySetCompatibility**
- DelegateIIB **DebenuPDFLibrarySetContentStreamFromString**
- DelegateIII **DebenuPDFLibrarySetContentStreamOptional**
- DelegateIIDDDD **DebenuPDFLibrarySetCropBox**
- DelegateIIWW **DebenuPDFLibrarySetCustomInformation**
- DelegateIIWD **DebenuPDFLibrarySetCustomLineDash**
- DelegateIIIIIDDDD **DebenuPDFLibrarySetDestProperties**
- DelegateIIIID **DebenuPDFLibrarySetDestValue**
- DelegateIIW **DebenuPDFLibrarySetDocumentMetadata**
- DelegateIIIIW **DebenuPDFLibrarySetEmbeddedFileStrProperty**
- DelegateIIDDD **DebenuPDFLibrarySetFillColor**
- DelegateIIDDDD **DebenuPDFLibrarySetFillColorCMYK**
- DelegateIIWD **DebenuPDFLibrarySetFillColorSep**
- DelegateIIW **DebenuPDFLibrarySetFillShader**
- DelegateIIW **DebenuPDFLibrarySetFillTilingPattern**
- DelegateIII **DebenuPDFLibrarySetFindImagesMode**
- DelegateIII **DebenuPDFLibrarySetFontEncoding**
- DelegateIIIIIIIIII **DebenuPDFLibrarySetFontFlags**
- DelegateIIII **DebenuPDFLibrarySetFormFieldAlignment**
- DelegateIIII **DebenuPDFLibrarySetFormFieldAnnotFlags**
- DelegateIIIDDD **DebenuPDFLibrarySetFormFieldBackgroundColor**
- DelegateIIIDDDD **DebenuPDFLibrarySetFormFieldBackgroundColorCMYK**
- DelegateIIID **DebenuPDFLibrarySetFormFieldBackgroundColorGray**
- DelegateIIIWD **DebenuPDFLibrarySetFormFieldBackgroundColorSep**
- DelegateIIIDDD **DebenuPDFLibrarySetFormFieldBorderColor**

- DelegateIIIDDDD **DebenuPDFLibrarySetFormFieldBorderColorCMYK**
- DelegateIIID **DebenuPDFLibrarySetFormFieldBorderColorGray**
- DelegateIIIWD **DebenuPDFLibrarySetFormFieldBorderColorSep**
- DelegateIIIDIDD **DebenuPDFLibrarySetFormFieldBorderStyle**
- DelegateIIIDDDD **DebenuPDFLibrarySetFormFieldBounds**
- DelegateIIII **DebenuPDFLibrarySetFormFieldCalcOrder**
- DelegateIIIW **DebenuPDFLibrarySetFormFieldCaption**
- DelegateIIIII **DebenuPDFLibrarySetFormFieldCheckStyle**
- DelegateIIIW **DebenuPDFLibrarySetFormFieldChildTitle**
- DelegateIIII **DebenuPDFLibrarySetFormFieldChoiceType**
- DelegateIIIDDD **DebenuPDFLibrarySetFormFieldColor**
- DelegateIIIDDDD **DebenuPDFLibrarySetFormFieldColorCMYK**
- DelegateIIIWD **DebenuPDFLibrarySetFormFieldColorSep**
- DelegateIIII **DebenuPDFLibrarySetFormFieldComb**
- DelegateIIIW **DebenuPDFLibrarySetFormFieldDefaultValue**
- DelegateIIIW **DebenuPDFLibrarySetFormFieldDescription**
- DelegateIIII **DebenuPDFLibrarySetFormFieldFlags**
- DelegateIIII **DebenuPDFLibrarySetFormFieldFont**
- DelegateIIII **DebenuPDFLibrarySetFormFieldHighlightMode**
- DelegateIIIII **DebenuPDFLibrarySetFormFieldIcon**
- DelegateIIIIIIII **DebenuPDFLibrarySetFormFieldIconStyle**
- DelegateIIII **DebenuPDFLibrarySetFormFieldMaxLen**
- DelegateIIII **DebenuPDFLibrarySetFormFieldNoExport**
- DelegateIIII **DebenuPDFLibrarySetFormFieldOptional**
- DelegateIIII **DebenuPDFLibrarySetFormFieldPage**
- DelegateIIII **DebenuPDFLibrarySetFormFieldPrintable**
- DelegateIIII **DebenuPDFLibrarySetFormFieldReadOnly**
- DelegateIIII **DebenuPDFLibrarySetFormFieldRequired**
- DelegateIIIW **DebenuPDFLibrarySetFormFieldResetAction**
- DelegateIIIWW **DebenuPDFLibrarySetFormFieldRichTextString**
- DelegateIIII **DebenuPDFLibrarySetFormFieldRotation**
- DelegateIIIII **DebenuPDFLibrarySetFormFieldSignatureImage**
- DelegateIIII **DebenuPDFLibrarySetFormFieldStandardFont**
- DelegateIIIWW **DebenuPDFLibrarySetFormFieldSubmitAction**
- DelegateIIIWWI **DebenuPDFLibrarySetFormFieldSubmitActionEx**
- DelegateIIII **DebenuPDFLibrarySetFormFieldTabOrder**
- DelegateIIIIIIII **DebenuPDFLibrarySetFormFieldTextFlags**
- DelegateIIID **DebenuPDFLibrarySetFormFieldTextSize**
- DelegateIIIW **DebenuPDFLibrarySetFormFieldValue**
- DelegateIIWW **DebenuPDFLibrarySetFormFieldValueByTitle**
- DelegateIIII **DebenuPDFLibrarySetFormFieldVisible**
- DelegateIIW **DebenuPDFLibrarySetGDIPlusFileName**
- DelegateIIII **DebenuPDFLibrarySetGDIPlusOptions**
- DelegateIIWI **DebenuPDFLibrarySetHTMLBoldFont**
- DelegateIIWI **DebenuPDFLibrarySetHTMLBoldItalicFont**
- DelegateIIWI **DebenuPDFLibrarySetHTMLItalicFont**
- DelegateIIWI **DebenuPDFLibrarySetHTMLNormalFont**
- DelegateIII **DebenuPDFLibrarySetImageAsMask**
- DelegateIIDDDDDD **DebenuPDFLibrarySetImageMask**
- DelegateIIDDDDDDDD **DebenuPDFLibrarySetImageMaskCMYK**
- DelegateIII **DebenuPDFLibrarySetImageMaskFromImage**
- DelegateIII **DebenuPDFLibrarySetImageOptional**
- DelegateIIIII **DebenuPDFLibrarySetImageResolution**
- DelegateIIIW **DebenuPDFLibrarySetInformation**
- DelegateIII **DebenuPDFLibrarySetJPEGQuality**

- DelegateIII **DebenuPDFLibrarySetJavaScriptMode**
- DelegateIIWI **DebenuPDFLibrarySetKerning**
- DelegateIII **DebenuPDFLibrarySetLineCap**
- DelegateIIDDD **DebenuPDFLibrarySetLineColor**
- DelegateIIDDDD **DebenuPDFLibrarySetLineColorCMYK**
- DelegateIIWD **DebenuPDFLibrarySetLineColorSep**
- DelegateIIDD **DebenuPDFLibrarySetLineDash**
- DelegateIIW **DebenuPDFLibrarySetLineDashEx**
- DelegateIII **DebenuPDFLibrarySetLineJoin**
- DelegateIIW **DebenuPDFLibrarySetLineShader**
- DelegateIID **DebenuPDFLibrarySetLineWidth**
- DelegateIIIDDDD **DebenuPDFLibrarySetMarkupAnnotStyle**
- DelegateIIII **DebenuPDFLibrarySetMeasureDictBoundsCount**
- DelegateIIIID **DebenuPDFLibrarySetMeasureDictBoundsItem**
- DelegateIIII **DebenuPDFLibrarySetMeasureDictCoordinateSystem**
- DelegateIIII **DebenuPDFLibrarySetMeasureDictGPTSCount**
- DelegateIIIID **DebenuPDFLibrarySetMeasureDictGPTSItem**
- DelegateIIII **DebenuPDFLibrarySetMeasureDictLPTSCount**
- DelegateIIIID **DebenuPDFLibrarySetMeasureDictLPTSItem**
- DelegateIIIIII **DebenuPDFLibrarySetMeasureDictPDU**
- DelegateIII **DebenuPDFLibrarySetMeasurementUnits**
- DelegateIII **DebenuPDFLibrarySetNeedAppearances**
- DelegateIIIB **DebenuPDFLibrarySetObjectFromString**
- DelegateIIII **DebenuPDFLibrarySetOpenActionDestination**
- DelegateIIIIIDDDD **DebenuPDFLibrarySetOpenActionDestinationFull**
- DelegateIIW **DebenuPDFLibrarySetOpenActionJavaScript**
- DelegateIIW **DebenuPDFLibrarySetOpenActionMenu**
- DelegateIIIII **DebenuPDFLibrarySetOptionalContentConfigLocked**
- DelegateIIIII **DebenuPDFLibrarySetOptionalContentConfigState**
- DelegateIIIW **DebenuPDFLibrarySetOptionalContentGroupName**
- DelegateIIII **DebenuPDFLibrarySetOptionalContentGroupPrintable**
- DelegateIIII **DebenuPDFLibrarySetOptionalContentGroupVisible**
- DelegateIII **DebenuPDFLibrarySetOrigin**
- DelegateIIIDDD **DebenuPDFLibrarySetOutlineColor**
- DelegateIIIID **DebenuPDFLibrarySetOutlineDestination**
- DelegateIIIIIIDDDD **DebenuPDFLibrarySetOutlineDestinationFull**
- DelegateIIIIDI **DebenuPDFLibrarySetOutlineDestinationZoom**
- DelegateIIIW **DebenuPDFLibrarySetOutlineJavaScript**
- DelegateIIIW **DebenuPDFLibrarySetOutlineNamedDestination**
- DelegateIIIW **DebenuPDFLibrarySetOutlineOpenFile**
- DelegateIIIWIIIDDDDI **DebenuPDFLibrarySetOutlineRemoteDestination**
- DelegateIIIII **DebenuPDFLibrarySetOutlineStyle**
- DelegateIIIW **DebenuPDFLibrarySetOutlineTitle**
- DelegateIIIW **DebenuPDFLibrarySetOutlineWebLink**
- DelegateIIIII **DebenuPDFLibrarySetOverprint**
- DelegateIII **DebenuPDFLibrarySetPDFAMode**
- DelegateIIIII **DebenuPDFLibrarySetPNGTransparencyColor**
- DelegateIIW **DebenuPDFLibrarySetPageActionMenu**
- DelegateIIIDDDD **DebenuPDFLibrarySetPageBox**
- DelegateIIB **DebenuPDFLibrarySetPageContentFromString**
- DelegateIIDD **DebenuPDFLibrarySetPageDimensions**
- DelegateIII **DebenuPDFLibrarySetPageLayout**
- DelegateIII **DebenuPDFLibrarySetPageMode**
- DelegateIIW **DebenuPDFLibrarySetPageSize**
- DelegateII **DebenuPDFLibrarySetPageThumbnail**

- DelegateIIIII **DebenuPDFLibrarySetPageTransparencyGroup**
- DelegateIII **DebenuPDFLibrarySetPrecision**
- DelegateIIB **DebenuPDFLibrarySetPrinterDevModeFromString**
- DelegateIII **DebenuPDFLibrarySetRenderCropType**
- DelegateIII **DebenuPDFLibrarySetRenderDCErasePage**
- DelegateIIII **DebenuPDFLibrarySetRenderDCOffset**
- DelegateIIII **DebenuPDFLibrarySetRenderOptions**
- DelegateIID **DebenuPDFLibrarySetRenderScale**
- DelegateIID **DebenuPDFLibrarySetScale**
- DelegateIIIW **DebenuPDFLibrarySetSignProcessField**
- DelegateIIIDDDD **DebenuPDFLibrarySetSignProcessFieldBounds**
- DelegateIIIWI **DebenuPDFLibrarySetSignProcessFieldImageFromFile**
- DelegateIIII **DebenuPDFLibrarySetSignProcessFieldPage**
- DelegateIIIWWW **DebenuPDFLibrarySetSignProcessInfo**
- DelegateIIIWW **DebenuPDFLibrarySetSignProcessPFXFromFile**
- DelegateIIII **DebenuPDFLibrarySetSignProcessPassthrough**
- DelegateIIII **DebenuPDFLibrarySetSignProcessSubFilter**
- DelegateIIIIDDD **DebenuPDFLibrarySetTableBorderColor**
- DelegateIIIIDDDD **DebenuPDFLibrarySetTableBorderColorCMYK**
- DelegateIIIID **DebenuPDFLibrarySetTableBorderWidth**
- DelegateIIIIIIII **DebenuPDFLibrarySetTableCellAlignment**
- DelegateIIIIIIIDDD **DebenuPDFLibrarySetTableCellBackgroundColor**
- DelegateIIIIIIIDDDD **DebenuPDFLibrarySetTableCellBackgroundColorCMYK**
- DelegateIIIIIIIIDDD **DebenuPDFLibrarySetTableCellBorderColor**
- DelegateIIIIIIIIDDDD **DebenuPDFLibrarySetTableCellBorderColorCMYK**
- DelegateIIIIIIIID **DebenuPDFLibrarySetTableCellBorderWidth**
- DelegateIIIIIW **DebenuPDFLibrarySetTableCellContent**
- DelegateIIIIIIIID **DebenuPDFLibrarySetTableCellPadding**
- DelegateIIIIIIIDDD **DebenuPDFLibrarySetTableCellTextColor**
- DelegateIIIIIIIDDDD **DebenuPDFLibrarySetTableCellTextColorCMYK**
- DelegateIIIIIIID **DebenuPDFLibrarySetTableCellTextSize**
- DelegateIIIIID **DebenuPDFLibrarySetTableColumnWidth**
- DelegateIIIIID **DebenuPDFLibrarySetTableRowHeight**
- DelegateIIIIDDD **DebenuPDFLibrarySetTableThinBorders**
- DelegateIIIIDDDD **DebenuPDFLibrarySetTableThinBordersCMYK**
- DelegateIIW **DebenuPDFLibrarySetTempFile**
- DelegateIIW **DebenuPDFLibrarySetTempPath**
- DelegateIII **DebenuPDFLibrarySetTextAlign**
- DelegateIID **DebenuPDFLibrarySetTextCharSpacing**
- DelegateIIDDD **DebenuPDFLibrarySetTextColor**
- DelegateIIDDDD **DebenuPDFLibrarySetTextColorCMYK**
- DelegateIIWD **DebenuPDFLibrarySetTextColorSep**
- DelegateIIDDDD **DebenuPDFLibrarySetTextExtractionArea**
- DelegateIIII **DebenuPDFLibrarySetTextExtractionOptions**
- DelegateIIIDD **DebenuPDFLibrarySetTextExtractionScaling**
- DelegateIID **DebenuPDFLibrarySetTextExtractionWordGap**
- DelegateIII **DebenuPDFLibrarySetTextHighlight**
- DelegateIIDDD **DebenuPDFLibrarySetTextHighlightColor**
- DelegateIIDDDD **DebenuPDFLibrarySetTextHighlightColorCMYK**
- DelegateIIWD **DebenuPDFLibrarySetTextHighlightColorSep**
- DelegateIII **DebenuPDFLibrarySetTextMode**
- DelegateIID **DebenuPDFLibrarySetTextRise**
- DelegateIID **DebenuPDFLibrarySetTextScaling**
- DelegateIIW **DebenuPDFLibrarySetTextShader**
- DelegateIID **DebenuPDFLibrarySetTextSize**

- DelegateIID **DebenuPDFLibrarySetTextSpacing**
- DelegateIII **DebenuPDFLibrarySetTextUnderline**
- DelegateIIDDD **DebenuPDFLibrarySetTextUnderlineColor**
- DelegateIIDDDD **DebenuPDFLibrarySetTextUnderlineColorCMYK**
- DelegateIIWD **DebenuPDFLibrarySetTextUnderlineColorSep**
- DelegateIIWD **DebenuPDFLibrarySetTextUnderlineCustomDash**
- DelegateIIDD **DebenuPDFLibrarySetTextUnderlineDash**
- DelegateIID **DebenuPDFLibrarySetTextUnderlineDistance**
- DelegateIID **DebenuPDFLibrarySetTextUnderlineWidth**
- DelegateIID **DebenuPDFLibrarySetTextWordSpacing**
- DelegateIII **DebenuPDFLibrarySetTransparency**
- DelegateIIII **DebenuPDFLibrarySetViewerPreferences**
- DelegateIIWI **DebenuPDFLibrarySetXFAFormFieldAccess**
- DelegateIIWDDD **DebenuPDFLibrarySetXFAFormFieldBorderColor**
- DelegateIIWI **DebenuPDFLibrarySetXFAFormFieldBorderPresence**
- DelegateIIWD **DebenuPDFLibrarySetXFAFormFieldBorderWidth**
- DelegateIIWW **DebenuPDFLibrarySetXFAFormFieldValue**
- DelegateIIBI **DebenuPDFLibrarySetXFAFromString**
- DelegateIIWII **DebenuPDFLibrarySetupCustomPrinter**
- DelegateIIWWWWWWWWW **DebenuPDFLibrarySignFile**
- DelegateIII **DebenuPDFLibrarySplitPageText**
- DelegateIIDD **DebenuPDFLibraryStartPath**
- DelegateIIWWII **DebenuPDFLibraryStoreCustomDataFromFile**
- DelegateIISBII **DebenuPDFLibraryStoreCustomDataFromString**
- DelegateII **DebenuPDFLibraryStringResultLength**
- DelegateII **DebenuPDFLibraryTestTempPath**
- DelegateIIWWWII **DebenuPDFLibraryTransformFile**
- DelegateIIW **DebenuPDFLibraryUnlockKey**
- DelegateII **DebenuPDFLibraryUnlocked**
- DelegateIII **DebenuPDFLibraryUpdateAndFlattenFormField**
- DelegateIII **DebenuPDFLibraryUpdateAppearanceStream**
- DelegateIII **DebenuPDFLibraryUseKerning**
- DelegateIII **DebenuPDFLibraryUseUnsafeContentStreams**

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire/bin/Debug/ThirdParty/DebenuPdfLibrary915.cs

5.5 Testing.Form1 Class Reference

Inheritance diagram for Testing.Form1:

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

5.5.1 Member Function Documentation

5.5.1.1 override void Testing.Form1.Dispose (bool *disposing*) [protected]

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

The documentation for this class was generated from the following files:

- C:/Code/Scanquire 4/Testing/Form1.cs
- C:/Code/Scanquire 4/Testing/Form1.Designer.cs

5.6 Scanquire.HelpDialog Class Reference

Inheritance diagram for Scanquire.HelpDialog:

Protected Member Functions

- override void **OnShown** (EventArgs e)
- void **LoadHelpFile** ()
- override void Dispose (bool disposing)

  Clean up any resources being used.

5.6.1 Member Function Documentation

5.6.1.1 override void Scanquire.HelpDialog.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

The documentation for this class was generated from the following files:

- C:/Code/Scanquire 4/Scanquire/HelpDialog.cs
- C:/Code/Scanquire 4/Scanquire/HelpDialog.Designer.cs

5.7 Scanquire.Public.Icons Class Reference

A strongly-typed resource class, for looking up localized strings, etc.

Properties

- static global::System.Resources.ResourceManager ResourceManager `[get]`

  Returns the cached ResourceManager instance used by this class.
- static global::System.Globalization.CultureInfo Culture `[get, set]`

  Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.
- static System.Drawing.Bitmap Append `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap BlackWhite `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.

- static System.Drawing.Bitmap CheckDocumentOff `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap CheckDocumentOn `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap CheckUncheckDocument `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap ColorWheel `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Crop `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap DeskewLeft `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap DeskewRight `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Icon E `[get]`

  Looks up a localized resource of type System.Drawing.Icon similar to (Icon).
- static System.Drawing.Bitmap Fill `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap FlipHorizontal `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap FlipVertical `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Help `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Information `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Insert `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Magnify `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap MoveDown `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap MoveUp `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap NewFile `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Pan `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap RedX `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Rotate180 `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Rotate180AndFlipHorizontal `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap RotateLeft `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap RotateRight `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap SaveFile `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap ScaleToOriginalSize `[get]`

Looks up a localized resource of type System.Drawing.Bitmap.

- static System.Drawing.Bitmap ScaleToViewer `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap ScaleToViewerHeight `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap ScaleToViewerWidth `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Scanner `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Select `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Settings `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Thumbnails `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap Undo `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap ZoomIn `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.
- static System.Drawing.Bitmap ZoomOut `[get]`

  Looks up a localized resource of type System.Drawing.Bitmap.

5.7.1 Detailed Description

A strongly-typed resource class, for looking up localized strings, etc.

5.7.2 Property Documentation

5.7.2.1 System.Drawing.Bitmap Scanquire.Public.Icons.Append `[static], [get]`

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.2 System.Drawing.Bitmap Scanquire.Public.Icons.BlackWhite `[static], [get]`

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.3 System.Drawing.Bitmap Scanquire.Public.Icons.CheckDocumentOff `[static], [get]`

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.4 System.Drawing.Bitmap Scanquire.Public.Icons.CheckDocumentOn `[static], [get]`

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.5 System.Drawing.Bitmap Scanquire.Public.Icons.CheckUncheckDocument `[static], [get]`

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.6 System.Drawing.Bitmap Scanquire.Public.Icons.ColorWheel [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.7 System.Drawing.Bitmap Scanquire.Public.Icons.Crop [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.8 global.System.Globalization.CultureInfo Scanquire.Public.Icons.Culture [static], [get], [set]

Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

5.7.2.9 System.Drawing.Bitmap Scanquire.Public.Icons.DeskewLeft [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.10 System.Drawing.Bitmap Scanquire.Public.Icons.DeskewRight [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.11 System.Drawing.Icon Scanquire.Public.Icons.E [static], [get]

Looks up a localized resource of type System.Drawing.Icon similar to (Icon).

5.7.2.12 System.Drawing.Bitmap Scanquire.Public.Icons.Fill [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.13 System.Drawing.Bitmap Scanquire.Public.Icons.FlipHorizontal [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.14 System.Drawing.Bitmap Scanquire.Public.Icons.FlipVertical [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.15 System.Drawing.Bitmap Scanquire.Public.Icons.Help [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.16 System.Drawing.Bitmap Scanquire.Public.Icons.Information [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.17 System.Drawing.Bitmap Scanquire.Public.Icons.Insert [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.18 System.Drawing.Bitmap Scanquire.Public.Icons.Magnify [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.19 System.Drawing.Bitmap Scanquire.Public.Icons.MoveDown [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.20 System.Drawing.Bitmap Scanquire.Public.Icons.MoveUp [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.21 System.Drawing.Bitmap Scanquire.Public.Icons.NewFile [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.22 System.Drawing.Bitmap Scanquire.Public.Icons.Pan [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.23 System.Drawing.Bitmap Scanquire.Public.Icons.RedX [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.24 global.System.Resources.ResourceManager Scanquire.Public.Icons.ResourceManager [static], [get]

Returns the cached ResourceManager instance used by this class.

5.7.2.25 System.Drawing.Bitmap Scanquire.Public.Icons.Rotate180 [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.26 System.Drawing.Bitmap Scanquire.Public.Icons.Rotate180AndFlipHorizontal [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.27 System.Drawing.Bitmap Scanquire.Public.Icons.RotateLeft [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.28 System.Drawing.Bitmap Scanquire.Public.Icons.RotateRight [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.29 System.Drawing.Bitmap Scanquire.Public.Icons.SaveFile [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.30 System.Drawing.Bitmap Scanquire.Public.Icons.ScaleToOriginalSize [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.31 System.Drawing.Bitmap Scanquire.Public.Icons.ScaleToViewer [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.32 System.Drawing.Bitmap Scanquire.Public.Icons.ScaleToViewerHeight [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.33 System.Drawing.Bitmap Scanquire.Public.Icons.ScaleToViewerWidth [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.34 System.Drawing.Bitmap Scanquire.Public.Icons.Scanner [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.35 System.Drawing.Bitmap Scanquire.Public.Icons.Select [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.36 System.Drawing.Bitmap Scanquire.Public.Icons.Settings [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.37 System.Drawing.Bitmap Scanquire.Public.Icons.Thumbnails [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.38 System.Drawing.Bitmap Scanquire.Public.Icons.Undo [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.39 System.Drawing.Bitmap Scanquire.Public.Icons.ZoomIn [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

5.7.2.40 System.Drawing.Bitmap Scanquire.Public.Icons.ZoomOut [static], [get]

Looks up a localized resource of type System.Drawing.Bitmap.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Icons.Designer.cs

5.8 Scanquire.Public.SQCommandImageBuilder.ImageBuilderInputDialog Class Reference

Inheritance diagram for Scanquire.Public.SQCommandImageBuilder.ImageBuilderInputDialog:

Public Member Functions

- override void **ClearInput** ()

Public Attributes

- ValidatingTextBox< string > **BookmarksTextBox**
- ValidatingTextBox< string > **HeaderTextBox**
- ValidatingTextBox< string > **CommentsTextBox**
- ValidatingTextBox< string > **FooterTextBox**

Protected Member Functions

- override void **OnShown** (EventArgs e)

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommandImageBuilder/SQCommandImageBuilder.cs

5.9 Scanquire.Public.ISQArchiver Interface Reference

Defines the interface between the Scanquire application and

Inheritance diagram for Scanquire.Public.ISQArchiver:

Public Member Functions

- IEnumerable< Task< SQImage > > Acquire (SQAcquireIntent intent, SQAcquireSource source, IProgress< ProgressEventArgs > progress, CancellationToken cancelToken)

  Handles retrieving documents from an archiver and feeding them as images into Scanquire
- Task Send (IEnumerable< SQImage > images, IProgress< ProgressEventArgs > progress, CancellationToken cancelToken)

  The initial entry point to send images from Scanquire to the archiver.
- Task Send (SQFile file, int fileNumber, IProgress< ProgressEventArgs > progress, CancellationToken cancelToken)

  Handles sending an individual file (composed of one or more images). Also in handles arbitrary (non image) files sent from a non-imaging application.

5.9.1 Detailed Description

Defines the interface between the Scanquire application and

5.9.2 Member Function Documentation

5.9.2.1 IEnumerable<Task<SQImage> > Scanquire.Public.ISQArchiver.Acquire (SQAcquireIntent *intent*, SQAcquireSource *source*, IProgress< ProgressEventArgs > *progress*, CancellationToken *cancelToken*)

Handles retrieving documents from an archiver and feeding them as images into Scanquire

Parameters

intent	The intent of the retrieval (new doc, append, insert, etc)
source	The expected source (scanner, file, etc) to retrieve the document from.

Implemented in Scanquire.Public.SQArchiverBase.

5.9.2.2 Task Scanquire.Public.ISQArchiver.Send (IEnumerable< SQImage > *images*, IProgress< ProgressEventArgs > *progress*, CancellationToken *cancelToken*)

The initial entry point to send images from Scanquire to the archiver.

Parameters

images	Individual images in the order they are presented in Scanquire

Implemented in Scanquire.Public.SQArchiverBase.

5.9.2.3 Task Scanquire.Public.ISQArchiver.Send (SQFile *file*, int *fileNumber*, IProgress< ProgressEventArgs > *progress*, CancellationToken *cancelToken*)

Handles sending an individual file (composed of one or more images). Also in handles arbitrary (non image) files sent from a non-imaging application.

Parameters

file	File to be sent to the archiver.
fileNumber	Index of the file being sent (for multiple files).

Implemented in Scanquire.Public.SQArchiverBase, and Scanquire.Public.SQFilesystemArchiver.

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQArchiver/ISQArchiver.cs

5.10 Scanquire.Public.ISQBarcodeEncoder Interface Reference

Interface for encoding and decoding barcode values

Inheritance diagram for Scanquire.Public.ISQBarcodeEncoder:

Public Member Functions

- bool TryEncode (ISQCommand command, out string barcodeText, out string caption)

    Attempt to encode a command to a barcode value.
- bool TryParse (string text, out IEnumerable< ISQCommand > commands)

    Attempt to decode a barcode value to a series of commands.

5.10.1 Detailed Description

Interface for encoding and decoding barcode values

5.10.2 Member Function Documentation

5.10.2.1 bool Scanquire.Public.ISQBarcodeEncoder.TryEncode (**ISQCommand** *command,* out string *barcodeText,* out string *caption*)

Attempt to encode a command to a barcode value.

Parameters

command	The command to encode.
barcodeText	If successful, the encoded text value to apply to the barcode.
caption	If successful, the human-readable caption to apply to the barcode.

Returns

True on success, False on failure.

Implemented in Scanquire.Public.SQBarcodeEncoder.

5.10.2.2 bool Scanquire.Public.ISQBarcodeEncoder.TryParse (string *text,* out IEnumerable< **ISQCommand** > *commands*)

Attempt to decode a barcode value to a series of commands.

Parameters

text	The encoded text to decode.
commands	If successful, a list containing all commands decoded from the barcode text.

Returns

True on success, False on failure.

Implemented in Scanquire.Public.SQBarcodeEncoder.

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQBarcodeEncoder/ISQBarcodeEncoder.cs

5.11 Scanquire.Public.ISQBarcodeReader Interface Reference

Inheritance diagram for Scanquire.Public.ISQBarcodeReader:

Public Member Functions

- Task< Barcode[ ]> Read (SQImage image, CancellationToken cToken)

  Read all barcodes from an image.
- Task< string[ ]> ReadText (SQImage image, CancellationToken cToken)

  Read all barcodes from an image.

5.11.1 Member Function Documentation

5.11.1.1 Task< Barcode[ ]> Scanquire.Public.ISQBarcodeReader.Read (SQImage *image,* CancellationToken *cToken*)

Read all barcodes from an image.

Parameters

image	The image to read.

Returns

List of barcodes read from the specified image

Implemented in Scanquire.Public.SQBarcodeReader, and Scanquire.Public.SQBarcodeReaderBase.

5.11.1.2 Task<string[ ]> Scanquire.Public.ISQBarcodeReader.ReadText (SQImage *image,* CancellationToken *cToken*)

Read all barcodes from an image.

Parameters

image	The image to read.

Returns

List of barcodes strings from the specified image

Implemented in Scanquire.Public.SQBarcodeReaderBase.

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQBarcodeReader/ISQBarcodeReader.cs

5.12 Scanquire.Public.ISQCommand Interface Reference

Empty interface definition of some command to apply to a document, file, image, page, etc.

Inheritance diagram for Scanquire.Public.ISQCommand:

5.12.1 Detailed Description

Empty interface definition of some command to apply to a document, file, image, page, etc.

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/ISQCommand.cs

5.13 Scanquire.Public.ISQCommand_Document Interface Reference

Interface for defining document oriented commands.

Inheritance diagram for Scanquire.Public.ISQCommand_Document:

5.13.1 Detailed Description

Interface for defining document oriented commands.

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/ISQCommand_Document.cs

5.14 Scanquire.Public.ISQCommand_File Interface Reference

Interface for defining file oriented commands.

5.14.1 Detailed Description

Interface for defining file oriented commands.

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/ISQCommand_File.cs

5.15 Scanquire.Public.ISQCommand_Image Interface Reference

Interface for defining image oriented commands.

Inheritance diagram for Scanquire.Public.ISQCommand_Image:

5.15.1 Detailed Description

Interface for defining image oriented commands.

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/ISQCommand_Image.cs

5.16 Scanquire.Public.ISQCommand_Page Interface Reference

Interface for defining page oriented commands.

Inheritance diagram for Scanquire.Public.ISQCommand_Page:

5.16.1 Detailed Description

Interface for defining page oriented commands.

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/ISQCommand_Page.cs

5.17 Scanquire.Public.ISQCommandImageBuilder Interface Reference

Definition of an object that can create an command based SQImage.

Inheritance diagram for Scanquire.Public.ISQCommandImageBuilder:

Public Member Functions

- IEnumerable< Task< SQImage > > **Build** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)

5.17.1 Detailed Description

Definition of an object that can create an command based SQImage.

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommandImageBuilder/ISQCommandImageBuilder.cs

5.18 Scanquire.Public.ISQCommandReader Interface Reference

Inheritance diagram for Scanquire.Public.ISQCommandReader:

Public Member Functions

- Task< IList< ISQCommand > > **Read** (SQImage image, int documentNumber, int pageNumber, CancellationToken cToken)

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommandReader/ISQCommandReader.cs

5.19 Scanquire.Public.ISQFileReader Interface Reference

Definition of an object that can read a file to produce a series of SQImages.

Inheritance diagram for Scanquire.Public.ISQFileReader:

Public Member Functions

- IEnumerable< Task< SQImage > > **Read** (SQFile file, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- IEnumerable< Task< SQImage > > **Read** (string path, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- Task< SQImage[ ]> **ReadAll** (SQFile file, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- Task< SQImage[ ]> **ReadAll** (string path, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

5.19.1 Detailed Description

Definition of an object that can read a file to produce a series of SQImages.

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFileReader/ISQFileReader.cs

5.20 Scanquire.Public.ISQFileWriter Interface Reference

Definition of an object capable to converting SQDocuments into SQFiles

Inheritance diagram for Scanquire.Public.ISQFileWriter:

Public Member Functions

- Task< SQFile > **Write** (SQDocument document, IProgress< ProgressEventArgs > progress, Cancellation↩Token cToken)
- IEnumerable< Task< SQFile > > **Write** (IEnumerable< SQDocument > document, IProgress< Progress↩EventArgs > progress, CancellationToken cToken)
- Task< SQFile[ ]> **WriteAll** (IEnumerable< SQDocument > document, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

Properties

- string FileExtension `[get, set]`

  File extension to apply when generating a destination file name.

5.20.1 Detailed Description

Definition of an object capable to converting SQDocuments into SQFiles

5.20.2 Property Documentation

5.20.2.1 string Scanquire.Public.ISQFileWriter.FileExtension `[get], [set]`

File extension to apply when generating a destination file name.

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFileWriter/ISQFileWriter.cs

5.21 Scanquire.Public.ISQImageProcessor Interface Reference

Defines an arbitrary way to modify images

Inheritance diagram for Scanquire.Public.ISQImageProcessor:

Public Member Functions

- void **ProcessImage** (ref SQImage image)

5.21.1 Detailed Description

Defines an arbitrary way to modify images

The documentation for this interface was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQImageProcessor/ISQImageProcessor.cs

5.22 Scanquire.MainForm Class Reference

Inheritance diagram for Scanquire.MainForm:

Protected Member Functions

- override void **OnFormClosing** (FormClosingEventArgs e)
- override async void **OnLoad** (EventArgs e)
- override bool **ProcessCmdKey** (ref Message msg, Keys keyData)
- override void **OnHelpRequested** (HelpEventArgs hevent)
- delegate Task **AcquireTask** (IEnumerable< Task< SQImage >> source)
- async Task **Save** (ISQArchiver archiver)
- void **ShowHelpFile** ()
- override void Dispose (bool disposing)

  Clean up any resources being used.

Protected Attributes

- Properties.Scanquire **DefaultSettings** = Properties.Scanquire.Default

Properties

- ISQArchiver SelectedArchiver `[get]`

  The currently active SQArchiver.
- ISQArchiver LocalArchiver `[get]`

  Specialized archiver to use for all To/From Local Archiver operations.
- bool ClearOnSave `[get, set]`

  If true: Clear the active batch anytime a save operation completes. If false: Active batch will remain after a save operation complates.

5.22.1 Member Function Documentation

5.22.1.1 override void Scanquire.MainForm.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

5.22.2 Property Documentation

5.22.2.1 bool Scanquire.MainForm.ClearOnSave `[get], [set]`

If true: Clear the active batch anytime a save operation completes. If false: Active batch will remain after a save operation complates.

5.22.2.2 ISQArchiver Scanquire.MainForm.LocalArchiver `[get]`

Specialized archiver to use for all To/From Local Archiver operations.

5.22.2.3 ISQArchiver Scanquire.MainForm.SelectedArchiver `[get]`

The currently active SQArchiver.

The documentation for this class was generated from the following files:

- C:/Code/Scanquire 4/Scanquire/MainForm.cs
- C:/Code/Scanquire 4/Scanquire/MainForm.Designer.cs

5.23 Scanquire.Public.SQTwainSettingsDialog.NewSettingDialog Class Reference

Dialog to create a new setting configuration for a device.

Inheritance diagram for Scanquire.Public.SQTwainSettingsDialog.NewSettingDialog:

Public Attributes

- ValidatingTextBox< string > SettingName

  Contains the name of the new setting.
- ListBox ImageProcessorsListBox

  Listing of all available image processors to add to the setting.

Properties

- IEnumerable< string > SelectedImageProcessors `[get, set]`

  The ImageProcessors the user has selected to apply to the setting.

5.23.1 Detailed Description

Dialog to create a new setting configuration for a device.

5.23.2 Member Data Documentation

5.23.2.1 ListBox Scanquire.Public.SQTwainSettingsDialog.NewSettingDialog.ImageProcessorsListBox

Listing of all available image processors to add to the setting.

5.23.2.2 ValidatingTextBox<string> Scanquire.Public.SQTwainSettingsDialog.NewSettingDialog.SettingName

Contains the name of the new setting.

5.23.3 Property Documentation

5.23.3.1 IEnumerable<string> Scanquire.Public.SQTwainSettingsDialog.NewSettingDialog.SelectedImageProcessors `[get]`, `[set]`

The ImageProcessors the user has selected to apply to the setting.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQTwain/SQTwainSettingsDialog.cs

5.24 Scanquire.Public.ThirdParty.OAuth2Token Class Reference

Public Member Functions

- **OAuth2Token** (JObject json)

Properties

- string **AccessToken** `[get, set]`
- string **RefreshToken** `[get, set]`
- string **TokenType** `[get, set]`
- string **Appcp** `[get, set]`
- string **Apicp** `[get, set]`
- string **Subdomain** `[get, set]`
- int **ExpiresIn** `[get, set]`

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/ThirdParty/ShareFile.cs

5.25 DebenuPDFLibraryDLL0915.PDFLibrary Class Reference

Public Types

- enum **AddImageOption** {
  ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
  Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
  ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
  Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
  ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
  Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
  ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
  Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
  ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
  Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6 }
- enum **ImageCompression** {
  None = 0, **Flate** = 1, **None** = 0, **Flate** = 1,
  None = 0, **Flate** = 1, **None** = 0, **Flate** = 1,
  None = 0, **Flate** = 1 }
- enum **DocumentOrigin** {
  BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
  BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
  BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
  BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
  BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3 }
- enum **DocumentProperty** {
  PDFVersion = 0, **Author** = 1, **Title** = 2, **Subject** = 3,
  Keywords = 4, **Creator** = 5, **Producer** = 6, **PDFVersion** = 0,
  Author = 1, **Title** = 2, **Subject** = 3, **Keywords** = 4,
  Creator = 5, **Producer** = 6, **PDFVersion** = 0, **Author** = 1,
  Title = 2, **Subject** = 3, **Keywords** = 4, **Creator** = 5,
  Producer = 6, **PDFVersion** = 0, **Author** = 1, **Title** = 2,
  Subject = 3, **Keywords** = 4, **Creator** = 5, **Producer** = 6,
  PDFVersion = 0, **Author** = 1, **Title** = 2, **Subject** = 3,
  Keywords = 4, **Creator** = 5, **Producer** = 6 }
- enum **Encryption** {
  NONE = -1, **BIT40** = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2,
  BIT256_AES = 3, **NONE** = -1, **BIT40** = 0, **BIT128_RC4** = 1,
  BIT128_AES = 2, **BIT256_AES** = 3, **NONE** = -1, **BIT40** = 0,
  BIT128_RC4 = 1, **BIT128_AES** = 2, **BIT256_AES** = 3, **NONE** = -1,
  BIT40 = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2, **BIT256_AES** = 3,
  NONE = -1, **BIT40** = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2,
  BIT256_AES = 3 }
- enum **ImageIntProperty** {
  ImageType = 400, **PixelWidth** = 401, **PixelHeight** = 402, **ImageType** = 400,
  PixelWidth = 401, **PixelHeight** = 402, **ImageType** = 400, **PixelWidth** = 401,
  PixelHeight = 402, **ImageType** = 400, **PixelWidth** = 401, **PixelHeight** = 402,
  ImageType = 400, **PixelWidth** = 401, **PixelHeight** = 402 }
- enum **ImageDblProperty** {
  TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
  BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
  TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
  BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
  TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
  BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
  TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
  BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
  TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
  BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508 }

- enum **ImageFormat** {
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4 }

- enum **MeasurementUnit** {
Points = 0, **Millimeters** = 1, **Inches** = 2, **Points** = 0,
Millimeters = 1, **Inches** = 2, **Points** = 0, **Millimeters** = 1,
Inches = 2, **Points** = 0, **Millimeters** = 1, **Inches** = 2,
Points = 0, **Millimeters** = 1, **Inches** = 2 }

- enum **Permission** {
PRINT = 1, **COPY** = 2, **CHANGE** = 4, **ADDNOTES** = 8,
FILLFIELDS = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128,
ALL = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL,
PRINT = 1, **COPY** = 2, **CHANGE** = 4,
ADDNOTES = 8, **FILLFIELDS** = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64,
PRINTFULL = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | AS↩
SEMBLE | PRINTFULL, **PRINT** = 1, **COPY** = 2,
CHANGE = 4, **ADDNOTES** = 8, **FILLFIELDS** = 16, **COPYACCESS** = 32,
ASSEMBLE = 64, **PRINTFULL** = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | C↩
OPYACCESS | ASSEMBLE | PRINTFULL, **PRINT** = 1,
COPY = 2, **CHANGE** = 4, **ADDNOTES** = 8, **FILLFIELDS** = 16,
COPYACCESS = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES
| FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL,
PRINT = 1, **COPY** = 2, **CHANGE** = 4, **ADDNOTES** = 8,
FILLFIELDS = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128,
ALL = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL
}

- enum **AddImageOption** {
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6 }

- enum **ImageCompression** {
None = 0, **Flate** = 1, **None** = 0, **Flate** = 1,
None = 0, **Flate** = 1, **None** = 0, **Flate** = 1,
None = 0, **Flate** = 1 }

- enum **DocumentOrigin** {
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3 }

- enum **DocumentProperty** {

PDFVersion = 0, **Author** = 1, **Title** = 2, **Subject** = 3,
Keywords = 4, **Creator** = 5, **Producer** = 6, **PDFVersion** = 0,
Author = 1, **Title** = 2, **Subject** = 3, **Keywords** = 4,
Creator = 5, **Producer** = 6, **PDFVersion** = 0, **Author** = 1,
Title = 2, **Subject** = 3, **Keywords** = 4, **Creator** = 5,
Producer = 6, **PDFVersion** = 0, **Author** = 1, **Title** = 2,
Subject = 3, **Keywords** = 4, **Creator** = 5, **Producer** = 6,
PDFVersion = 0, **Author** = 1, **Title** = 2, **Subject** = 3,
Keywords = 4, **Creator** = 5, **Producer** = 6 }

- enum **Encryption** {
  NONE = -1, **BIT40** = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2,
  BIT256_AES = 3, **NONE** = -1, **BIT40** = 0, **BIT128_RC4** = 1,
  BIT128_AES = 2, **BIT256_AES** = 3, **NONE** = -1, **BIT40** = 0,
  BIT128_RC4 = 1, **BIT128_AES** = 2, **BIT256_AES** = 3, **NONE** = -1,
  BIT40 = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2, **BIT256_AES** = 3,
  NONE = -1, **BIT40** = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2,
  BIT256_AES = 3 }
- enum **ImageIntProperty** {
  ImageType = 400, **PixelWidth** = 401, **PixelHeight** = 402, **ImageType** = 400,
  PixelWidth = 401, **PixelHeight** = 402, **ImageType** = 400, **PixelWidth** = 401,
  PixelHeight = 402, **ImageType** = 400, **PixelWidth** = 401, **PixelHeight** = 402,
  ImageType = 400, **PixelWidth** = 401, **PixelHeight** = 402 }
- enum **ImageDblProperty** {
  TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
  BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
  TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
  BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
  TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
  BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
  TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
  BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
  TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
  BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508 }
- enum **ImageFormat** {
  JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
  JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
  JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
  JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
  JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4 }
- enum **MeasurementUnit** {
  Points = 0, **Millimeters** = 1, **Inches** = 2, **Points** = 0,
  Millimeters = 1, **Inches** = 2, **Points** = 0, **Millimeters** = 1,
  Inches = 2, **Points** = 0, **Millimeters** = 1, **Inches** = 2,
  Points = 0, **Millimeters** = 1, **Inches** = 2 }
- enum **Permission** {
  PRINT = 1, **COPY** = 2, **CHANGE** = 4, **ADDNOTES** = 8,
  FILLFIELDS = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128,
  ALL = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL,
  PRINT = 1, **COPY** = 2, **CHANGE** = 4,
  ADDNOTES = 8, **FILLFIELDS** = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64,
  PRINTFULL = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | AS↩
  SEMBLE | PRINTFULL, **PRINT** = 1, **COPY** = 2,
  CHANGE = 4, **ADDNOTES** = 8, **FILLFIELDS** = 16, **COPYACCESS** = 32,
  ASSEMBLE = 64, **PRINTFULL** = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | C↩
  OPYACCESS | ASSEMBLE | PRINTFULL, **PRINT** = 1,
  COPY = 2, **CHANGE** = 4, **ADDNOTES** = 8, **FILLFIELDS** = 16,
  COPYACCESS = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES

| FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL,
PRINT = 1, **COPY** = 2, **CHANGE** = 4, **ADDNOTES** = 8,
FILLFIELDS = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128,
ALL = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL
}

- enum **AddImageOption** {
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6 }

- enum **ImageCompression** {
None = 0, **Flate** = 1, **None** = 0, **Flate** = 1,
None = 0, **Flate** = 1, **None** = 0, **Flate** = 1,
None = 0, **Flate** = 1 }

- enum **DocumentOrigin** {
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3 }

- enum **DocumentProperty** {
PDFVersion = 0, **Author** = 1, **Title** = 2, **Subject** = 3,
Keywords = 4, **Creator** = 5, **Producer** = 6, **PDFVersion** = 0,
Author = 1, **Title** = 2, **Subject** = 3, **Keywords** = 4,
Creator = 5, **Producer** = 6, **PDFVersion** = 0, **Author** = 1,
Title = 2, **Subject** = 3, **Keywords** = 4, **Creator** = 5,
Producer = 6, **PDFVersion** = 0, **Author** = 1, **Title** = 2,
Subject = 3, **Keywords** = 4, **Creator** = 5, **Producer** = 6,
PDFVersion = 0, **Author** = 1, **Title** = 2, **Subject** = 3,
Keywords = 4, **Creator** = 5, **Producer** = 6 }

- enum **Encryption** {
NONE = -1, **BIT40** = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2,
BIT256_AES = 3, **NONE** = -1, **BIT40** = 0, **BIT128_RC4** = 1,
BIT128_AES = 2, **BIT256_AES** = 3, **NONE** = -1, **BIT40** = 0,
BIT128_RC4 = 1, **BIT128_AES** = 2, **BIT256_AES** = 3, **NONE** = -1,
BIT40 = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2, **BIT256_AES** = 3,
NONE = -1, **BIT40** = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2,
BIT256_AES = 3 }

- enum **ImageIntProperty** {
ImageType = 400, **PixelWidth** = 401, **PixelHeight** = 402, **ImageType** = 400,
PixelWidth = 401, **PixelHeight** = 402, **ImageType** = 400, **PixelWidth** = 401,
PixelHeight = 402, **ImageType** = 400, **PixelWidth** = 401, **PixelHeight** = 402,
ImageType = 400, **PixelWidth** = 401, **PixelHeight** = 402 }

- enum **ImageDblProperty** {

TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508 }

- enum **ImageFormat** {
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4 }
- enum **MeasurementUnit** {
Points = 0, **Millimeters** = 1, **Inches** = 2, **Points** = 0,
Millimeters = 1, **Inches** = 2, **Points** = 0, **Millimeters** = 1,
Inches = 2, **Points** = 0, **Millimeters** = 1, **Inches** = 2,
Points = 0, **Millimeters** = 1, **Inches** = 2 }
- enum **Permission** {
PRINT = 1, **COPY** = 2, **CHANGE** = 4, **ADDNOTES** = 8,
FILLFIELDS = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128,
ALL = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL,
PRINT = 1, **COPY** = 2, **CHANGE** = 4,
ADDNOTES = 8, **FILLFIELDS** = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64,
PRINTFULL = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | AS↩
SEMBLE | PRINTFULL, **PRINT** = 1, **COPY** = 2,
CHANGE = 4, **ADDNOTES** = 8, **FILLFIELDS** = 16, **COPYACCESS** = 32,
ASSEMBLE = 64, **PRINTFULL** = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | C↩
OPYACCESS | ASSEMBLE | PRINTFULL, **PRINT** = 1,
COPY = 2, **CHANGE** = 4, **ADDNOTES** = 8, **FILLFIELDS** = 16,
COPYACCESS = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES
| FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL,
PRINT = 1, **COPY** = 2, **CHANGE** = 4, **ADDNOTES** = 8,
FILLFIELDS = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128,
ALL = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL
}
- enum **AddImageOption** {
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6 }
- enum **ImageCompression** {
None = 0, **Flate** = 1, **None** = 0, **Flate** = 1,
None = 0, **Flate** = 1, **None** = 0, **Flate** = 1,
None = 0, **Flate** = 1 }
- enum **DocumentOrigin** {

BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3 }

- enum **DocumentProperty** {
PDFVersion = 0, **Author** = 1, **Title** = 2, **Subject** = 3,
Keywords = 4, **Creator** = 5, **Producer** = 6, **PDFVersion** = 0,
Author = 1, **Title** = 2, **Subject** = 3, **Keywords** = 4,
Creator = 5, **Producer** = 6, **PDFVersion** = 0, **Author** = 1,
Title = 2, **Subject** = 3, **Keywords** = 4, **Creator** = 5,
Producer = 6, **PDFVersion** = 0, **Author** = 1, **Title** = 2,
Subject = 3, **Keywords** = 4, **Creator** = 5, **Producer** = 6,
PDFVersion = 0, **Author** = 1, **Title** = 2, **Subject** = 3,
Keywords = 4, **Creator** = 5, **Producer** = 6 }
- enum **Encryption** {
NONE = -1, **BIT40** = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2,
BIT256_AES = 3, **NONE** = -1, **BIT40** = 0, **BIT128_RC4** = 1,
BIT128_AES = 2, **BIT256_AES** = 3, **NONE** = -1, **BIT40** = 0,
BIT128_RC4 = 1, **BIT128_AES** = 2, **BIT256_AES** = 3, **NONE** = -1,
BIT40 = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2, **BIT256_AES** = 3,
NONE = -1, **BIT40** = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2,
BIT256_AES = 3 }
- enum **ImageIntProperty** {
ImageType = 400, **PixelWidth** = 401, **PixelHeight** = 402, **ImageType** = 400,
PixelWidth = 401, **PixelHeight** = 402, **ImageType** = 400, **PixelWidth** = 401,
PixelHeight = 402, **ImageType** = 400, **PixelWidth** = 401, **PixelHeight** = 402,
ImageType = 400, **PixelWidth** = 401, **PixelHeight** = 402 }
- enum **ImageDblProperty** {
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508 }
- enum **ImageFormat** {
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4 }
- enum **MeasurementUnit** {
Points = 0, **Millimeters** = 1, **Inches** = 2, **Points** = 0,
Millimeters = 1, **Inches** = 2, **Points** = 0, **Millimeters** = 1,
Inches = 2, **Points** = 0, **Millimeters** = 1, **Inches** = 2,
Points = 0, **Millimeters** = 1, **Inches** = 2 }
- enum **Permission** {
PRINT = 1, **COPY** = 2, **CHANGE** = 4, **ADDNOTES** = 8,
FILLFIELDS = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128,
ALL = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL,
PRINT = 1, **COPY** = 2, **CHANGE** = 4,
ADDNOTES = 8, **FILLFIELDS** = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64,
PRINTFULL = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | AS←

SEMBLE | PRINTFULL, **PRINT** = 1, **COPY** = 2,
CHANGE = 4, **ADDNOTES** = 8, **FILLFIELDS** = 16, **COPYACCESS** = 32,
ASSEMBLE = 64, **PRINTFULL** = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | C↩
OPYACCESS | ASSEMBLE | PRINTFULL, **PRINT** = 1,
COPY = 2, **CHANGE** = 4, **ADDNOTES** = 8, **FILLFIELDS** = 16,
COPYACCESS = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL,
PRINT = 1, **COPY** = 2, **CHANGE** = 4, **ADDNOTES** = 8,
FILLFIELDS = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128,
ALL = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL
}

- enum **AddImageOption** {
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6,
ForceGDIPlus = -1, **Default** = 0, **AlphaAsGreyscale** = 1, **ImageAndAlpha8Bit** = 2,
Image8Bit = 3, **Alpha8Bit** = 4, **ImageWithAlpha8Bit** = 5, **ImageAndAlpha** = 6 }
- enum **ImageCompression** {
None = 0, **Flate** = 1, **None** = 0, **Flate** = 1,
None = 0, **Flate** = 1, **None** = 0, **Flate** = 1,
None = 0, **Flate** = 1 }
- enum **DocumentOrigin** {
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3,
BottomLeft = 0, **TopLeft** = 1, **TopRight** = 2, **BottomRight** = 3 }
- enum **DocumentProperty** {
PDFVersion = 0, **Author** = 1, **Title** = 2, **Subject** = 3,
Keywords = 4, **Creator** = 5, **Producer** = 6, **PDFVersion** = 0,
Author = 1, **Title** = 2, **Subject** = 3, **Keywords** = 4,
Creator = 5, **Producer** = 6, **PDFVersion** = 0, **Author** = 1,
Title = 2, **Subject** = 3, **Keywords** = 4, **Creator** = 5,
Producer = 6, **PDFVersion** = 0, **Author** = 1, **Title** = 2,
Subject = 3, **Keywords** = 4, **Creator** = 5, **Producer** = 6,
PDFVersion = 0, **Author** = 1, **Title** = 2, **Subject** = 3,
Keywords = 4, **Creator** = 5, **Producer** = 6 }
- enum **Encryption** {
NONE = -1, **BIT40** = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2,
BIT256_AES = 3, **NONE** = -1, **BIT40** = 0, **BIT128_RC4** = 1,
BIT128_AES = 2, **BIT256_AES** = 3, **NONE** = -1, **BIT40** = 0,
BIT128_RC4 = 1, **BIT128_AES** = 2, **BIT256_AES** = 3, **NONE** = -1,
BIT40 = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2, **BIT256_AES** = 3,
NONE = -1, **BIT40** = 0, **BIT128_RC4** = 1, **BIT128_AES** = 2,
BIT256_AES = 3 }
- enum **ImageIntProperty** {
ImageType = 400, **PixelWidth** = 401, **PixelHeight** = 402, **ImageType** = 400,
PixelWidth = 401, **PixelHeight** = 402, **ImageType** = 400, **PixelWidth** = 401,
PixelHeight = 402, **ImageType** = 400, **PixelWidth** = 401, **PixelHeight** = 402,
ImageType = 400, **PixelWidth** = 401, **PixelHeight** = 402 }
- enum **ImageDblProperty** {

TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508,
TopLeftX = 501, **TopLeftY** = 502, **TopRightX** = 503, **TopRightY** = 504,
BottomRightX = 505, **BottomRightY** = 506, **BottomLeftX** = 507, **BottomLeftY** = 508 }

- enum **ImageFormat** {
  JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
  JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
  JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
  JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4,
  JPEG = 1, **BMP** = 2, **TIFF** = 3, **PNG** = 4 }
- enum **MeasurementUnit** {
  Points = 0, **Millimeters** = 1, **Inches** = 2, **Points** = 0,
  Millimeters = 1, **Inches** = 2, **Points** = 0, **Millimeters** = 1,
  Inches = 2, **Points** = 0, **Millimeters** = 1, **Inches** = 2,
  Points = 0, **Millimeters** = 1, **Inches** = 2 }
- enum **Permission** {
  PRINT = 1, **COPY** = 2, **CHANGE** = 4, **ADDNOTES** = 8,
  FILLFIELDS = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128,
  ALL = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL,
  PRINT = 1, **COPY** = 2, **CHANGE** = 4,
  ADDNOTES = 8, **FILLFIELDS** = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64,
  PRINTFULL = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL, **PRINT** = 1, **COPY** = 2,
  CHANGE = 4, **ADDNOTES** = 8, **FILLFIELDS** = 16, **COPYACCESS** = 32,
  ASSEMBLE = 64, **PRINTFULL** = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL, **PRINT** = 1,
  COPY = 2, **CHANGE** = 4, **ADDNOTES** = 8, **FILLFIELDS** = 16,
  COPYACCESS = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128, **ALL** = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL,
  PRINT = 1, **COPY** = 2, **CHANGE** = 4, **ADDNOTES** = 8,
  FILLFIELDS = 16, **COPYACCESS** = 32, **ASSEMBLE** = 64, **PRINTFULL** = 128,
  ALL = PRINT | COPY | CHANGE | ADDNOTES | FILLFIELDS | COPYACCESS | ASSEMBLE | PRINTFULL
  }

Public Member Functions

- **PDFLibrary** (string dllFileName)
- bool **LibraryLoaded** ()
- void **ReleaseLibrary** ()
- int **AddArcToPath** (double CenterX, double CenterY, double TotalAngle)
- int **AddBoxToPath** (double Left, double Top, double Width, double Height)
- int **AddCJKFont** (int CJKFontID)
- int **AddCurveToPath** (double CtAX, double CtAY, double CtBX, double CtBY, double EndX, double EndY)
- int **AddEmbeddedFile** (string FileName, string MIMEType)
- int **AddFileAttachment** (string Title, int EmbeddedFileID)
- int **AddFormFieldChoiceSub** (int Index, string SubName, string DisplayName)
- int **AddFormFieldSub** (int Index, string SubName)
- int **AddFormFont** (int FontID)
- int **AddGlobalJavaScript** (string PackageName, string JavaScript)
- int **AddImageFromFile** (string FileName, int Options)

- int **AddImageFromFileOffset** (string FileName, int Offset, int DataLength, int Options)
- int **AddImageFromString** (byte[ ] Source, int Options)
- int **AddLGIDictToPage** (string DictContent)
- int **AddLineToPath** (double EndX, double EndY)
- int **AddLinkToDestination** (double Left, double Top, double Width, double Height, int DestID, int Options)
- int **AddLinkToEmbeddedFile** (double Left, double Top, double Width, double Height, int EmbeddedFileID, string Title, string Contents, int IconType, int Transpareny)
- int **AddLinkToFile** (double Left, double Top, double Width, double Height, string FileName, int Page, double Position, int NewWindow, int Options)
- int **AddLinkToJavaScript** (double Left, double Top, double Width, double Height, string JavaScript, int Options)
- int **AddLinkToLocalFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AddLinkToPage** (double Left, double Top, double Width, double Height, int Page, double Position, int Options)
- int **AddLinkToWeb** (double Left, double Top, double Width, double Height, string Link, int Options)
- int **AddNoteAnnotation** (double Left, double Top, int AnnotType, double PopupLeft, double PopupTop, double PopupWidth, double PopupHeight, string Title, string Contents, double Red, double Green, double Blue, int Open)
- int **AddOpenTypeFontFromFile** (string FileName, int Options)
- int **AddPageLabels** (int Start, int Style, int Offset, string Prefix)
- int **AddSVGAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AddSWFAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, string Title, int Options)
- int **AddSeparationColor** (string ColorName, double C, double M, double Y, double K, int Options)
- int **AddStampAnnotation** (double Left, double Top, double Width, double Height, int StampType, string Title, string Contents, double Red, double Green, double Blue, int Options)
- int **AddStandardFont** (int StandardFontID)
- int **AddSubsettedFont** (string FontName, int CharsetIndex, string SubsetChars)
- int **AddTextMarkupAnnotation** (int MarkupType, double Left, double Top, double Width, double Height)
- int **AddToFileList** (string ListName, string FileName)
- int **AddTrueTypeFont** (string FontName, int Embed)
- int **AddTrueTypeFontFromFile** (string FileName)
- int **AddTrueTypeSubsettedFont** (string FontName, string SubsetChars, int Options)
- int **AddType1Font** (string FileName)
- int **AddU3DAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AnalyseFile** (string InputFileName, string Password)
- int **AnnotationCount** ()
- int **AnsiStringResultLength** ()
- int **AppendSpace** (double RelativeSpace)
- int **AppendTableColumns** (int TableID, int NewColumnCount)
- int **AppendTableRows** (int TableID, int NewRowCount)
- int **AppendText** (string Text)
- int **ApplyStyle** (string StyleName)
- int **AttachAnnotToForm** (int Index)
- int **BalanceContentStream** ()
- int **BalancePageTree** (int Options)
- int **BeginPageUpdate** ()
- int **CapturePage** (int Page)
- int **CapturePageEx** (int Page, int Options)
- int **CharWidth** (int CharCode)
- int **CheckFileCompliance** (string InputFileName, string Password, int ComplianceTest, int Options)
- int **CheckObjects** ()
- int **CheckPageAnnots** ()

- int **ClearFileList** (string ListName)
- int **ClearImage** (int ImageID)
- int **ClearPageLabels** ()
- int **ClearTextFormatting** ()
- int **CloneOutlineAction** (int OutlineID)
- int **ClonePages** (int StartPage, int EndPage, int RepeatCount)
- int **CloseOutline** (int OutlineID)
- int **ClosePath** ()
- int **CombineContentStreams** ()
- int **CompareOutlines** (int FirstOutlineID, int SecondOutlineID)
- int **CompressContent** ()
- int **CompressFonts** (int Compress)
- int **CompressImages** (int Compress)
- int **CompressPage** ()
- int **ContentStreamCount** ()
- int **ContentStreamSafe** ()
- int **CopyPageRanges** (int DocumentID, string RangeList)
- int **CopyPageRangesEx** (int DocumentID, string RangeList, int Options)
- int **CreateNewObject** ()
- int **CreateTable** (int RowCount, int ColumnCount)
- int **DAAppendFile** (int FileHandle)
- int **DACapturePage** (int FileHandle, int PageRef)
- int **DACapturePageEx** (int FileHandle, int PageRef, int Options)
- int **DACloseFile** (int FileHandle)
- int **DADrawCapturedPage** (int FileHandle, int DACaptureID, int DestPageRef, double PntLeft, double Pnt↩ Bottom, double PntWidth, double PntHeight)
- int **DADrawRotatedCapturedPage** (int FileHandle, int DACaptureID, int DestPageRef, double PntLeft, double PntBottom, double PntWidth, double PntHeight, double Angle)
- int **DAEmbedFileStreams** (int FileHandle, string RootPath)
- string **DAExtractPageText** (int FileHandle, int PageRef, int Options)
- int **DAExtractPageTextBlocks** (int FileHandle, int PageRef, int ExtractOptions)
- int **DAFindPage** (int FileHandle, int Page)
- int **DAGetAnnotationCount** (int FileHandle, int PageRef)
- int **DAGetFormFieldCount** (int FileHandle)
- string **DAGetFormFieldTitle** (int FileHandle, int FieldIndex)
- string **DAGetFormFieldValue** (int FileHandle, int FieldIndex)
- byte[ ] **DAGetImageDataToString** (int FileHandle, int ImageListID, int ImageIndex)
- double **DAGetImageDblProperty** (int FileHandle, int ImageListID, int ImageIndex, int PropertyID)
- int **DAGetImageIntProperty** (int FileHandle, int ImageListID, int ImageIndex, int PropertyID)
- int **DAGetImageListCount** (int FileHandle, int ImageListID)
- string **DAGetInformation** (int FileHandle, string Key)
- int **DAGetObjectCount** (int FileHandle)
- byte[ ] **DAGetObjectToString** (int FileHandle, int ObjectNumber)
- double **DAGetPageBox** (int FileHandle, int PageRef, int BoxIndex, int Dimension)
- byte[ ] **DAGetPageContentToString** (int FileHandle, int PageRef)
- int **DAGetPageCount** (int FileHandle)
- double **DAGetPageHeight** (int FileHandle, int PageRef)
- int **DAGetPageImageList** (int FileHandle, int PageRef)
- double **DAGetPageWidth** (int FileHandle, int PageRef)
- double **DAGetTextBlockBound** (int TextBlockListID, int Index, int BoundIndex)
- double **DAGetTextBlockCharWidth** (int TextBlockListID, int Index, int CharIndex)
- double **DAGetTextBlockColor** (int TextBlockListID, int Index, int ColorComponent)
- int **DAGetTextBlockColorType** (int TextBlockListID, int Index)
- int **DAGetTextBlockCount** (int TextBlockListID)
- string **DAGetTextBlockFontName** (int TextBlockListID, int Index)

- double **DAGetTextBlockFontSize** (int TextBlockListID, int Index)
- string **DAGetTextBlockText** (int TextBlockListID, int Index)
- int **DAHasPageBox** (int FileHandle, int PageRef, int BoxIndex)
- int **DAHidePage** (int FileHandle, int PageRef)
- int **DAMovePage** (int FileHandle, int PageRef, int TargetPageRef, int Options)
- int **DANewPage** (int FileHandle)
- int **DANewPages** (int FileHandle, int PageCount)
- int **DAOpenFile** (string InputFileName, string Password)
- int **DAOpenFileReadOnly** (string InputFileName, string Password)
- int **DAPageRotation** (int FileHandle, int PageRef)
- int **DAReleaseImageList** (int FileHandle, int ImageListID)
- int **DAReleaseTextBlocks** (int TextBlockListID)
- int **DARemoveUsageRights** (int FileHandle)
- int **DARenderPageToDC** (int FileHandle, int PageRef, int DPI, IntPtr DC)
- int **DARenderPageToFile** (int FileHandle, int PageRef, int Options, int DPI, string FileName)
- byte[ ] **DARenderPageToString** (int FileHandle, int PageRef, int Options, int DPI)
- int **DARotatePage** (int FileHandle, int PageRef, int Angle, int Options)
- int **DASaveAsFile** (int FileHandle, string OutputFileName)
- int **DASaveImageDataToFile** (int FileHandle, int ImageListID, int ImageIndex, string ImageFileName)
- int **DASetInformation** (int FileHandle, string Key, string NewValue)
- int **DASetPageBox** (int FileHandle, int PageRef, int BoxIndex, double X1, double Y1, double X2, double Y2)
- int **DASetPageSize** (int FileHandle, int PageRef, double PntWidth, double PntHeight)
- int **DASetTextExtractionArea** (double Left, double Top, double Width, double Height)
- int **DASetTextExtractionOptions** (int OptionID, int NewValue)
- int **DASetTextExtractionScaling** (int Options, double Horizontal, double Vertical)
- int **DASetTextExtractionWordGap** (double NewWordGap)
- int **DAShiftedHeader** (int FileHandle)
- int **Decrypt** ()
- int **DecryptFile** (string InputFileName, string OutputFileName, string Password)
- int **DeleteAnalysis** (int AnalysisID)
- int **DeleteAnnotation** (int Index)
- int **DeleteContentStream** ()
- int **DeleteFormField** (int Index)
- int **DeleteOptionalContentGroup** (int OptionalContentGroupID)
- int **DeletePageLGIDict** (int DictIndex)
- int **DeletePages** (int StartPage, int PageCount)
- int **DocJavaScriptAction** (string ActionType, string JavaScript)
- int **DocumentCount** ()
- int **DrawArc** (double XPos, double YPos, double Radius, double StartAngle, double EndAngle, int Pie, int DrawOptions)
- int **DrawBarcode** (double Left, double Top, double Width, double Height, string Text, int Barcode, int Options)
- int **DrawBox** (double Left, double Top, double Width, double Height, int DrawOptions)
- int **DrawCapturedPage** (int CaptureID, double Left, double Top, double Width, double Height)
- int **DrawCapturedPageMatrix** (int CaptureID, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawCircle** (double XPos, double YPos, double Radius, int DrawOptions)
- int **DrawDataMatrixSymbol** (double Left, double Top, double ModuleSize, string Text, int Encoding, int SymbolSize, int Options)
- int **DrawEllipse** (double XPos, double YPos, double Width, double Height, int DrawOptions)
- int **DrawEllipticArc** (double XPos, double YPos, double Width, double Height, double StartAngle, double EndAngle, int Pie, int DrawOptions)
- int **DrawHTMLText** (double Left, double Top, double Width, string HTMLText)
- string **DrawHTMLTextBox** (double Left, double Top, double Width, double Height, string HTMLText)
- string **DrawHTMLTextBoxMatrix** (double Width, double Height, string HTMLText, double M11, double M12, double M21, double M22, double MDX, double MDY)

- int **DrawImage** (double Left, double Top, double Width, double Height)
- int **DrawImageMatrix** (double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawIntelligentMailBarcode** (double Left, double Top, double BarWidth, double FullBarHeight, double TrackerHeight, double SpaceWidth, string BarcodeData, int Options)
- int **DrawLine** (double StartX, double StartY, double EndX, double EndY)
- int **DrawMultiLineText** (double XPos, double YPos, string Delimiter, string Text)
- int **DrawPDF417Symbol** (double Left, double Top, string Text, int Options)
- int **DrawPDF417SymbolEx** (double Left, double Top, string Text, int Options, int FixedColumns, int Fixed↩ Rows, int ErrorLevel, double ModuleSize, double HeightWidthRatio)
- int **DrawPath** (int PathOptions)
- int **DrawPathEvenOdd** (int PathOptions)
- int **DrawPostScriptXObject** (int PSRef)
- int **DrawRotatedBox** (double Left, double Bottom, double Width, double Height, double Angle, int Draw↩ Options)
- int **DrawRotatedCapturedPage** (int CaptureID, double Left, double Bottom, double Width, double Height, double Angle)
- int **DrawRotatedImage** (double Left, double Bottom, double Width, double Height, double Angle)
- int **DrawRotatedMultiLineText** (double XPos, double YPos, double Angle, string Delimiter, string Text)
- int **DrawRotatedText** (double XPos, double YPos, double Angle, string Text)
- int **DrawRotatedTextBox** (double Left, double Top, double Width, double Height, double Angle, string Text, int Options)
- int **DrawRoundedBox** (double Left, double Top, double Width, double Height, double Radius, int Draw↩ Options)
- int **DrawScaledImage** (double Left, double Top, double Scale)
- int **DrawSpacedText** (double XPos, double YPos, double Spacing, string Text)
- double **DrawTableRows** (int TableID, double Left, double Top, double Height, int FirstRow, int LastRow)
- int **DrawText** (double XPos, double YPos, string Text)
- int **DrawTextArc** (double XPos, double YPos, double Radius, double Angle, string Text, int DrawOptions)
- int **DrawTextBox** (double Left, double Top, double Width, double Height, string Text, int Options)
- int **DrawTextBoxMatrix** (double Width, double Height, string Text, int Options, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawWrappedText** (double XPos, double YPos, double Width, string Text)
- int **EditableContentStream** ()
- int **EmbedFile** (string Title, string FileName, string MIMEType)
- int **EmbeddedFileCount** ()
- int **EncapsulateContentStream** ()
- int **EncodePermissions** (int CanPrint, int CanCopy, int CanChange, int CanAddNotes, int CanFillFields, int CanCopyAccess, int CanAssemble, int CanPrintFull)
- int **Encrypt** (string Owner, string User, int Strength, int Permissions)
- int **EncryptFile** (string InputFileName, string OutputFileName, string Owner, string User, int Strength, int Permissions)
- int **EncryptWithFingerprint** (string Fingerprint)
- int **EncryptionAlgorithm** ()
- int **EncryptionStatus** ()
- int **EncryptionStrength** ()
- int **EndPageUpdate** ()
- int **EndSignProcessToFile** (int SignProcessID, string OutputFile)
- byte[ ] **EndSignProcessToString** (int SignProcessID)
- byte[ ] **ExtractFilePageContentToString** (string InputFileName, string Password, int Page)
- string **ExtractFilePageText** (string InputFileName, string Password, int Page, int Options)
- int **ExtractFilePageTextBlocks** (string InputFileName, string Password, int Page, int Options)
- int **ExtractFilePages** (string InputFileName, string Password, string OutputFileName, string RangeList)
- int **ExtractFilePagesEx** (string InputFileName, string Password, string OutputFileName, string RangeList, int Options)
- int **ExtractPageRanges** (string RangeList)

- int **ExtractPageTextBlocks** (int ExtractOptions)
- int **ExtractPages** (int StartPage, int PageCount)
- int **FileListCount** (string ListName)
- string **FileListItem** (string ListName, int Index)
- int **FindFonts** ()
- int **FindFormFieldByTitle** (string Title)
- int **FindImages** ()
- int **FitImage** (double Left, double Top, double Width, double Height, int HAlign, int VAlign, int Rotate)
- int **FitRotatedTextBox** (double Left, double Top, double Width, double Height, double Angle, string Text, int Options)
- int **FitTextBox** (double Left, double Top, double Width, double Height, string Text, int Options)
- int **FlattenAnnot** (int Index, int Options)
- int **FlattenFormField** (int Index)
- int **FontCount** ()
- string **FontFamily** ()
- int **FontHasKerning** ()
- string **FontName** ()
- string **FontReference** ()
- int **FontSize** ()
- int **FontType** ()
- int **FormFieldCount** ()
- int **FormFieldHasParent** (int Index)
- int **FormFieldJavaScriptAction** (int Index, string ActionType, string JavaScript)
- int **FormFieldWebLinkAction** (int Index, string ActionType, string Link)
- int **GetActionDest** (int ActionID)
- string **GetActionType** (int ActionID)
- string **GetActionURL** (int ActionID)
- string **GetAnalysisInfo** (int AnalysisID, int AnalysisItem)
- int **GetAnnotActionID** (int Index)
- double **GetAnnotDblProperty** (int Index, int Tag)
- int **GetAnnotDest** (int Index)
- int **GetAnnotIntProperty** (int Index, int Tag)
- int **GetAnnotQuadCount** (int Index)
- double **GetAnnotQuadPoints** (int Index, int QuadNumber, int PointNumber)
- int **GetAnnotSoundToFile** (int Index, int Options, string SoundFileName)
- byte[ ] **GetAnnotSoundToString** (int Index, int Options)
- string **GetAnnotStrProperty** (int Index, int Tag)
- double **GetBarcodeWidth** (double NominalWidth, string Text, int Barcode)
- string **GetBaseURL** ()
- int **GetCSDictEPSG** (int CSDictID)
- int **GetCSDictType** (int CSDictID)
- string **GetCSDictWKT** (int CSDictID)
- IntPtr **GetCanvasDC** (int CanvasWidth, int CanvasHeight)
- string **GetCatalogInformation** (string Key)
- byte[ ] **GetContentStreamToString** ()
- string **GetCustomInformation** (string Key)
- string **GetCustomKeys** (int Location)
- string **GetDefaultPrinterName** ()
- string **GetDestName** (int DestID)
- int **GetDestPage** (int DestID)
- int **GetDestType** (int DestID)
- double **GetDestValue** (int DestID, int ValueKey)
- string **GetDocJavaScript** (string ActionType)
- string **GetDocumentFileName** ()
- int **GetDocumentFileSize** ()

- int **GetDocumentID** (int Index)
- string **GetDocumentIdentifier** (int Part, int Options)
- string **GetDocumentMetadata** ()
- int **GetDocumentRepaired** ()
- string **GetDocumentResourceList** ()
- int **GetEmbeddedFileContentToFile** (int Index, string FileName)
- byte[ ] **GetEmbeddedFileContentToString** (int Index)
- int **GetEmbeddedFileID** (int Index)
- int **GetEmbeddedFileIntProperty** (int Index, int Tag)
- string **GetEmbeddedFileStrProperty** (int Index, int Tag)
- string **GetEncryptionFingerprint** ()
- string **GetFileMetadata** (string InputFileName, string Password)
- int **GetFirstChildOutline** (int OutlineID)
- int **GetFirstOutline** ()
- int **GetFontEncoding** ()
- int **GetFontFlags** (int FontFlagItemID)
- int **GetFontID** (int Index)
- int **GetFontObjectNumber** ()
- int **GetFormFieldActionID** (int Index, string TriggerEvent)
- int **GetFormFieldAlignment** (int Index)
- int **GetFormFieldAnnotFlags** (int Index)
- double **GetFormFieldBackgroundColor** (int Index, int ColorComponent)
- int **GetFormFieldBackgroundColorType** (int Index)
- double **GetFormFieldBorderColor** (int Index, int ColorComponent)
- int **GetFormFieldBorderColorType** (int Index)
- double **GetFormFieldBorderProperty** (int Index, int PropKey)
- int **GetFormFieldBorderStyle** (int Index)
- double **GetFormFieldBound** (int Index, int Edge)
- string **GetFormFieldCaption** (int Index)
- int **GetFormFieldCheckStyle** (int Index)
- string **GetFormFieldChildTitle** (int Index)
- int **GetFormFieldChoiceType** (int Index)
- double **GetFormFieldColor** (int Index, int ColorComponent)
- int **GetFormFieldComb** (int Index)
- string **GetFormFieldDefaultValue** (int Index)
- string **GetFormFieldDescription** (int Index)
- int **GetFormFieldFlags** (int Index)
- string **GetFormFieldFontName** (int Index)
- string **GetFormFieldJavaScript** (int Index, string ActionType)
- int **GetFormFieldKidCount** (int Index)
- int **GetFormFieldMaxLen** (int Index)
- int **GetFormFieldNoExport** (int Index)
- int **GetFormFieldPage** (int Index)
- int **GetFormFieldPrintable** (int Index)
- int **GetFormFieldReadOnly** (int Index)
- int **GetFormFieldRequired** (int Index)
- string **GetFormFieldRichTextString** (int Index, string Key)
- int **GetFormFieldRotation** (int Index)
- int **GetFormFieldSubCount** (int Index)
- string **GetFormFieldSubDisplayName** (int Index, int SubIndex)
- string **GetFormFieldSubName** (int Index, int SubIndex)
- int **GetFormFieldSubTempIndex** (int Index, int SubIndex)
- int **GetFormFieldTabOrder** (int Index)
- int **GetFormFieldTabOrderEx** (int Index, int Options)
- int **GetFormFieldTextFlags** (int Index, int ValueKey)

- double **GetFormFieldTextSize** (int Index)
- string **GetFormFieldTitle** (int Index)
- int **GetFormFieldType** (int Index)
- string **GetFormFieldValue** (int Index)
- string **GetFormFieldValueByTitle** (string Title)
- int **GetFormFieldVisible** (int Index)
- string **GetFormFieldWebLink** (int Index, string ActionType)
- int **GetFormFontCount** ()
- string **GetFormFontName** (int FontIndex)
- string **GetGlobalJavaScript** (string PackageName)
- double **GetHTMLTextHeight** (double Width, string HTMLText)
- int **GetHTMLTextLineCount** (double Width, string HTMLText)
- double **GetHTMLTextWidth** (double MaxWidth, string HTMLText)
- int **GetImageID** (int Index)
- int **GetImageListCount** (int ImageListID)
- byte[ ] **GetImageListItemDataToString** (int ImageListID, int ImageIndex, int Options)
- double **GetImageListItemDblProperty** (int ImageListID, int ImageIndex, int PropertyID)
- int **GetImageListItemIntProperty** (int ImageListID, int ImageIndex, int PropertyID)
- int **GetImageMeasureDict** ()
- int **GetImagePageCount** (string FileName)
- int **GetImagePageCountFromString** (byte[ ] Source)
- int **GetImagePtDataDict** ()
- string **GetInformation** (int Key)
- string **GetInstalledFontsByCharset** (int CharsetIndex, int Options)
- string **GetInstalledFontsByCodePage** (int CodePage, int Options)
- int **GetKerning** (string CharPair)
- string **GetLatestPrinterNames** ()
- int **GetMaxObjectNumber** ()
- int **GetMeasureDictBoundsCount** (int MeasureDictID)
- double **GetMeasureDictBoundsItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictCoordinateSystem** (int MeasureDictID)
- int **GetMeasureDictDCSDict** (int MeasureDictID)
- int **GetMeasureDictGCSDict** (int MeasureDictID)
- int **GetMeasureDictGPTSCount** (int MeasureDictID)
- double **GetMeasureDictGPTSItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictLPTSCount** (int MeasureDictID)
- double **GetMeasureDictLPTSItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictPDU** (int MeasureDictID, int UnitIndex)
- int **GetNamedDestination** (string DestName)
- int **GetNextOutline** (int OutlineID)
- int **GetObjectCount** ()
- int **GetObjectDecodeError** (int ObjectNumber)
- byte[ ] **GetObjectToString** (int ObjectNumber)
- int **GetOpenActionDestination** ()
- string **GetOpenActionJavaScript** ()
- int **GetOptionalContentConfigCount** ()
- int **GetOptionalContentConfigLocked** (int OptionalContentConfigID, int OptionalContentGroupID)
- int **GetOptionalContentConfigOrderCount** (int OptionalContentConfigID)
- int **GetOptionalContentConfigOrderItemID** (int OptionalContentConfigID, int ItemIndex)
- string **GetOptionalContentConfigOrderItemLabel** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigOrderItemLevel** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigOrderItemType** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigState** (int OptionalContentConfigID, int OptionalContentGroupID)
- int **GetOptionalContentGroupID** (int Index)
- string **GetOptionalContentGroupName** (int OptionalContentGroupID)

- int **GetOptionalContentGroupPrintable** (int OptionalContentGroupID)
- int **GetOptionalContentGroupVisible** (int OptionalContentGroupID)
- int **GetOrigin** ()
- int **GetOutlineActionID** (int OutlineID)
- double **GetOutlineColor** (int OutlineID, int ColorComponent)
- int **GetOutlineDest** (int OutlineID)
- int **GetOutlineID** (int Index)
- string **GetOutlineJavaScript** (int OutlineID)
- int **GetOutlineObjectNumber** (int OutlineID)
- string **GetOutlineOpenFile** (int OutlineID)
- int **GetOutlinePage** (int OutlineID)
- int **GetOutlineStyle** (int OutlineID)
- string **GetOutlineWebLink** (int OutlineID)
- double **GetPageBox** (int BoxType, int Dimension)
- string **GetPageColorSpaces** (int Options)
- byte[ ] **GetPageContentToString** ()
- int **GetPageImageList** (int Options)
- string **GetPageJavaScript** (string ActionType)
- string **GetPageLGIDictContent** (int DictIndex)
- int **GetPageLGIDictCount** ()
- string **GetPageLabel** (int Page)
- int **GetPageLayout** ()
- byte[ ] **GetPageMetricsToString** (int StartPage, int EndPage, int Options)
- int **GetPageMode** ()
- string **GetPageText** (int ExtractOptions)
- int **GetPageViewPortCount** ()
- int **GetPageViewPortID** (int Index)
- int **GetParentOutline** (int OutlineID)
- int **GetPrevOutline** (int OutlineID)
- string **GetPrinterBins** (string PrinterName)
- byte[ ] **GetPrinterDevModeToString** (string PrinterName)
- string **GetPrinterMediaTypes** (string PrinterName)
- string **GetPrinterNames** ()
- double **GetRenderScale** ()
- int **GetSignProcessByteRange** (int SignProcessID, int ArrayPosition)
- int **GetSignProcessResult** (int SignProcessID)
- int **GetStringListCount** (int StringListID)
- string **GetStringListItem** (int StringListID, int ItemIndex)
- double **GetTableCellDblProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- int **GetTableCellIntProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- string **GetTableCellStrProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- int **GetTableColumnCount** (int TableID)
- int **GetTableLastDrawnRow** (int TableID)
- int **GetTableRowCount** (int TableID)
- string **GetTempPath** ()
- double **GetTextAscent** ()
- double **GetTextBlockBound** (int TextBlockListID, int Index, int BoundIndex)
- double **GetTextBlockCharWidth** (int TextBlockListID, int Index, int CharIndex)
- double **GetTextBlockColor** (int TextBlockListID, int Index, int ColorComponent)
- int **GetTextBlockColorType** (int TextBlockListID, int Index)
- int **GetTextBlockCount** (int TextBlockListID)
- string **GetTextBlockFontName** (int TextBlockListID, int Index)
- double **GetTextBlockFontSize** (int TextBlockListID, int Index)
- string **GetTextBlockText** (int TextBlockListID, int Index)
- double **GetTextBound** (int Edge)

- double **GetTextDescent** ()
- double **GetTextHeight** ()
- double **GetTextSize** ()
- double **GetTextWidth** (string Text)
- string **GetUnicodeCharactersFromEncoding** (int Encoding)
- double **GetViewPortBBox** (int ViewPortID, int Dimension)
- int **GetViewPortMeasureDict** (int ViewPortID)
- string **GetViewPortName** (int ViewPortID)
- int **GetViewPortPtDataDict** (int ViewPortID)
- int **GetViewerPreferences** (int Option)
- string **GetWrappedText** (double Width, string Delimiter, string Text)
- string **GetWrappedTextBreakString** (double Width, string Delimiter, string Text)
- double **GetWrappedTextHeight** (double Width, string Text)
- int **GetWrappedTextLineCount** (double Width, string Text)
- int **GetXFAFormFieldCount** ()
- string **GetXFAFormFieldName** (int Index)
- string **GetXFAFormFieldNames** (string Delimiter)
- string **GetXFAFormFieldValue** (string XFAFieldName)
- byte[ ] **GetXFAToString** (int Options)
- int **GlobalJavaScriptCount** ()
- string **GlobalJavaScriptPackageName** (int Index)
- int **HasFontResources** ()
- int **HasPageBox** (int BoxType)
- int **HidePage** ()
- int **ImageCount** ()
- int **ImageFillColor** ()
- int **ImageHeight** ()
- int **ImageHorizontalResolution** ()
- int **ImageResolutionUnits** ()
- int **ImageType** ()
- int **ImageVerticalResolution** ()
- int **ImageWidth** ()
- int **ImportEMFFromFile** (string FileName, int FontOptions, int GeneralOptions)
- int **InsertPages** (int StartPage, int PageCount)
- int **InsertTableColumns** (int TableID, int Position, int NewColumnCount)
- int **InsertTableRows** (int TableID, int Position, int NewRowCount)
- int **IsAnnotFormField** (int Index)
- int **IsTaggedPDF** ()
- int **LastErrorCode** ()
- string **LastRenderError** ()
- string **LibraryVersion** ()
- string **LicenseInfo** ()
- int **Linearized** ()
- int **LoadFromCanvasDC** (double DPI, int Options)
- int **LoadFromFile** (string FileName, string Password)
- int **LoadFromString** (byte[ ] Source, string Password)
- int **LoadState** ()
- int **MergeDocument** (int DocumentID)
- int **MergeFileList** (string ListName, string OutputFileName)
- int **MergeFileListFast** (string ListName, string OutputFileName)
- int **MergeFiles** (string FirstFileName, string SecondFileName, string OutputFileName)
- int **MergeTableCells** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn)
- int **MoveContentStream** (int FromPosition, int ToPosition)
- int **MoveOutlineAfter** (int OutlineID, int SiblingID)
- int **MoveOutlineBefore** (int OutlineID, int SiblingID)

- int **MovePage** (int NewPosition)
- int **MovePath** (double NewX, double NewY)
- int **MultiplyScale** (double MultScaleBy)
- int **NewChildFormField** (int Index, string Title, int FieldType)
- int **NewContentStream** ()
- string **NewCustomPrinter** (string OriginalPrinterName)
- int **NewDestination** (int DestPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **NewDocument** ()
- int **NewFormField** (string Title, int FieldType)
- int **NewNamedDestination** (string DestName, int DestID)
- int **NewOptionalContentGroup** (string GroupName)
- int **NewOutline** (int Parent, string Title, int DestPage, double DestPosition)
- int **NewPage** ()
- int **NewPageFromCanvasDC** (int DPI, int Options)
- int **NewPages** (int PageCount)
- int **NewPostScriptXObject** (string PS)
- int **NewRGBAxialShader** (string ShaderName, double StartX, double StartY, double StartRed, double Start↩ Green, double StartBlue, double EndX, double EndY, double EndRed, double EndGreen, double EndBlue, int Extend)
- int **NewSignProcessFromFile** (string InputFile, string Password)
- int **NewSignProcessFromString** (byte[ ] Source, string Password)
- int **NewStaticOutline** (int Parent, string Title)
- int **NewTilingPatternFromCapturedPage** (string PatternName, int CaptureID)
- int **NoEmbedFontListAdd** (string FontName)
- int **NoEmbedFontListCount** ()
- string **NoEmbedFontListGet** (int Index)
- int **NoEmbedFontListRemoveAll** ()
- int **NoEmbedFontListRemoveIndex** (int Index)
- int **NoEmbedFontListRemoveName** (string FontName)
- int **NormalizePage** (int NormalizeOptions)
- int **OpenOutline** (int OutlineID)
- int **OptionalContentGroupCount** ()
- int **OutlineCount** ()
- string **OutlineTitle** (int OutlineID)
- int **PageCount** ()
- double **PageHeight** ()
- int **PageJavaScriptAction** (string ActionType, string JavaScript)
- int **PageRotation** ()
- double **PageWidth** ()
- int **PrintDocument** (string PrinterName, int StartPage, int EndPage, int Options)
- int **PrintDocumentToFile** (string PrinterName, int StartPage, int EndPage, int Options, string FileName)
- int **PrintOptions** (int PageScaling, int AutoRotateCenter, string Title)
- int **PrintPages** (string PrinterName, string PageRanges, int Options)
- int **PrintPagesToFile** (string PrinterName, string PageRanges, int Options, string FileName)
- int **ReleaseImage** (int ImageID)
- int **ReleaseImageList** (int ImageListID)
- int **ReleaseSignProcess** (int SignProcessID)
- int **ReleaseStringList** (int StringListID)
- int **ReleaseTextBlocks** (int TextBlockListID)
- int **RemoveAppearanceStream** (int Index)
- int **RemoveCustomInformation** (string Key)
- int **RemoveDocument** (int DocumentID)
- int **RemoveEmbeddedFile** (int Index)
- int **RemoveFormFieldBackgroundColor** (int Index)

- int **RemoveFormFieldBorderColor** (int Index)
- int **RemoveGlobalJavaScript** (string PackageName)
- int **RemoveOpenAction** ()
- int **RemoveOutline** (int OutlineID)
- int **RemoveSharedContentStreams** ()
- int **RemoveStyle** (string StyleName)
- int **RemoveUsageRights** ()
- int **RemoveXFAEntries** (int Options)
- int **RenderDocumentToFile** (int DPI, int StartPage, int EndPage, int Options, string FileName)
- int **RenderPageToDC** (int DPI, int Page, IntPtr DC)
- int **RenderPageToFile** (int DPI, int Page, int Options, string FileName)
- byte[ ] **RenderPageToString** (int DPI, int Page, int Options)
- int **ReplaceFonts** ()
- int **ReplaceImage** (int OriginalImageID, int NewImageID)
- int **ReplaceTag** (string Tag, string NewValue)
- int **RequestPrinterStatus** (int StatusCommand)
- int **RetrieveCustomDataToFile** (string Key, string FileName, int Location)
- byte[ ] **RetrieveCustomDataToString** (string Key, int Location)
- int **ReverseImage** (int Reset)
- int **RotatePage** (int PageRotation)
- int **SaveFontToFile** (string FileName)
- int **SaveImageListItemDataToFile** (int ImageListID, int ImageIndex, int Options, string ImageFileName)
- int **SaveImageToFile** (string FileName)
- byte[ ] **SaveImageToString** ()
- int **SaveState** ()
- int **SaveStyle** (string StyleName)
- int **SaveToFile** (string FileName)
- byte[ ] **SaveToString** ()
- int **SecurityInfo** (int SecurityItem)
- int **SelectContentStream** (int NewIndex)
- int **SelectDocument** (int DocumentID)
- int **SelectFont** (int FontID)
- int **SelectImage** (int ImageID)
- int **SelectPage** (int PageNumber)
- int **SelectRenderer** (int RendererID)
- int **SelectedDocument** ()
- int **SelectedFont** ()
- int **SelectedImage** ()
- int **SelectedPage** ()
- int **SetActionURL** (int ActionID, string NewURL)
- int **SetAnnotBorderColor** (int Index, double Red, double Green, double Blue)
- int **SetAnnotBorderStyle** (int Index, double Width, int Style, double DashOn, double DashOff)
- int **SetAnnotContents** (int Index, string NewContents)
- int **SetAnnotDblProperty** (int Index, int Tag, double NewValue)
- int **SetAnnotIntProperty** (int Index, int Tag, int NewValue)
- int **SetAnnotQuadPoints** (int Index, int QuadNumber, double X1, double Y1, double X2, double Y2, double X3, double Y3, double X4, double Y4)
- int **SetAnnotRect** (int Index, double Left, double Top, double Width, double Height)
- int **SetAnnotStrProperty** (int Index, int Tag, string NewValue)
- int **SetAnsiMode** (int NewAnsiMode)
- int **SetBaseURL** (string NewBaseURL)
- int **SetBlendMode** (int BlendMode)
- int **SetBreakString** (string NewBreakString)
- int **SetCSDictEPSG** (int CSDictID, int NewEPSG)
- int **SetCSDictType** (int CSDictID, int NewDictType)

- int **SetCSDictWKT** (int CSDictID, string NewWKT)
- int **SetCairoFileName** (string FileName)
- int **SetCapturedPageOptional** (int CaptureID, int OptionalContentGroupID)
- int **SetCapturedPageTransparencyGroup** (int CaptureID, int CS, int Isolate, int Knockout)
- int **SetCatalogInformation** (string Key, string NewValue)
- int **SetCharWidth** (int CharCode, int NewWidth)
- int **SetClippingPath** ()
- int **SetClippingPathEvenOdd** ()
- int **SetCompatibility** (int CompatibilityItem, int CompatibilityMode)
- int **SetContentStreamFromString** (byte[ ] Source)
- int **SetContentStreamOptional** (int OptionalContentGroupID)
- int **SetCropBox** (double Left, double Top, double Width, double Height)
- int **SetCustomInformation** (string Key, string NewValue)
- int **SetCustomLineDash** (string DashPattern, double DashPhase)
- int **SetDestProperties** (int DestID, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetDestValue** (int DestID, int ValueKey, double NewValue)
- int **SetDocumentMetadata** (string XMP)
- int **SetEmbeddedFileStrProperty** (int Index, int Tag, string NewValue)
- int **SetFillColor** (double Red, double Green, double Blue)
- int **SetFillColorCMYK** (double C, double M, double Y, double K)
- int **SetFillColorSep** (string ColorName, double Tint)
- int **SetFillShader** (string ShaderName)
- int **SetFillTilingPattern** (string PatternName)
- int **SetFindImagesMode** (int NewFindImagesMode)
- int **SetFontEncoding** (int Encoding)
- int **SetFontFlags** (int Fixed, int Serif, int Symbolic, int Script, int Italic, int AllCap, int SmallCap, int ForceBold)
- int **SetFormFieldAlignment** (int Index, int Alignment)
- int **SetFormFieldAnnotFlags** (int Index, int NewFlags)
- int **SetFormFieldBackgroundColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldBackgroundColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldBackgroundColorGray** (int Index, double Gray)
- int **SetFormFieldBackgroundColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldBorderColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldBorderColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldBorderColorGray** (int Index, double Gray)
- int **SetFormFieldBorderColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldBorderStyle** (int Index, double Width, int Style, double DashOn, double DashOff)
- int **SetFormFieldBounds** (int Index, double Left, double Top, double Width, double Height)
- int **SetFormFieldCalcOrder** (int Index, int Order)
- int **SetFormFieldCaption** (int Index, string NewCaption)
- int **SetFormFieldCheckStyle** (int Index, int CheckStyle, int Position)
- int **SetFormFieldChildTitle** (int Index, string NewTitle)
- int **SetFormFieldChoiceType** (int Index, int ChoiceType)
- int **SetFormFieldColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldComb** (int Index, int Comb)
- int **SetFormFieldDefaultValue** (int Index, string NewDefaultValue)
- int **SetFormFieldDescription** (int Index, string NewDescription)
- int **SetFormFieldFlags** (int Index, int NewFlags)
- int **SetFormFieldFont** (int Index, int FontIndex)
- int **SetFormFieldHighlightMode** (int Index, int NewMode)
- int **SetFormFieldIcon** (int Index, int IconType, int CaptureID)

- int **SetFormFieldIconStyle** (int Index, int Placement, int Scale, int ScaleType, int HorizontalShift, int Vertical↩ Shift)
- int **SetFormFieldMaxLen** (int Index, int NewMaxLen)
- int **SetFormFieldNoExport** (int Index, int NoExport)
- int **SetFormFieldOptional** (int Index, int OptionalContentGroupID)
- int **SetFormFieldPage** (int Index, int NewPage)
- int **SetFormFieldPrintable** (int Index, int Printable)
- int **SetFormFieldReadOnly** (int Index, int ReadOnly)
- int **SetFormFieldRequired** (int Index, int Required)
- int **SetFormFieldResetAction** (int Index, string ActionType)
- int **SetFormFieldRichTextString** (int Index, string Key, string NewValue)
- int **SetFormFieldRotation** (int Index, int Angle)
- int **SetFormFieldSignatureImage** (int Index, int ImageID, int Options)
- int **SetFormFieldStandardFont** (int Index, int StandardFontID)
- int **SetFormFieldSubmitAction** (int Index, string ActionType, string Link)
- int **SetFormFieldSubmitActionEx** (int Index, string ActionType, string Link, int Flags)
- int **SetFormFieldTabOrder** (int Index, int Order)
- int **SetFormFieldTextFlags** (int Index, int Multiline, int Password, int FileSelect, int DoNotSpellCheck, int DoNotScroll)
- int **SetFormFieldTextSize** (int Index, double NewTextSize)
- int **SetFormFieldValue** (int Index, string NewValue)
- int **SetFormFieldValueByTitle** (string Title, string NewValue)
- int **SetFormFieldVisible** (int Index, int Visible)
- int **SetGDIPlusFileName** (string DLLFileName)
- int **SetGDIPlusOptions** (int OptionID, int NewValue)
- int **SetHTMLBoldFont** (string FontSet, int FontID)
- int **SetHTMLBoldItalicFont** (string FontSet, int FontID)
- int **SetHTMLItalicFont** (string FontSet, int FontID)
- int **SetHTMLNormalFont** (string FontSet, int FontID)
- int **SetImageAsMask** (int MaskType)
- int **SetImageMask** (double FromRed, double FromGreen, double FromBlue, double ToRed, double ToGreen, double ToBlue)
- int **SetImageMaskCMYK** (double FromC, double FromM, double FromY, double FromK, double ToC, double ToM, double ToY, double ToK)
- int **SetImageMaskFromImage** (int ImageID)
- int **SetImageOptional** (int OptionalContentGroupID)
- int **SetImageResolution** (int Horizontal, int Vertical, int Units)
- int **SetInformation** (int Key, string NewValue)
- int **SetJPEGQuality** (int Quality)
- int **SetJavaScriptMode** (int JSMode)
- int **SetKerning** (string CharPair, int Adjustment)
- int **SetLineCap** (int LineCap)
- int **SetLineColor** (double Red, double Green, double Blue)
- int **SetLineColorCMYK** (double C, double M, double Y, double K)
- int **SetLineColorSep** (string ColorName, double Tint)
- int **SetLineDash** (double DashOn, double DashOff)
- int **SetLineDashEx** (string DashValues)
- int **SetLineJoin** (int LineJoin)
- int **SetLineShader** (string ShaderName)
- int **SetLineWidth** (double LineWidth)
- int **SetMarkupAnnotStyle** (int Index, double Red, double Green, double Blue, double Transparency)
- int **SetMeasureDictBoundsCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictBoundsItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictCoordinateSystem** (int MeasureDictID, int CoordinateSystemID)
- int **SetMeasureDictGPTSCount** (int MeasureDictID, int NewCount)

- int **SetMeasureDictGPTSItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictLPTSCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictLPTSItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictPDU** (int MeasureDictID, int LinearUnit, int AreaUnit, int AngularUnit)
- int **SetMeasurementUnits** (int MeasurementUnits)
- int **SetNeedAppearances** (int NewValue)
- int **SetObjectFromString** (int ObjectNumber, byte[ ] Source)
- int **SetOpenActionDestination** (int OpenPage, int Zoom)
- int **SetOpenActionDestinationFull** (int OpenPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetOpenActionJavaScript** (string JavaScript)
- int **SetOpenActionMenu** (string MenuItem)
- int **SetOptionalContentConfigLocked** (int OptionalContentConfigID, int OptionalContentGroupID, int New←Locked)
- int **SetOptionalContentConfigState** (int OptionalContentConfigID, int OptionalContentGroupID, int New←State)
- int **SetOptionalContentGroupName** (int OptionalContentGroupID, string NewGroupName)
- int **SetOptionalContentGroupPrintable** (int OptionalContentGroupID, int Printable)
- int **SetOptionalContentGroupVisible** (int OptionalContentGroupID, int Visible)
- int **SetOrigin** (int Origin)
- int **SetOutlineColor** (int OutlineID, double Red, double Green, double Blue)
- int **SetOutlineDestination** (int OutlineID, int DestPage, double DestPosition)
- int **SetOutlineDestinationFull** (int OutlineID, int DestPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetOutlineDestinationZoom** (int OutlineID, int DestPage, double DestPosition, int Zoom)
- int **SetOutlineJavaScript** (int OutlineID, string JavaScript)
- int **SetOutlineNamedDestination** (int OutlineID, string DestName)
- int **SetOutlineOpenFile** (int OutlineID, string FileName)
- int **SetOutlineRemoteDestination** (int OutlineID, string FileName, int OpenPage, int Zoom, int DestType, double PntLeft, double PntTop, double PntRight, double PntBottom, int NewWindow)
- int **SetOutlineStyle** (int OutlineID, int SetItalic, int SetBold)
- int **SetOutlineTitle** (int OutlineID, string NewTitle)
- int **SetOutlineWebLink** (int OutlineID, string Link)
- int **SetOverprint** (int StrokingOverprint, int OtherOverprint, int OverprintMode)
- int **SetPDFAMode** (int NewMode)
- int **SetPNGTransparencyColor** (int RedByte, int GreenByte, int BlueByte)
- int **SetPageActionMenu** (string MenuItem)
- int **SetPageBox** (int BoxType, double Left, double Top, double Width, double Height)
- int **SetPageContentFromString** (byte[ ] Source)
- int **SetPageDimensions** (double NewPageWidth, double NewPageHeight)
- int **SetPageLayout** (int NewPageLayout)
- int **SetPageMode** (int NewPageMode)
- int **SetPageSize** (string PaperName)
- int **SetPageThumbnail** ()
- int **SetPageTransparencyGroup** (int CS, int Isolate, int Knockout)
- int **SetPrecision** (int NewPrecision)
- int **SetPrinterDevModeFromString** (byte[ ] Source)
- int **SetRenderCropType** (int NewCropType)
- int **SetRenderDCErasePage** (int NewErasePage)
- int **SetRenderDCOffset** (int NewOffsetX, int NewOffsetY)
- int **SetRenderOptions** (int OptionID, int NewValue)
- int **SetRenderScale** (double NewScale)
- int **SetScale** (double NewScale)
- int **SetSignProcessField** (int SignProcessID, string SignatureFieldName)
- int **SetSignProcessFieldBounds** (int SignProcessID, double Left, double Top, double Width, double Height)

- int **SetSignProcessFieldImageFromFile** (int SignProcessID, string ImageFileName, int Options)
- int **SetSignProcessFieldPage** (int SignProcessID, int SignaturePage)
- int **SetSignProcessInfo** (int SignProcessID, string Reason, string Location, string ContactInfo)
- int **SetSignProcessPFXFromFile** (int SignProcessID, string PFXFileName, string PFXPassword)
- int **SetSignProcessPassthrough** (int SignProcessID, int SignatureLength)
- int **SetSignProcessSubFilter** (int SignProcessID, int SubFilter)
- int **SetTableBorderColor** (int TableID, int BorderIndex, double Red, double Green, double Blue)
- int **SetTableBorderColorCMYK** (int TableID, int BorderIndex, double C, double M, double Y, double K)
- int **SetTableBorderWidth** (int TableID, int BorderIndex, double NewWidth)
- int **SetTableCellAlignment** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int New←CellAlignment)
- int **SetTableCellBackgroundColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double Red, double Green, double Blue)
- int **SetTableCellBackgroundColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int Last←Column, double C, double M, double Y, double K)
- int **SetTableCellBorderColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double Red, double Green, double Blue)
- int **SetTableCellBorderColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double C, double M, double Y, double K)
- int **SetTableCellBorderWidth** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double NewWidth)
- int **SetTableCellContent** (int TableID, int RowNumber, int ColumnNumber, string HTMLText)
- int **SetTableCellPadding** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int Border←Index, double NewPadding)
- int **SetTableCellTextColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double Red, double Green, double Blue)
- int **SetTableCellTextColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double C, double M, double Y, double K)
- int **SetTableCellTextSize** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double NewTextSize)
- int **SetTableColumnWidth** (int TableID, int FirstColumn, int LastColumn, double NewWidth)
- int **SetTableRowHeight** (int TableID, int FirstRow, int LastRow, double NewHeight)
- int **SetTableThinBorders** (int TableID, int ThinBorders, double Red, double Green, double Blue)
- int **SetTableThinBordersCMYK** (int TableID, int ThinBorders, double C, double M, double Y, double K)
- int **SetTempFile** (string FileName)
- int **SetTempPath** (string NewPath)
- int **SetTextAlign** (int TextAlign)
- int **SetTextCharSpacing** (double CharSpacing)
- int **SetTextColor** (double Red, double Green, double Blue)
- int **SetTextColorCMYK** (double C, double M, double Y, double K)
- int **SetTextColorSep** (string ColorName, double Tint)
- int **SetTextExtractionArea** (double Left, double Top, double Width, double Height)
- int **SetTextExtractionOptions** (int OptionID, int NewValue)
- int **SetTextExtractionScaling** (int Options, double Horizontal, double Vertical)
- int **SetTextExtractionWordGap** (double NewWordGap)
- int **SetTextHighlight** (int Highlight)
- int **SetTextHighlightColor** (double Red, double Green, double Blue)
- int **SetTextHighlightColorCMYK** (double C, double M, double Y, double K)
- int **SetTextHighlightColorSep** (string ColorName, double Tint)
- int **SetTextMode** (int TextMode)
- int **SetTextRise** (double Rise)
- int **SetTextScaling** (double ScalePercentage)
- int **SetTextShader** (string ShaderName)
- int **SetTextSize** (double TextSize)
- int **SetTextSpacing** (double Spacing)

- int **SetTextUnderline** (int Underline)
- int **SetTextUnderlineColor** (double Red, double Green, double Blue)
- int **SetTextUnderlineColorCMYK** (double C, double M, double Y, double K)
- int **SetTextUnderlineColorSep** (string ColorName, double Tint)
- int **SetTextUnderlineCustomDash** (string DashPattern, double DashPhase)
- int **SetTextUnderlineDash** (double DashOn, double DashOff)
- int **SetTextUnderlineDistance** (double UnderlineDistance)
- int **SetTextUnderlineWidth** (double UnderlineWidth)
- int **SetTextWordSpacing** (double WordSpacing)
- int **SetTransparency** (int Transparency)
- int **SetViewerPreferences** (int Option, int NewValue)
- int **SetXFAFormFieldAccess** (string XFAFieldName, int NewAccess)
- int **SetXFAFormFieldBorderColor** (string XFAFieldName, double Red, double Green, double Blue)
- int **SetXFAFormFieldBorderPresence** (string XFAFieldName, int NewPresence)
- int **SetXFAFormFieldBorderWidth** (string XFAFieldName, double BorderWidth)
- int **SetXFAFormFieldValue** (string XFAFieldName, string NewValue)
- int **SetXFAFromString** (byte[ ] Source, int Options)
- int **SetupCustomPrinter** (string CustomPrinterName, int Setting, int NewValue)
- int **SignFile** (string InputFileName, string OpenPassword, string SignatureFieldName, string OutputFileName, string PFXFileName, string PFXPassword, string Reason, string Location, string ContactInfo)
- int **SplitPageText** (int Options)
- int **StartPath** (double StartX, double StartY)
- int **StoreCustomDataFromFile** (string Key, string FileName, int Location, int Options)
- int **StoreCustomDataFromString** (string Key, byte[ ] NewValue, int Location, int Options)
- int **StringResultLength** ()
- int **TestTempPath** ()
- int **TransformFile** (string InputFileName, string Password, string OutputFileName, int TransformType, int Options)
- int **UnlockKey** (string LicenseKey)
- int **Unlocked** ()
- int **UpdateAndFlattenFormField** (int Index)
- int **UpdateAppearanceStream** (int Index)
- int **UseKerning** (int Kern)
- int **UseUnsafeContentStreams** (int SafetyLevel)
- **PDFLibrary** (string dllFileName)
- bool **LibraryLoaded** ()
- void **ReleaseLibrary** ()
- int **AddArcToPath** (double CenterX, double CenterY, double TotalAngle)
- int **AddBoxToPath** (double Left, double Top, double Width, double Height)
- int **AddCJKFont** (int CJKFontID)
- int **AddCurveToPath** (double CtAX, double CtAY, double CtBX, double CtBY, double EndX, double EndY)
- int **AddEmbeddedFile** (string FileName, string MIMEType)
- int **AddFileAttachment** (string Title, int EmbeddedFileID)
- int **AddFormFieldChoiceSub** (int Index, string SubName, string DisplayName)
- int **AddFormFieldSub** (int Index, string SubName)
- int **AddFormFont** (int FontID)
- int **AddGlobalJavaScript** (string PackageName, string JavaScript)
- int **AddImageFromFile** (string FileName, int Options)
- int **AddImageFromFileOffset** (string FileName, int Offset, int DataLength, int Options)
- int **AddImageFromString** (byte[ ] Source, int Options)
- int **AddLGIDictToPage** (string DictContent)
- int **AddLineToPath** (double EndX, double EndY)
- int **AddLinkToDestination** (double Left, double Top, double Width, double Height, int DestID, int Options)
- int **AddLinkToEmbeddedFile** (double Left, double Top, double Width, double Height, int EmbeddedFileID, string Title, string Contents, int IconType, int Transpareny)

- int **AddLinkToFile** (double Left, double Top, double Width, double Height, string FileName, int Page, double Position, int NewWindow, int Options)
- int **AddLinkToJavaScript** (double Left, double Top, double Width, double Height, string JavaScript, int Options)
- int **AddLinkToLocalFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AddLinkToPage** (double Left, double Top, double Width, double Height, int Page, double Position, int Options)
- int **AddLinkToWeb** (double Left, double Top, double Width, double Height, string Link, int Options)
- int **AddNoteAnnotation** (double Left, double Top, int AnnotType, double PopupLeft, double PopupTop, double PopupWidth, double PopupHeight, string Title, string Contents, double Red, double Green, double Blue, int Open)
- int **AddOpenTypeFontFromFile** (string FileName, int Options)
- int **AddPageLabels** (int Start, int Style, int Offset, string Prefix)
- int **AddSVGAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AddSWFAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, string Title, int Options)
- int **AddSeparationColor** (string ColorName, double C, double M, double Y, double K, int Options)
- int **AddStampAnnotation** (double Left, double Top, double Width, double Height, int StampType, string Title, string Contents, double Red, double Green, double Blue, int Options)
- int **AddStandardFont** (int StandardFontID)
- int **AddSubsettedFont** (string FontName, int CharsetIndex, string SubsetChars)
- int **AddTextMarkupAnnotation** (int MarkupType, double Left, double Top, double Width, double Height)
- int **AddToFileList** (string ListName, string FileName)
- int **AddTrueTypeFont** (string FontName, int Embed)
- int **AddTrueTypeFontFromFile** (string FileName)
- int **AddTrueTypeSubsettedFont** (string FontName, string SubsetChars, int Options)
- int **AddType1Font** (string FileName)
- int **AddU3DAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AnalyseFile** (string InputFileName, string Password)
- int **AnnotationCount** ()
- int **AnsiStringResultLength** ()
- int **AppendSpace** (double RelativeSpace)
- int **AppendTableColumns** (int TableID, int NewColumnCount)
- int **AppendTableRows** (int TableID, int NewRowCount)
- int **AppendText** (string Text)
- int **ApplyStyle** (string StyleName)
- int **AttachAnnotToForm** (int Index)
- int **BalanceContentStream** ()
- int **BalancePageTree** (int Options)
- int **BeginPageUpdate** ()
- int **CapturePage** (int Page)
- int **CapturePageEx** (int Page, int Options)
- int **CharWidth** (int CharCode)
- int **CheckFileCompliance** (string InputFileName, string Password, int ComplianceTest, int Options)
- int **CheckObjects** ()
- int **CheckPageAnnots** ()
- int **ClearFileList** (string ListName)
- int **ClearImage** (int ImageID)
- int **ClearPageLabels** ()
- int **ClearTextFormatting** ()
- int **CloneOutlineAction** (int OutlineID)
- int **ClonePages** (int StartPage, int EndPage, int RepeatCount)
- int **CloseOutline** (int OutlineID)

- int **ClosePath** ()
- int **CombineContentStreams** ()
- int **CompareOutlines** (int FirstOutlineID, int SecondOutlineID)
- int **CompressContent** ()
- int **CompressFonts** (int Compress)
- int **CompressImages** (int Compress)
- int **CompressPage** ()
- int **ContentStreamCount** ()
- int **ContentStreamSafe** ()
- int **CopyPageRanges** (int DocumentID, string RangeList)
- int **CopyPageRangesEx** (int DocumentID, string RangeList, int Options)
- int **CreateNewObject** ()
- int **CreateTable** (int RowCount, int ColumnCount)
- int **DAAppendFile** (int FileHandle)
- int **DACapturePage** (int FileHandle, int PageRef)
- int **DACapturePageEx** (int FileHandle, int PageRef, int Options)
- int **DACloseFile** (int FileHandle)
- int **DADrawCapturedPage** (int FileHandle, int DACaptureID, int DestPageRef, double PntLeft, double Pnt↩ Bottom, double PntWidth, double PntHeight)
- int **DADrawRotatedCapturedPage** (int FileHandle, int DACaptureID, int DestPageRef, double PntLeft, double PntBottom, double PntWidth, double PntHeight, double Angle)
- int **DAEmbedFileStreams** (int FileHandle, string RootPath)
- string **DAExtractPageText** (int FileHandle, int PageRef, int Options)
- int **DAExtractPageTextBlocks** (int FileHandle, int PageRef, int ExtractOptions)
- int **DAFindPage** (int FileHandle, int Page)
- int **DAGetAnnotationCount** (int FileHandle, int PageRef)
- int **DAGetFormFieldCount** (int FileHandle)
- string **DAGetFormFieldTitle** (int FileHandle, int FieldIndex)
- string **DAGetFormFieldValue** (int FileHandle, int FieldIndex)
- byte[ ] **DAGetImageDataToString** (int FileHandle, int ImageListID, int ImageIndex)
- double **DAGetImageDblProperty** (int FileHandle, int ImageListID, int ImageIndex, int PropertyID)
- int **DAGetImageIntProperty** (int FileHandle, int ImageListID, int ImageIndex, int PropertyID)
- int **DAGetImageListCount** (int FileHandle, int ImageListID)
- string **DAGetInformation** (int FileHandle, string Key)
- int **DAGetObjectCount** (int FileHandle)
- byte[ ] **DAGetObjectToString** (int FileHandle, int ObjectNumber)
- double **DAGetPageBox** (int FileHandle, int PageRef, int BoxIndex, int Dimension)
- byte[ ] **DAGetPageContentToString** (int FileHandle, int PageRef)
- int **DAGetPageCount** (int FileHandle)
- double **DAGetPageHeight** (int FileHandle, int PageRef)
- int **DAGetPageImageList** (int FileHandle, int PageRef)
- double **DAGetPageWidth** (int FileHandle, int PageRef)
- double **DAGetTextBlockBound** (int TextBlockListID, int Index, int BoundIndex)
- double **DAGetTextBlockCharWidth** (int TextBlockListID, int Index, int CharIndex)
- double **DAGetTextBlockColor** (int TextBlockListID, int Index, int ColorComponent)
- int **DAGetTextBlockColorType** (int TextBlockListID, int Index)
- int **DAGetTextBlockCount** (int TextBlockListID)
- string **DAGetTextBlockFontName** (int TextBlockListID, int Index)
- double **DAGetTextBlockFontSize** (int TextBlockListID, int Index)
- string **DAGetTextBlockText** (int TextBlockListID, int Index)
- int **DAHasPageBox** (int FileHandle, int PageRef, int BoxIndex)
- int **DAHidePage** (int FileHandle, int PageRef)
- int **DAMovePage** (int FileHandle, int PageRef, int TargetPageRef, int Options)
- int **DANewPage** (int FileHandle)
- int **DANewPages** (int FileHandle, int PageCount)

- int **DAOpenFile** (string InputFileName, string Password)
- int **DAOpenFileReadOnly** (string InputFileName, string Password)
- int **DAPageRotation** (int FileHandle, int PageRef)
- int **DAReleaseImageList** (int FileHandle, int ImageListID)
- int **DAReleaseTextBlocks** (int TextBlockListID)
- int **DARemoveUsageRights** (int FileHandle)
- int **DARenderPageToDC** (int FileHandle, int PageRef, int DPI, IntPtr DC)
- int **DARenderPageToFile** (int FileHandle, int PageRef, int Options, int DPI, string FileName)
- byte[ ] **DARenderPageToString** (int FileHandle, int PageRef, int Options, int DPI)
- int **DARotatePage** (int FileHandle, int PageRef, int Angle, int Options)
- int **DASaveAsFile** (int FileHandle, string OutputFileName)
- int **DASaveImageDataToFile** (int FileHandle, int ImageListID, int ImageIndex, string ImageFileName)
- int **DASetInformation** (int FileHandle, string Key, string NewValue)
- int **DASetPageBox** (int FileHandle, int PageRef, int BoxIndex, double X1, double Y1, double X2, double Y2)
- int **DASetPageSize** (int FileHandle, int PageRef, double PntWidth, double PntHeight)
- int **DASetTextExtractionArea** (double Left, double Top, double Width, double Height)
- int **DASetTextExtractionOptions** (int OptionID, int NewValue)
- int **DASetTextExtractionScaling** (int Options, double Horizontal, double Vertical)
- int **DASetTextExtractionWordGap** (double NewWordGap)
- int **DAShiftedHeader** (int FileHandle)
- int **Decrypt** ()
- int **DecryptFile** (string InputFileName, string OutputFileName, string Password)
- int **DeleteAnalysis** (int AnalysisID)
- int **DeleteAnnotation** (int Index)
- int **DeleteContentStream** ()
- int **DeleteFormField** (int Index)
- int **DeleteOptionalContentGroup** (int OptionalContentGroupID)
- int **DeletePageLGIDict** (int DictIndex)
- int **DeletePages** (int StartPage, int PageCount)
- int **DocJavaScriptAction** (string ActionType, string JavaScript)
- int **DocumentCount** ()
- int **DrawArc** (double XPos, double YPos, double Radius, double StartAngle, double EndAngle, int Pie, int DrawOptions)
- int **DrawBarcode** (double Left, double Top, double Width, double Height, string Text, int Barcode, int Options)
- int **DrawBox** (double Left, double Top, double Width, double Height, int DrawOptions)
- int **DrawCapturedPage** (int CaptureID, double Left, double Top, double Width, double Height)
- int **DrawCapturedPageMatrix** (int CaptureID, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawCircle** (double XPos, double YPos, double Radius, int DrawOptions)
- int **DrawDataMatrixSymbol** (double Left, double Top, double ModuleSize, string Text, int Encoding, int SymbolSize, int Options)
- int **DrawEllipse** (double XPos, double YPos, double Width, double Height, int DrawOptions)
- int **DrawEllipticArc** (double XPos, double YPos, double Width, double Height, double StartAngle, double EndAngle, int Pie, int DrawOptions)
- int **DrawHTMLText** (double Left, double Top, double Width, string HTMLText)
- string **DrawHTMLTextBox** (double Left, double Top, double Width, double Height, string HTMLText)
- string **DrawHTMLTextBoxMatrix** (double Width, double Height, string HTMLText, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawImage** (double Left, double Top, double Width, double Height)
- int **DrawImageMatrix** (double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawIntelligentMailBarcode** (double Left, double Top, double BarWidth, double FullBarHeight, double TrackerHeight, double SpaceWidth, string BarcodeData, int Options)
- int **DrawLine** (double StartX, double StartY, double EndX, double EndY)
- int **DrawMultiLineText** (double XPos, double YPos, string Delimiter, string Text)
- int **DrawPDF417Symbol** (double Left, double Top, string Text, int Options)

- int **DrawPDF417SymbolEx** (double Left, double Top, string Text, int Options, int FixedColumns, int Fixed←↩ Rows, int ErrorLevel, double ModuleSize, double HeightWidthRatio)
- int **DrawPath** (int PathOptions)
- int **DrawPathEvenOdd** (int PathOptions)
- int **DrawPostScriptXObject** (int PSRef)
- int **DrawRotatedBox** (double Left, double Bottom, double Width, double Height, double Angle, int Draw←↩ Options)
- int **DrawRotatedCapturedPage** (int CaptureID, double Left, double Bottom, double Width, double Height, double Angle)
- int **DrawRotatedImage** (double Left, double Bottom, double Width, double Height, double Angle)
- int **DrawRotatedMultiLineText** (double XPos, double YPos, double Angle, string Delimiter, string Text)
- int **DrawRotatedText** (double XPos, double YPos, double Angle, string Text)
- int **DrawRotatedTextBox** (double Left, double Top, double Width, double Height, double Angle, string Text, int Options)
- int **DrawRoundedBox** (double Left, double Top, double Width, double Height, double Radius, int Draw←↩ Options)
- int **DrawScaledImage** (double Left, double Top, double Scale)
- int **DrawSpacedText** (double XPos, double YPos, double Spacing, string Text)
- double **DrawTableRows** (int TableID, double Left, double Top, double Height, int FirstRow, int LastRow)
- int **DrawText** (double XPos, double YPos, string Text)
- int **DrawTextArc** (double XPos, double YPos, double Radius, double Angle, string Text, int DrawOptions)
- int **DrawTextBox** (double Left, double Top, double Width, double Height, string Text, int Options)
- int **DrawTextBoxMatrix** (double Width, double Height, string Text, int Options, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawWrappedText** (double XPos, double YPos, double Width, string Text)
- int **EditableContentStream** ()
- int **EmbedFile** (string Title, string FileName, string MIMEType)
- int **EmbeddedFileCount** ()
- int **EncapsulateContentStream** ()
- int **EncodePermissions** (int CanPrint, int CanCopy, int CanChange, int CanAddNotes, int CanFillFields, int CanCopyAccess, int CanAssemble, int CanPrintFull)
- int **Encrypt** (string Owner, string User, int Strength, int Permissions)
- int **EncryptFile** (string InputFileName, string OutputFileName, string Owner, string User, int Strength, int Permissions)
- int **EncryptWithFingerprint** (string Fingerprint)
- int **EncryptionAlgorithm** ()
- int **EncryptionStatus** ()
- int **EncryptionStrength** ()
- int **EndPageUpdate** ()
- int **EndSignProcessToFile** (int SignProcessID, string OutputFile)
- byte[ ] **EndSignProcessToString** (int SignProcessID)
- byte[ ] **ExtractFilePageContentToString** (string InputFileName, string Password, int Page)
- string **ExtractFilePageText** (string InputFileName, string Password, int Page, int Options)
- int **ExtractFilePageTextBlocks** (string InputFileName, string Password, int Page, int Options)
- int **ExtractFilePages** (string InputFileName, string Password, string OutputFileName, string RangeList)
- int **ExtractFilePagesEx** (string InputFileName, string Password, string OutputFileName, string RangeList, int Options)
- int **ExtractPageRanges** (string RangeList)
- int **ExtractPageTextBlocks** (int ExtractOptions)
- int **ExtractPages** (int StartPage, int PageCount)
- int **FileListCount** (string ListName)
- string **FileListItem** (string ListName, int Index)
- int **FindFonts** ()
- int **FindFormFieldByTitle** (string Title)
- int **FindImages** ()

- int **FitImage** (double Left, double Top, double Width, double Height, int HAlign, int VAlign, int Rotate)
- int **FitRotatedTextBox** (double Left, double Top, double Width, double Height, double Angle, string Text, int Options)
- int **FitTextBox** (double Left, double Top, double Width, double Height, string Text, int Options)
- int **FlattenAnnot** (int Index, int Options)
- int **FlattenFormField** (int Index)
- int **FontCount** ()
- string **FontFamily** ()
- int **FontHasKerning** ()
- string **FontName** ()
- string **FontReference** ()
- int **FontSize** ()
- int **FontType** ()
- int **FormFieldCount** ()
- int **FormFieldHasParent** (int Index)
- int **FormFieldJavaScriptAction** (int Index, string ActionType, string JavaScript)
- int **FormFieldWebLinkAction** (int Index, string ActionType, string Link)
- int **GetActionDest** (int ActionID)
- string **GetActionType** (int ActionID)
- string **GetActionURL** (int ActionID)
- string **GetAnalysisInfo** (int AnalysisID, int AnalysisItem)
- int **GetAnnotActionID** (int Index)
- double **GetAnnotDblProperty** (int Index, int Tag)
- int **GetAnnotDest** (int Index)
- int **GetAnnotIntProperty** (int Index, int Tag)
- int **GetAnnotQuadCount** (int Index)
- double **GetAnnotQuadPoints** (int Index, int QuadNumber, int PointNumber)
- int **GetAnnotSoundToFile** (int Index, int Options, string SoundFileName)
- byte[ ] **GetAnnotSoundToString** (int Index, int Options)
- string **GetAnnotStrProperty** (int Index, int Tag)
- double **GetBarcodeWidth** (double NominalWidth, string Text, int Barcode)
- string **GetBaseURL** ()
- int **GetCSDictEPSG** (int CSDictID)
- int **GetCSDictType** (int CSDictID)
- string **GetCSDictWKT** (int CSDictID)
- IntPtr **GetCanvasDC** (int CanvasWidth, int CanvasHeight)
- string **GetCatalogInformation** (string Key)
- byte[ ] **GetContentStreamToString** ()
- string **GetCustomInformation** (string Key)
- string **GetCustomKeys** (int Location)
- string **GetDefaultPrinterName** ()
- string **GetDestName** (int DestID)
- int **GetDestPage** (int DestID)
- int **GetDestType** (int DestID)
- double **GetDestValue** (int DestID, int ValueKey)
- string **GetDocJavaScript** (string ActionType)
- string **GetDocumentFileName** ()
- int **GetDocumentFileSize** ()
- int **GetDocumentID** (int Index)
- string **GetDocumentIdentifier** (int Part, int Options)
- string **GetDocumentMetadata** ()
- int **GetDocumentRepaired** ()
- string **GetDocumentResourceList** ()
- int **GetEmbeddedFileContentToFile** (int Index, string FileName)
- byte[ ] **GetEmbeddedFileContentToString** (int Index)

- int **GetEmbeddedFileID** (int Index)
- int **GetEmbeddedFileIntProperty** (int Index, int Tag)
- string **GetEmbeddedFileStrProperty** (int Index, int Tag)
- string **GetEncryptionFingerprint** ()
- string **GetFileMetadata** (string InputFileName, string Password)
- int **GetFirstChildOutline** (int OutlineID)
- int **GetFirstOutline** ()
- int **GetFontEncoding** ()
- int **GetFontFlags** (int FontFlagItemID)
- int **GetFontID** (int Index)
- int **GetFontObjectNumber** ()
- int **GetFormFieldActionID** (int Index, string TriggerEvent)
- int **GetFormFieldAlignment** (int Index)
- int **GetFormFieldAnnotFlags** (int Index)
- double **GetFormFieldBackgroundColor** (int Index, int ColorComponent)
- int **GetFormFieldBackgroundColorType** (int Index)
- double **GetFormFieldBorderColor** (int Index, int ColorComponent)
- int **GetFormFieldBorderColorType** (int Index)
- double **GetFormFieldBorderProperty** (int Index, int PropKey)
- int **GetFormFieldBorderStyle** (int Index)
- double **GetFormFieldBound** (int Index, int Edge)
- string **GetFormFieldCaption** (int Index)
- int **GetFormFieldCheckStyle** (int Index)
- string **GetFormFieldChildTitle** (int Index)
- int **GetFormFieldChoiceType** (int Index)
- double **GetFormFieldColor** (int Index, int ColorComponent)
- int **GetFormFieldComb** (int Index)
- string **GetFormFieldDefaultValue** (int Index)
- string **GetFormFieldDescription** (int Index)
- int **GetFormFieldFlags** (int Index)
- string **GetFormFieldFontName** (int Index)
- string **GetFormFieldJavaScript** (int Index, string ActionType)
- int **GetFormFieldKidCount** (int Index)
- int **GetFormFieldMaxLen** (int Index)
- int **GetFormFieldNoExport** (int Index)
- int **GetFormFieldPage** (int Index)
- int **GetFormFieldPrintable** (int Index)
- int **GetFormFieldReadOnly** (int Index)
- int **GetFormFieldRequired** (int Index)
- string **GetFormFieldRichTextString** (int Index, string Key)
- int **GetFormFieldRotation** (int Index)
- int **GetFormFieldSubCount** (int Index)
- string **GetFormFieldSubDisplayName** (int Index, int SubIndex)
- string **GetFormFieldSubName** (int Index, int SubIndex)
- int **GetFormFieldSubTempIndex** (int Index, int SubIndex)
- int **GetFormFieldTabOrder** (int Index)
- int **GetFormFieldTabOrderEx** (int Index, int Options)
- int **GetFormFieldTextFlags** (int Index, int ValueKey)
- double **GetFormFieldTextSize** (int Index)
- string **GetFormFieldTitle** (int Index)
- int **GetFormFieldType** (int Index)
- string **GetFormFieldValue** (int Index)
- string **GetFormFieldValueByTitle** (string Title)
- int **GetFormFieldVisible** (int Index)
- string **GetFormFieldWebLink** (int Index, string ActionType)

- int **GetFormFontCount** ()
- string **GetFormFontName** (int FontIndex)
- string **GetGlobalJavaScript** (string PackageName)
- double **GetHTMLTextHeight** (double Width, string HTMLText)
- int **GetHTMLTextLineCount** (double Width, string HTMLText)
- double **GetHTMLTextWidth** (double MaxWidth, string HTMLText)
- int **GetImageID** (int Index)
- int **GetImageListCount** (int ImageListID)
- byte[ ] **GetImageListItemDataToString** (int ImageListID, int ImageIndex, int Options)
- double **GetImageListItemDblProperty** (int ImageListID, int ImageIndex, int PropertyID)
- int **GetImageListItemIntProperty** (int ImageListID, int ImageIndex, int PropertyID)
- int **GetImageMeasureDict** ()
- int **GetImagePageCount** (string FileName)
- int **GetImagePageCountFromString** (byte[ ] Source)
- int **GetImagePtDataDict** ()
- string **GetInformation** (int Key)
- string **GetInstalledFontsByCharset** (int CharsetIndex, int Options)
- string **GetInstalledFontsByCodePage** (int CodePage, int Options)
- int **GetKerning** (string CharPair)
- string **GetLatestPrinterNames** ()
- int **GetMaxObjectNumber** ()
- int **GetMeasureDictBoundsCount** (int MeasureDictID)
- double **GetMeasureDictBoundsItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictCoordinateSystem** (int MeasureDictID)
- int **GetMeasureDictDCSDict** (int MeasureDictID)
- int **GetMeasureDictGCSDict** (int MeasureDictID)
- int **GetMeasureDictGPTSCount** (int MeasureDictID)
- double **GetMeasureDictGPTSItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictLPTSCount** (int MeasureDictID)
- double **GetMeasureDictLPTSItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictPDU** (int MeasureDictID, int UnitIndex)
- int **GetNamedDestination** (string DestName)
- int **GetNextOutline** (int OutlineID)
- int **GetObjectCount** ()
- int **GetObjectDecodeError** (int ObjectNumber)
- byte[ ] **GetObjectToString** (int ObjectNumber)
- int **GetOpenActionDestination** ()
- string **GetOpenActionJavaScript** ()
- int **GetOptionalContentConfigCount** ()
- int **GetOptionalContentConfigLocked** (int OptionalContentConfigID, int OptionalContentGroupID)
- int **GetOptionalContentConfigOrderCount** (int OptionalContentConfigID)
- int **GetOptionalContentConfigOrderItemID** (int OptionalContentConfigID, int ItemIndex)
- string **GetOptionalContentConfigOrderItemLabel** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigOrderItemLevel** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigOrderItemType** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigState** (int OptionalContentConfigID, int OptionalContentGroupID)
- int **GetOptionalContentGroupID** (int Index)
- string **GetOptionalContentGroupName** (int OptionalContentGroupID)
- int **GetOptionalContentGroupPrintable** (int OptionalContentGroupID)
- int **GetOptionalContentGroupVisible** (int OptionalContentGroupID)
- int **GetOrigin** ()
- int **GetOutlineActionID** (int OutlineID)
- double **GetOutlineColor** (int OutlineID, int ColorComponent)
- int **GetOutlineDest** (int OutlineID)
- int **GetOutlineID** (int Index)

- string **GetOutlineJavaScript** (int OutlineID)
- int **GetOutlineObjectNumber** (int OutlineID)
- string **GetOutlineOpenFile** (int OutlineID)
- int **GetOutlinePage** (int OutlineID)
- int **GetOutlineStyle** (int OutlineID)
- string **GetOutlineWebLink** (int OutlineID)
- double **GetPageBox** (int BoxType, int Dimension)
- string **GetPageColorSpaces** (int Options)
- byte[ ] **GetPageContentToString** ()
- int **GetPageImageList** (int Options)
- string **GetPageJavaScript** (string ActionType)
- string **GetPageLGIDictContent** (int DictIndex)
- int **GetPageLGIDictCount** ()
- string **GetPageLabel** (int Page)
- int **GetPageLayout** ()
- byte[ ] **GetPageMetricsToString** (int StartPage, int EndPage, int Options)
- int **GetPageMode** ()
- string **GetPageText** (int ExtractOptions)
- int **GetPageViewPortCount** ()
- int **GetPageViewPortID** (int Index)
- int **GetParentOutline** (int OutlineID)
- int **GetPrevOutline** (int OutlineID)
- string **GetPrinterBins** (string PrinterName)
- byte[ ] **GetPrinterDevModeToString** (string PrinterName)
- string **GetPrinterMediaTypes** (string PrinterName)
- string **GetPrinterNames** ()
- double **GetRenderScale** ()
- int **GetSignProcessByteRange** (int SignProcessID, int ArrayPosition)
- int **GetSignProcessResult** (int SignProcessID)
- int **GetStringListCount** (int StringListID)
- string **GetStringListItem** (int StringListID, int ItemIndex)
- double **GetTableCellDblProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- int **GetTableCellIntProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- string **GetTableCellStrProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- int **GetTableColumnCount** (int TableID)
- int **GetTableLastDrawnRow** (int TableID)
- int **GetTableRowCount** (int TableID)
- string **GetTempPath** ()
- double **GetTextAscent** ()
- double **GetTextBlockBound** (int TextBlockListID, int Index, int BoundIndex)
- double **GetTextBlockCharWidth** (int TextBlockListID, int Index, int CharIndex)
- double **GetTextBlockColor** (int TextBlockListID, int Index, int ColorComponent)
- int **GetTextBlockColorType** (int TextBlockListID, int Index)
- int **GetTextBlockCount** (int TextBlockListID)
- string **GetTextBlockFontName** (int TextBlockListID, int Index)
- double **GetTextBlockFontSize** (int TextBlockListID, int Index)
- string **GetTextBlockText** (int TextBlockListID, int Index)
- double **GetTextBound** (int Edge)
- double **GetTextDescent** ()
- double **GetTextHeight** ()
- double **GetTextSize** ()
- double **GetTextWidth** (string Text)
- string **GetUnicodeCharactersFromEncoding** (int Encoding)
- double **GetViewPortBBox** (int ViewPortID, int Dimension)
- int **GetViewPortMeasureDict** (int ViewPortID)

- string **GetViewPortName** (int ViewPortID)
- int **GetViewPortPtDataDict** (int ViewPortID)
- int **GetViewerPreferences** (int Option)
- string **GetWrappedText** (double Width, string Delimiter, string Text)
- string **GetWrappedTextBreakString** (double Width, string Delimiter, string Text)
- double **GetWrappedTextHeight** (double Width, string Text)
- int **GetWrappedTextLineCount** (double Width, string Text)
- int **GetXFAFormFieldCount** ()
- string **GetXFAFormFieldName** (int Index)
- string **GetXFAFormFieldNames** (string Delimiter)
- string **GetXFAFormFieldValue** (string XFAFieldName)
- byte[ ] **GetXFAToString** (int Options)
- int **GlobalJavaScriptCount** ()
- string **GlobalJavaScriptPackageName** (int Index)
- int **HasFontResources** ()
- int **HasPageBox** (int BoxType)
- int **HidePage** ()
- int **ImageCount** ()
- int **ImageFillColor** ()
- int **ImageHeight** ()
- int **ImageHorizontalResolution** ()
- int **ImageResolutionUnits** ()
- int **ImageType** ()
- int **ImageVerticalResolution** ()
- int **ImageWidth** ()
- int **ImportEMFFromFile** (string FileName, int FontOptions, int GeneralOptions)
- int **InsertPages** (int StartPage, int PageCount)
- int **InsertTableColumns** (int TableID, int Position, int NewColumnCount)
- int **InsertTableRows** (int TableID, int Position, int NewRowCount)
- int **IsAnnotFormField** (int Index)
- int **IsTaggedPDF** ()
- int **LastErrorCode** ()
- string **LastRenderError** ()
- string **LibraryVersion** ()
- string **LicenseInfo** ()
- int **Linearized** ()
- int **LoadFromCanvasDC** (double DPI, int Options)
- int **LoadFromFile** (string FileName, string Password)
- int **LoadFromString** (byte[ ] Source, string Password)
- int **LoadState** ()
- int **MergeDocument** (int DocumentID)
- int **MergeFileList** (string ListName, string OutputFileName)
- int **MergeFileListFast** (string ListName, string OutputFileName)
- int **MergeFiles** (string FirstFileName, string SecondFileName, string OutputFileName)
- int **MergeTableCells** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn)
- int **MoveContentStream** (int FromPosition, int ToPosition)
- int **MoveOutlineAfter** (int OutlineID, int SiblingID)
- int **MoveOutlineBefore** (int OutlineID, int SiblingID)
- int **MovePage** (int NewPosition)
- int **MovePath** (double NewX, double NewY)
- int **MultiplyScale** (double MultScaleBy)
- int **NewChildFormField** (int Index, string Title, int FieldType)
- int **NewContentStream** ()
- string **NewCustomPrinter** (string OriginalPrinterName)

- int **NewDestination** (int DestPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **NewDocument** ()
- int **NewFormField** (string Title, int FieldType)
- int **NewNamedDestination** (string DestName, int DestID)
- int **NewOptionalContentGroup** (string GroupName)
- int **NewOutline** (int Parent, string Title, int DestPage, double DestPosition)
- int **NewPage** ()
- int **NewPageFromCanvasDC** (int DPI, int Options)
- int **NewPages** (int PageCount)
- int **NewPostScriptXObject** (string PS)
- int **NewRGBAxialShader** (string ShaderName, double StartX, double StartY, double StartRed, double Start↩Green, double StartBlue, double EndX, double EndY, double EndRed, double EndGreen, double EndBlue, int Extend)
- int **NewSignProcessFromFile** (string InputFile, string Password)
- int **NewSignProcessFromString** (byte[ ] Source, string Password)
- int **NewStaticOutline** (int Parent, string Title)
- int **NewTilingPatternFromCapturedPage** (string PatternName, int CaptureID)
- int **NoEmbedFontListAdd** (string FontName)
- int **NoEmbedFontListCount** ()
- string **NoEmbedFontListGet** (int Index)
- int **NoEmbedFontListRemoveAll** ()
- int **NoEmbedFontListRemoveIndex** (int Index)
- int **NoEmbedFontListRemoveName** (string FontName)
- int **NormalizePage** (int NormalizeOptions)
- int **OpenOutline** (int OutlineID)
- int **OptionalContentGroupCount** ()
- int **OutlineCount** ()
- string **OutlineTitle** (int OutlineID)
- int **PageCount** ()
- double **PageHeight** ()
- int **PageJavaScriptAction** (string ActionType, string JavaScript)
- int **PageRotation** ()
- double **PageWidth** ()
- int **PrintDocument** (string PrinterName, int StartPage, int EndPage, int Options)
- int **PrintDocumentToFile** (string PrinterName, int StartPage, int EndPage, int Options, string FileName)
- int **PrintOptions** (int PageScaling, int AutoRotateCenter, string Title)
- int **PrintPages** (string PrinterName, string PageRanges, int Options)
- int **PrintPagesToFile** (string PrinterName, string PageRanges, int Options, string FileName)
- int **ReleaseImage** (int ImageID)
- int **ReleaseImageList** (int ImageListID)
- int **ReleaseSignProcess** (int SignProcessID)
- int **ReleaseStringList** (int StringListID)
- int **ReleaseTextBlocks** (int TextBlockListID)
- int **RemoveAppearanceStream** (int Index)
- int **RemoveCustomInformation** (string Key)
- int **RemoveDocument** (int DocumentID)
- int **RemoveEmbeddedFile** (int Index)
- int **RemoveFormFieldBackgroundColor** (int Index)
- int **RemoveFormFieldBorderColor** (int Index)
- int **RemoveGlobalJavaScript** (string PackageName)
- int **RemoveOpenAction** ()
- int **RemoveOutline** (int OutlineID)
- int **RemoveSharedContentStreams** ()
- int **RemoveStyle** (string StyleName)

- int **RemoveUsageRights** ()
- int **RemoveXFAEntries** (int Options)
- int **RenderDocumentToFile** (int DPI, int StartPage, int EndPage, int Options, string FileName)
- int **RenderPageToDC** (int DPI, int Page, IntPtr DC)
- int **RenderPageToFile** (int DPI, int Page, int Options, string FileName)
- byte[ ] **RenderPageToString** (int DPI, int Page, int Options)
- int **ReplaceFonts** ()
- int **ReplaceImage** (int OriginalImageID, int NewImageID)
- int **ReplaceTag** (string Tag, string NewValue)
- int **RequestPrinterStatus** (int StatusCommand)
- int **RetrieveCustomDataToFile** (string Key, string FileName, int Location)
- byte[ ] **RetrieveCustomDataToString** (string Key, int Location)
- int **ReverseImage** (int Reset)
- int **RotatePage** (int PageRotation)
- int **SaveFontToFile** (string FileName)
- int **SaveImageListItemDataToFile** (int ImageListID, int ImageIndex, int Options, string ImageFileName)
- int **SaveImageToFile** (string FileName)
- byte[ ] **SaveImageToString** ()
- int **SaveState** ()
- int **SaveStyle** (string StyleName)
- int **SaveToFile** (string FileName)
- byte[ ] **SaveToString** ()
- int **SecurityInfo** (int SecurityItem)
- int **SelectContentStream** (int NewIndex)
- int **SelectDocument** (int DocumentID)
- int **SelectFont** (int FontID)
- int **SelectImage** (int ImageID)
- int **SelectPage** (int PageNumber)
- int **SelectRenderer** (int RendererID)
- int **SelectedDocument** ()
- int **SelectedFont** ()
- int **SelectedImage** ()
- int **SelectedPage** ()
- int **SetActionURL** (int ActionID, string NewURL)
- int **SetAnnotBorderColor** (int Index, double Red, double Green, double Blue)
- int **SetAnnotBorderStyle** (int Index, double Width, int Style, double DashOn, double DashOff)
- int **SetAnnotContents** (int Index, string NewContents)
- int **SetAnnotDblProperty** (int Index, int Tag, double NewValue)
- int **SetAnnotIntProperty** (int Index, int Tag, int NewValue)
- int **SetAnnotQuadPoints** (int Index, int QuadNumber, double X1, double Y1, double X2, double Y2, double X3, double Y3, double X4, double Y4)
- int **SetAnnotRect** (int Index, double Left, double Top, double Width, double Height)
- int **SetAnnotStrProperty** (int Index, int Tag, string NewValue)
- int **SetAnsiMode** (int NewAnsiMode)
- int **SetBaseURL** (string NewBaseURL)
- int **SetBlendMode** (int BlendMode)
- int **SetBreakString** (string NewBreakString)
- int **SetCSDictEPSG** (int CSDictID, int NewEPSG)
- int **SetCSDictType** (int CSDictID, int NewDictType)
- int **SetCSDictWKT** (int CSDictID, string NewWKT)
- int **SetCairoFileName** (string FileName)
- int **SetCapturedPageOptional** (int CaptureID, int OptionalContentGroupID)
- int **SetCapturedPageTransparencyGroup** (int CaptureID, int CS, int Isolate, int Knockout)
- int **SetCatalogInformation** (string Key, string NewValue)
- int **SetCharWidth** (int CharCode, int NewWidth)

- int **SetClippingPath** ()
- int **SetClippingPathEvenOdd** ()
- int **SetCompatibility** (int CompatibilityItem, int CompatibilityMode)
- int **SetContentStreamFromString** (byte[ ] Source)
- int **SetContentStreamOptional** (int OptionalContentGroupID)
- int **SetCropBox** (double Left, double Top, double Width, double Height)
- int **SetCustomInformation** (string Key, string NewValue)
- int **SetCustomLineDash** (string DashPattern, double DashPhase)
- int **SetDestProperties** (int DestID, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetDestValue** (int DestID, int ValueKey, double NewValue)
- int **SetDocumentMetadata** (string XMP)
- int **SetEmbeddedFileStrProperty** (int Index, int Tag, string NewValue)
- int **SetFillColor** (double Red, double Green, double Blue)
- int **SetFillColorCMYK** (double C, double M, double Y, double K)
- int **SetFillColorSep** (string ColorName, double Tint)
- int **SetFillShader** (string ShaderName)
- int **SetFillTilingPattern** (string PatternName)
- int **SetFindImagesMode** (int NewFindImagesMode)
- int **SetFontEncoding** (int Encoding)
- int **SetFontFlags** (int Fixed, int Serif, int Symbolic, int Script, int Italic, int AllCap, int SmallCap, int ForceBold)
- int **SetFormFieldAlignment** (int Index, int Alignment)
- int **SetFormFieldAnnotFlags** (int Index, int NewFlags)
- int **SetFormFieldBackgroundColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldBackgroundColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldBackgroundColorGray** (int Index, double Gray)
- int **SetFormFieldBackgroundColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldBorderColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldBorderColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldBorderColorGray** (int Index, double Gray)
- int **SetFormFieldBorderColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldBorderStyle** (int Index, double Width, int Style, double DashOn, double DashOff)
- int **SetFormFieldBounds** (int Index, double Left, double Top, double Width, double Height)
- int **SetFormFieldCalcOrder** (int Index, int Order)
- int **SetFormFieldCaption** (int Index, string NewCaption)
- int **SetFormFieldCheckStyle** (int Index, int CheckStyle, int Position)
- int **SetFormFieldChildTitle** (int Index, string NewTitle)
- int **SetFormFieldChoiceType** (int Index, int ChoiceType)
- int **SetFormFieldColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldComb** (int Index, int Comb)
- int **SetFormFieldDefaultValue** (int Index, string NewDefaultValue)
- int **SetFormFieldDescription** (int Index, string NewDescription)
- int **SetFormFieldFlags** (int Index, int NewFlags)
- int **SetFormFieldFont** (int Index, int FontIndex)
- int **SetFormFieldHighlightMode** (int Index, int NewMode)
- int **SetFormFieldIcon** (int Index, int IconType, int CaptureID)
- int **SetFormFieldIconStyle** (int Index, int Placement, int Scale, int ScaleType, int HorizontalShift, int Vertical←Shift)
- int **SetFormFieldMaxLen** (int Index, int NewMaxLen)
- int **SetFormFieldNoExport** (int Index, int NoExport)
- int **SetFormFieldOptional** (int Index, int OptionalContentGroupID)
- int **SetFormFieldPage** (int Index, int NewPage)
- int **SetFormFieldPrintable** (int Index, int Printable)

- int **SetFormFieldReadOnly** (int Index, int ReadOnly)
- int **SetFormFieldRequired** (int Index, int Required)
- int **SetFormFieldResetAction** (int Index, string ActionType)
- int **SetFormFieldRichTextString** (int Index, string Key, string NewValue)
- int **SetFormFieldRotation** (int Index, int Angle)
- int **SetFormFieldSignatureImage** (int Index, int ImageID, int Options)
- int **SetFormFieldStandardFont** (int Index, int StandardFontID)
- int **SetFormFieldSubmitAction** (int Index, string ActionType, string Link)
- int **SetFormFieldSubmitActionEx** (int Index, string ActionType, string Link, int Flags)
- int **SetFormFieldTabOrder** (int Index, int Order)
- int **SetFormFieldTextFlags** (int Index, int Multiline, int Password, int FileSelect, int DoNotSpellCheck, int DoNotScroll)
- int **SetFormFieldTextSize** (int Index, double NewTextSize)
- int **SetFormFieldValue** (int Index, string NewValue)
- int **SetFormFieldValueByTitle** (string Title, string NewValue)
- int **SetFormFieldVisible** (int Index, int Visible)
- int **SetGDIPlusFileName** (string DLLFileName)
- int **SetGDIPlusOptions** (int OptionID, int NewValue)
- int **SetHTMLBoldFont** (string FontSet, int FontID)
- int **SetHTMLBoldItalicFont** (string FontSet, int FontID)
- int **SetHTMLItalicFont** (string FontSet, int FontID)
- int **SetHTMLNormalFont** (string FontSet, int FontID)
- int **SetImageAsMask** (int MaskType)
- int **SetImageMask** (double FromRed, double FromGreen, double FromBlue, double ToRed, double ToGreen, double ToBlue)
- int **SetImageMaskCMYK** (double FromC, double FromM, double FromY, double FromK, double ToC, double ToM, double ToY, double ToK)
- int **SetImageMaskFromImage** (int ImageID)
- int **SetImageOptional** (int OptionalContentGroupID)
- int **SetImageResolution** (int Horizontal, int Vertical, int Units)
- int **SetInformation** (int Key, string NewValue)
- int **SetJPEGQuality** (int Quality)
- int **SetJavaScriptMode** (int JSMode)
- int **SetKerning** (string CharPair, int Adjustment)
- int **SetLineCap** (int LineCap)
- int **SetLineColor** (double Red, double Green, double Blue)
- int **SetLineColorCMYK** (double C, double M, double Y, double K)
- int **SetLineColorSep** (string ColorName, double Tint)
- int **SetLineDash** (double DashOn, double DashOff)
- int **SetLineDashEx** (string DashValues)
- int **SetLineJoin** (int LineJoin)
- int **SetLineShader** (string ShaderName)
- int **SetLineWidth** (double LineWidth)
- int **SetMarkupAnnotStyle** (int Index, double Red, double Green, double Blue, double Transparency)
- int **SetMeasureDictBoundsCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictBoundsItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictCoordinateSystem** (int MeasureDictID, int CoordinateSystemID)
- int **SetMeasureDictGPTSCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictGPTSItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictLPTSCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictLPTSItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictPDU** (int MeasureDictID, int LinearUnit, int AreaUnit, int AngularUnit)
- int **SetMeasurementUnits** (int MeasurementUnits)
- int **SetNeedAppearances** (int NewValue)
- int **SetObjectFromString** (int ObjectNumber, byte[ ] Source)

- int **SetOpenActionDestination** (int OpenPage, int Zoom)
- int **SetOpenActionDestinationFull** (int OpenPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetOpenActionJavaScript** (string JavaScript)
- int **SetOpenActionMenu** (string MenuItem)
- int **SetOptionalContentConfigLocked** (int OptionalContentConfigID, int OptionalContentGroupID, int New↩ Locked)
- int **SetOptionalContentConfigState** (int OptionalContentConfigID, int OptionalContentGroupID, int New↩ State)
- int **SetOptionalContentGroupName** (int OptionalContentGroupID, string NewGroupName)
- int **SetOptionalContentGroupPrintable** (int OptionalContentGroupID, int Printable)
- int **SetOptionalContentGroupVisible** (int OptionalContentGroupID, int Visible)
- int **SetOrigin** (int Origin)
- int **SetOutlineColor** (int OutlineID, double Red, double Green, double Blue)
- int **SetOutlineDestination** (int OutlineID, int DestPage, double DestPosition)
- int **SetOutlineDestinationFull** (int OutlineID, int DestPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetOutlineDestinationZoom** (int OutlineID, int DestPage, double DestPosition, int Zoom)
- int **SetOutlineJavaScript** (int OutlineID, string JavaScript)
- int **SetOutlineNamedDestination** (int OutlineID, string DestName)
- int **SetOutlineOpenFile** (int OutlineID, string FileName)
- int **SetOutlineRemoteDestination** (int OutlineID, string FileName, int OpenPage, int Zoom, int DestType, double PntLeft, double PntTop, double PntRight, double PntBottom, int NewWindow)
- int **SetOutlineStyle** (int OutlineID, int SetItalic, int SetBold)
- int **SetOutlineTitle** (int OutlineID, string NewTitle)
- int **SetOutlineWebLink** (int OutlineID, string Link)
- int **SetOverprint** (int StrokingOverprint, int OtherOverprint, int OverprintMode)
- int **SetPDFAMode** (int NewMode)
- int **SetPNGTransparencyColor** (int RedByte, int GreenByte, int BlueByte)
- int **SetPageActionMenu** (string MenuItem)
- int **SetPageBox** (int BoxType, double Left, double Top, double Width, double Height)
- int **SetPageContentFromString** (byte[ ] Source)
- int **SetPageDimensions** (double NewPageWidth, double NewPageHeight)
- int **SetPageLayout** (int NewPageLayout)
- int **SetPageMode** (int NewPageMode)
- int **SetPageSize** (string PaperName)
- int **SetPageThumbnail** ()
- int **SetPageTransparencyGroup** (int CS, int Isolate, int Knockout)
- int **SetPrecision** (int NewPrecision)
- int **SetPrinterDevModeFromString** (byte[ ] Source)
- int **SetRenderCropType** (int NewCropType)
- int **SetRenderDCErasePage** (int NewErasePage)
- int **SetRenderDCOffset** (int NewOffsetX, int NewOffsetY)
- int **SetRenderOptions** (int OptionID, int NewValue)
- int **SetRenderScale** (double NewScale)
- int **SetScale** (double NewScale)
- int **SetSignProcessField** (int SignProcessID, string SignatureFieldName)
- int **SetSignProcessFieldBounds** (int SignProcessID, double Left, double Top, double Width, double Height)
- int **SetSignProcessFieldImageFromFile** (int SignProcessID, string ImageFileName, int Options)
- int **SetSignProcessFieldPage** (int SignProcessID, int SignaturePage)
- int **SetSignProcessInfo** (int SignProcessID, string Reason, string Location, string ContactInfo)
- int **SetSignProcessPFXFromFile** (int SignProcessID, string PFXFileName, string PFXPassword)
- int **SetSignProcessPassthrough** (int SignProcessID, int SignatureLength)
- int **SetSignProcessSubFilter** (int SignProcessID, int SubFilter)
- int **SetTableBorderColor** (int TableID, int BorderIndex, double Red, double Green, double Blue)

- int **SetTableBorderColorCMYK** (int TableID, int BorderIndex, double C, double M, double Y, double K)
- int **SetTableBorderWidth** (int TableID, int BorderIndex, double NewWidth)
- int **SetTableCellAlignment** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int New↩ CellAlignment)
- int **SetTableCellBackgroundColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double Red, double Green, double Blue)
- int **SetTableCellBackgroundColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int Last↩ Column, double C, double M, double Y, double K)
- int **SetTableCellBorderColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double Red, double Green, double Blue)
- int **SetTableCellBorderColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double C, double M, double Y, double K)
- int **SetTableCellBorderWidth** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double NewWidth)
- int **SetTableCellContent** (int TableID, int RowNumber, int ColumnNumber, string HTMLText)
- int **SetTableCellPadding** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int Border↩ Index, double NewPadding)
- int **SetTableCellTextColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double Red, double Green, double Blue)
- int **SetTableCellTextColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double C, double M, double Y, double K)
- int **SetTableCellTextSize** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double NewTextSize)
- int **SetTableColumnWidth** (int TableID, int FirstColumn, int LastColumn, double NewWidth)
- int **SetTableRowHeight** (int TableID, int FirstRow, int LastRow, double NewHeight)
- int **SetTableThinBorders** (int TableID, int ThinBorders, double Red, double Green, double Blue)
- int **SetTableThinBordersCMYK** (int TableID, int ThinBorders, double C, double M, double Y, double K)
- int **SetTempFile** (string FileName)
- int **SetTempPath** (string NewPath)
- int **SetTextAlign** (int TextAlign)
- int **SetTextCharSpacing** (double CharSpacing)
- int **SetTextColor** (double Red, double Green, double Blue)
- int **SetTextColorCMYK** (double C, double M, double Y, double K)
- int **SetTextColorSep** (string ColorName, double Tint)
- int **SetTextExtractionArea** (double Left, double Top, double Width, double Height)
- int **SetTextExtractionOptions** (int OptionID, int NewValue)
- int **SetTextExtractionScaling** (int Options, double Horizontal, double Vertical)
- int **SetTextExtractionWordGap** (double NewWordGap)
- int **SetTextHighlight** (int Highlight)
- int **SetTextHighlightColor** (double Red, double Green, double Blue)
- int **SetTextHighlightColorCMYK** (double C, double M, double Y, double K)
- int **SetTextHighlightColorSep** (string ColorName, double Tint)
- int **SetTextMode** (int TextMode)
- int **SetTextRise** (double Rise)
- int **SetTextScaling** (double ScalePercentage)
- int **SetTextShader** (string ShaderName)
- int **SetTextSize** (double TextSize)
- int **SetTextSpacing** (double Spacing)
- int **SetTextUnderline** (int Underline)
- int **SetTextUnderlineColor** (double Red, double Green, double Blue)
- int **SetTextUnderlineColorCMYK** (double C, double M, double Y, double K)
- int **SetTextUnderlineColorSep** (string ColorName, double Tint)
- int **SetTextUnderlineCustomDash** (string DashPattern, double DashPhase)
- int **SetTextUnderlineDash** (double DashOn, double DashOff)
- int **SetTextUnderlineDistance** (double UnderlineDistance)

- int **SetTextUnderlineWidth** (double UnderlineWidth)
- int **SetTextWordSpacing** (double WordSpacing)
- int **SetTransparency** (int Transparency)
- int **SetViewerPreferences** (int Option, int NewValue)
- int **SetXFAFormFieldAccess** (string XFAFieldName, int NewAccess)
- int **SetXFAFormFieldBorderColor** (string XFAFieldName, double Red, double Green, double Blue)
- int **SetXFAFormFieldBorderPresence** (string XFAFieldName, int NewPresence)
- int **SetXFAFormFieldBorderWidth** (string XFAFieldName, double BorderWidth)
- int **SetXFAFormFieldValue** (string XFAFieldName, string NewValue)
- int **SetXFAFromString** (byte[ ] Source, int Options)
- int **SetupCustomPrinter** (string CustomPrinterName, int Setting, int NewValue)
- int **SignFile** (string InputFileName, string OpenPassword, string SignatureFieldName, string OutputFileName, string PFXFileName, string PFXPassword, string Reason, string Location, string ContactInfo)
- int **SplitPageText** (int Options)
- int **StartPath** (double StartX, double StartY)
- int **StoreCustomDataFromFile** (string Key, string FileName, int Location, int Options)
- int **StoreCustomDataFromString** (string Key, byte[ ] NewValue, int Location, int Options)
- int **StringResultLength** ()
- int **TestTempPath** ()
- int **TransformFile** (string InputFileName, string Password, string OutputFileName, int TransformType, int Options)
- int **UnlockKey** (string LicenseKey)
- int **Unlocked** ()
- int **UpdateAndFlattenFormField** (int Index)
- int **UpdateAppearanceStream** (int Index)
- int **UseKerning** (int Kern)
- int **UseUnsafeContentStreams** (int SafetyLevel)
- PDFLibrary ()

  Constructor modified by e-Docs USA to initialize the licence key.
- **PDFLibrary** (string dllFileName)
- bool **LibraryLoaded** ()
- void **ReleaseLibrary** ()
- int **AddArcToPath** (double CenterX, double CenterY, double TotalAngle)
- int **AddBoxToPath** (double Left, double Top, double Width, double Height)
- int **AddCJKFont** (int CJKFontID)
- int **AddCurveToPath** (double CtAX, double CtAY, double CtBX, double CtBY, double EndX, double EndY)
- int **AddEmbeddedFile** (string FileName, string MIMEType)
- int **AddFileAttachment** (string Title, int EmbeddedFileID)
- int **AddFormFieldChoiceSub** (int Index, string SubName, string DisplayName)
- int **AddFormFieldSub** (int Index, string SubName)
- int **AddFormFont** (int FontID)
- int **AddGlobalJavaScript** (string PackageName, string JavaScript)
- int **AddImageFromFile** (string FileName, int Options)
- int **AddImageFromFileOffset** (string FileName, int Offset, int DataLength, int Options)
- int **AddImageFromString** (byte[ ] Source, int Options)
- int **AddLGIDictToPage** (string DictContent)
- int **AddLineToPath** (double EndX, double EndY)
- int **AddLinkToDestination** (double Left, double Top, double Width, double Height, int DestID, int Options)
- int **AddLinkToEmbeddedFile** (double Left, double Top, double Width, double Height, int EmbeddedFileID, string Title, string Contents, int IconType, int Transpareny)
- int **AddLinkToFile** (double Left, double Top, double Width, double Height, string FileName, int Page, double Position, int NewWindow, int Options)
- int **AddLinkToJavaScript** (double Left, double Top, double Width, double Height, string JavaScript, int Options)
- int **AddLinkToLocalFile** (double Left, double Top, double Width, double Height, string FileName, int Options)

- int **AddLinkToPage** (double Left, double Top, double Width, double Height, int Page, double Position, int Options)
- int **AddLinkToWeb** (double Left, double Top, double Width, double Height, string Link, int Options)
- int **AddNoteAnnotation** (double Left, double Top, int AnnotType, double PopupLeft, double PopupTop, double PopupWidth, double PopupHeight, string Title, string Contents, double Red, double Green, double Blue, int Open)
- int **AddOpenTypeFontFromFile** (string FileName, int Options)
- int **AddPageLabels** (int Start, int Style, int Offset, string Prefix)
- int **AddSVGAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AddSWFAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, string Title, int Options)
- int **AddSeparationColor** (string ColorName, double C, double M, double Y, double K, int Options)
- int **AddStampAnnotation** (double Left, double Top, double Width, double Height, int StampType, string Title, string Contents, double Red, double Green, double Blue, int Options)
- int **AddStandardFont** (int StandardFontID)
- int **AddSubsettedFont** (string FontName, int CharsetIndex, string SubsetChars)
- int **AddTextMarkupAnnotation** (int MarkupType, double Left, double Top, double Width, double Height)
- int **AddToFileList** (string ListName, string FileName)
- int **AddTrueTypeFont** (string FontName, int Embed)
- int **AddTrueTypeFontFromFile** (string FileName)
- int **AddTrueTypeSubsettedFont** (string FontName, string SubsetChars, int Options)
- int **AddType1Font** (string FileName)
- int **AddU3DAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AnalyseFile** (string InputFileName, string Password)
- int **AnnotationCount** ()
- int **AnsiStringResultLength** ()
- int **AppendSpace** (double RelativeSpace)
- int **AppendTableColumns** (int TableID, int NewColumnCount)
- int **AppendTableRows** (int TableID, int NewRowCount)
- int **AppendText** (string Text)
- int **ApplyStyle** (string StyleName)
- int **AttachAnnotToForm** (int Index)
- int **BalanceContentStream** ()
- int **BalancePageTree** (int Options)
- int **BeginPageUpdate** ()
- int **CapturePage** (int Page)
- int **CapturePageEx** (int Page, int Options)
- int **CharWidth** (int CharCode)
- int **CheckFileCompliance** (string InputFileName, string Password, int ComplianceTest, int Options)
- int **CheckObjects** ()
- int **CheckPageAnnots** ()
- int **ClearFileList** (string ListName)
- int **ClearImage** (int ImageID)
- int **ClearPageLabels** ()
- int **ClearTextFormatting** ()
- int **CloneOutlineAction** (int OutlineID)
- int **ClonePages** (int StartPage, int EndPage, int RepeatCount)
- int **CloseOutline** (int OutlineID)
- int **ClosePath** ()
- int **CombineContentStreams** ()
- int **CompareOutlines** (int FirstOutlineID, int SecondOutlineID)
- int **CompressContent** ()
- int **CompressFonts** (int Compress)

- int **CompressImages** (int Compress)
- int **CompressPage** ()
- int **ContentStreamCount** ()
- int **ContentStreamSafe** ()
- int **CopyPageRanges** (int DocumentID, string RangeList)
- int **CopyPageRangesEx** (int DocumentID, string RangeList, int Options)
- int **CreateNewObject** ()
- int **CreateTable** (int RowCount, int ColumnCount)
- int **DAAppendFile** (int FileHandle)
- int **DACapturePage** (int FileHandle, int PageRef)
- int **DACapturePageEx** (int FileHandle, int PageRef, int Options)
- int **DACloseFile** (int FileHandle)
- int **DADrawCapturedPage** (int FileHandle, int DACaptureID, int DestPageRef, double PntLeft, double Pnt↩ Bottom, double PntWidth, double PntHeight)
- int **DADrawRotatedCapturedPage** (int FileHandle, int DACaptureID, int DestPageRef, double PntLeft, double PntBottom, double PntWidth, double PntHeight, double Angle)
- int **DAEmbedFileStreams** (int FileHandle, string RootPath)
- string **DAExtractPageText** (int FileHandle, int PageRef, int Options)
- int **DAExtractPageTextBlocks** (int FileHandle, int PageRef, int ExtractOptions)
- int **DAFindPage** (int FileHandle, int Page)
- int **DAGetAnnotationCount** (int FileHandle, int PageRef)
- int **DAGetFormFieldCount** (int FileHandle)
- string **DAGetFormFieldTitle** (int FileHandle, int FieldIndex)
- string **DAGetFormFieldValue** (int FileHandle, int FieldIndex)
- byte[ ] **DAGetImageDataToString** (int FileHandle, int ImageListID, int ImageIndex)
- double **DAGetImageDblProperty** (int FileHandle, int ImageListID, int ImageIndex, int PropertyID)
- int **DAGetImageIntProperty** (int FileHandle, int ImageListID, int ImageIndex, int PropertyID)
- int **DAGetImageListCount** (int FileHandle, int ImageListID)
- string **DAGetInformation** (int FileHandle, string Key)
- int **DAGetObjectCount** (int FileHandle)
- byte[ ] **DAGetObjectToString** (int FileHandle, int ObjectNumber)
- double **DAGetPageBox** (int FileHandle, int PageRef, int BoxIndex, int Dimension)
- byte[ ] **DAGetPageContentToString** (int FileHandle, int PageRef)
- int **DAGetPageCount** (int FileHandle)
- double **DAGetPageHeight** (int FileHandle, int PageRef)
- int **DAGetPageImageList** (int FileHandle, int PageRef)
- double **DAGetPageWidth** (int FileHandle, int PageRef)
- double **DAGetTextBlockBound** (int TextBlockListID, int Index, int BoundIndex)
- double **DAGetTextBlockCharWidth** (int TextBlockListID, int Index, int CharIndex)
- double **DAGetTextBlockColor** (int TextBlockListID, int Index, int ColorComponent)
- int **DAGetTextBlockColorType** (int TextBlockListID, int Index)
- int **DAGetTextBlockCount** (int TextBlockListID)
- string **DAGetTextBlockFontName** (int TextBlockListID, int Index)
- double **DAGetTextBlockFontSize** (int TextBlockListID, int Index)
- string **DAGetTextBlockText** (int TextBlockListID, int Index)
- int **DAHasPageBox** (int FileHandle, int PageRef, int BoxIndex)
- int **DAHidePage** (int FileHandle, int PageRef)
- int **DAMovePage** (int FileHandle, int PageRef, int TargetPageRef, int Options)
- int **DANewPage** (int FileHandle)
- int **DANewPages** (int FileHandle, int PageCount)
- int **DAOpenFile** (string InputFileName, string Password)
- int **DAOpenFileReadOnly** (string InputFileName, string Password)
- int **DAPageRotation** (int FileHandle, int PageRef)
- int **DAReleaseImageList** (int FileHandle, int ImageListID)
- int **DAReleaseTextBlocks** (int TextBlockListID)

- int **DARemoveUsageRights** (int FileHandle)
- int **DARenderPageToDC** (int FileHandle, int PageRef, int DPI, IntPtr DC)
- int **DARenderPageToFile** (int FileHandle, int PageRef, int Options, int DPI, string FileName)
- byte[ ] **DARenderPageToString** (int FileHandle, int PageRef, int Options, int DPI)
- int **DARotatePage** (int FileHandle, int PageRef, int Angle, int Options)
- int **DASaveAsFile** (int FileHandle, string OutputFileName)
- int **DASaveImageDataToFile** (int FileHandle, int ImageListID, int ImageIndex, string ImageFileName)
- int **DASetInformation** (int FileHandle, string Key, string NewValue)
- int **DASetPageBox** (int FileHandle, int PageRef, int BoxIndex, double X1, double Y1, double X2, double Y2)
- int **DASetPageSize** (int FileHandle, int PageRef, double PntWidth, double PntHeight)
- int **DASetTextExtractionArea** (double Left, double Top, double Width, double Height)
- int **DASetTextExtractionOptions** (int OptionID, int NewValue)
- int **DASetTextExtractionScaling** (int Options, double Horizontal, double Vertical)
- int **DASetTextExtractionWordGap** (double NewWordGap)
- int **DAShiftedHeader** (int FileHandle)
- int **Decrypt** ()
- int **DecryptFile** (string InputFileName, string OutputFileName, string Password)
- int **DeleteAnalysis** (int AnalysisID)
- int **DeleteAnnotation** (int Index)
- int **DeleteContentStream** ()
- int **DeleteFormField** (int Index)
- int **DeleteOptionalContentGroup** (int OptionalContentGroupID)
- int **DeletePageLGIDict** (int DictIndex)
- int **DeletePages** (int StartPage, int PageCount)
- int **DocJavaScriptAction** (string ActionType, string JavaScript)
- int **DocumentCount** ()
- int **DrawArc** (double XPos, double YPos, double Radius, double StartAngle, double EndAngle, int Pie, int DrawOptions)
- int **DrawBarcode** (double Left, double Top, double Width, double Height, string Text, int Barcode, int Options)
- int **DrawBox** (double Left, double Top, double Width, double Height, int DrawOptions)
- int **DrawCapturedPage** (int CaptureID, double Left, double Top, double Width, double Height)
- int **DrawCapturedPageMatrix** (int CaptureID, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawCircle** (double XPos, double YPos, double Radius, int DrawOptions)
- int **DrawDataMatrixSymbol** (double Left, double Top, double ModuleSize, string Text, int Encoding, int SymbolSize, int Options)
- int **DrawEllipse** (double XPos, double YPos, double Width, double Height, int DrawOptions)
- int **DrawEllipticArc** (double XPos, double YPos, double Width, double Height, double StartAngle, double EndAngle, int Pie, int DrawOptions)
- int **DrawHTMLText** (double Left, double Top, double Width, string HTMLText)
- string **DrawHTMLTextBox** (double Left, double Top, double Width, double Height, string HTMLText)
- string **DrawHTMLTextBoxMatrix** (double Width, double Height, string HTMLText, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawImage** (double Left, double Top, double Width, double Height)
- int **DrawImageMatrix** (double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawIntelligentMailBarcode** (double Left, double Top, double BarWidth, double FullBarHeight, double TrackerHeight, double SpaceWidth, string BarcodeData, int Options)
- int **DrawLine** (double StartX, double StartY, double EndX, double EndY)
- int **DrawMultiLineText** (double XPos, double YPos, string Delimiter, string Text)
- int **DrawPDF417Symbol** (double Left, double Top, string Text, int Options)
- int **DrawPDF417SymbolEx** (double Left, double Top, string Text, int Options, int FixedColumns, int Fixed←Rows, int ErrorLevel, double ModuleSize, double HeightWidthRatio)
- int **DrawPath** (int PathOptions)
- int **DrawPathEvenOdd** (int PathOptions)
- int **DrawPostScriptXObject** (int PSRef)

- int **DrawRotatedBox** (double Left, double Bottom, double Width, double Height, double Angle, int Draw←↩ Options)
- int **DrawRotatedCapturedPage** (int CaptureID, double Left, double Bottom, double Width, double Height, double Angle)
- int **DrawRotatedImage** (double Left, double Bottom, double Width, double Height, double Angle)
- int **DrawRotatedMultiLineText** (double XPos, double YPos, double Angle, string Delimiter, string Text)
- int **DrawRotatedText** (double XPos, double YPos, double Angle, string Text)
- int **DrawRotatedTextBox** (double Left, double Top, double Width, double Height, double Angle, string Text, int Options)
- int **DrawRoundedBox** (double Left, double Top, double Width, double Height, double Radius, int Draw←↩ Options)
- int **DrawScaledImage** (double Left, double Top, double Scale)
- int **DrawSpacedText** (double XPos, double YPos, double Spacing, string Text)
- double **DrawTableRows** (int TableID, double Left, double Top, double Height, int FirstRow, int LastRow)
- int **DrawText** (double XPos, double YPos, string Text)
- int **DrawTextArc** (double XPos, double YPos, double Radius, double Angle, string Text, int DrawOptions)
- int **DrawTextBox** (double Left, double Top, double Width, double Height, string Text, int Options)
- int **DrawTextBoxMatrix** (double Width, double Height, string Text, int Options, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawWrappedText** (double XPos, double YPos, double Width, string Text)
- int **EditableContentStream** ()
- int **EmbedFile** (string Title, string FileName, string MIMEType)
- int **EmbeddedFileCount** ()
- int **EncapsulateContentStream** ()
- int **EncodePermissions** (int CanPrint, int CanCopy, int CanChange, int CanAddNotes, int CanFillFields, int CanCopyAccess, int CanAssemble, int CanPrintFull)
- int **Encrypt** (string Owner, string User, int Strength, int Permissions)
- int **EncryptFile** (string InputFileName, string OutputFileName, string Owner, string User, int Strength, int Permissions)
- int **EncryptWithFingerprint** (string Fingerprint)
- int **EncryptionAlgorithm** ()
- int **EncryptionStatus** ()
- int **EncryptionStrength** ()
- int **EndPageUpdate** ()
- int **EndSignProcessToFile** (int SignProcessID, string OutputFile)
- byte[ ] **EndSignProcessToString** (int SignProcessID)
- byte[ ] **ExtractFilePageContentToString** (string InputFileName, string Password, int Page)
- string **ExtractFilePageText** (string InputFileName, string Password, int Page, int Options)
- int **ExtractFilePageTextBlocks** (string InputFileName, string Password, int Page, int Options)
- int **ExtractFilePages** (string InputFileName, string Password, string OutputFileName, string RangeList)
- int **ExtractFilePagesEx** (string InputFileName, string Password, string OutputFileName, string RangeList, int Options)
- int **ExtractPageRanges** (string RangeList)
- int **ExtractPageTextBlocks** (int ExtractOptions)
- int **ExtractPages** (int StartPage, int PageCount)
- int **FileListCount** (string ListName)
- string **FileListItem** (string ListName, int Index)
- int **FindFonts** ()
- int **FindFormFieldByTitle** (string Title)
- int **FindImages** ()
- int **FitImage** (double Left, double Top, double Width, double Height, int HAlign, int VAlign, int Rotate)
- int **FitRotatedTextBox** (double Left, double Top, double Width, double Height, double Angle, string Text, int Options)
- int **FitTextBox** (double Left, double Top, double Width, double Height, string Text, int Options)
- int **FlattenAnnot** (int Index, int Options)

- int **FlattenFormField** (int Index)
- int **FontCount** ()
- string **FontFamily** ()
- int **FontHasKerning** ()
- string **FontName** ()
- string **FontReference** ()
- int **FontSize** ()
- int **FontType** ()
- int **FormFieldCount** ()
- int **FormFieldHasParent** (int Index)
- int **FormFieldJavaScriptAction** (int Index, string ActionType, string JavaScript)
- int **FormFieldWebLinkAction** (int Index, string ActionType, string Link)
- int **GetActionDest** (int ActionID)
- string **GetActionType** (int ActionID)
- string **GetActionURL** (int ActionID)
- string **GetAnalysisInfo** (int AnalysisID, int AnalysisItem)
- int **GetAnnotActionID** (int Index)
- double **GetAnnotDblProperty** (int Index, int Tag)
- int **GetAnnotDest** (int Index)
- int **GetAnnotIntProperty** (int Index, int Tag)
- int **GetAnnotQuadCount** (int Index)
- double **GetAnnotQuadPoints** (int Index, int QuadNumber, int PointNumber)
- int **GetAnnotSoundToFile** (int Index, int Options, string SoundFileName)
- byte[ ] **GetAnnotSoundToString** (int Index, int Options)
- string **GetAnnotStrProperty** (int Index, int Tag)
- double **GetBarcodeWidth** (double NominalWidth, string Text, int Barcode)
- string **GetBaseURL** ()
- int **GetCSDictEPSG** (int CSDictID)
- int **GetCSDictType** (int CSDictID)
- string **GetCSDictWKT** (int CSDictID)
- IntPtr **GetCanvasDC** (int CanvasWidth, int CanvasHeight)
- string **GetCatalogInformation** (string Key)
- byte[ ] **GetContentStreamToString** ()
- string **GetCustomInformation** (string Key)
- string **GetCustomKeys** (int Location)
- string **GetDefaultPrinterName** ()
- string **GetDestName** (int DestID)
- int **GetDestPage** (int DestID)
- int **GetDestType** (int DestID)
- double **GetDestValue** (int DestID, int ValueKey)
- string **GetDocJavaScript** (string ActionType)
- string **GetDocumentFileName** ()
- int **GetDocumentFileSize** ()
- int **GetDocumentID** (int Index)
- string **GetDocumentIdentifier** (int Part, int Options)
- string **GetDocumentMetadata** ()
- int **GetDocumentRepaired** ()
- string **GetDocumentResourceList** ()
- int **GetEmbeddedFileContentToFile** (int Index, string FileName)
- byte[ ] **GetEmbeddedFileContentToString** (int Index)
- int **GetEmbeddedFileID** (int Index)
- int **GetEmbeddedFileIntProperty** (int Index, int Tag)
- string **GetEmbeddedFileStrProperty** (int Index, int Tag)
- string **GetEncryptionFingerprint** ()
- string **GetFileMetadata** (string InputFileName, string Password)

- int **GetFirstChildOutline** (int OutlineID)
- int **GetFirstOutline** ()
- int **GetFontEncoding** ()
- int **GetFontFlags** (int FontFlagItemID)
- int **GetFontID** (int Index)
- int **GetFontObjectNumber** ()
- int **GetFormFieldActionID** (int Index, string TriggerEvent)
- int **GetFormFieldAlignment** (int Index)
- int **GetFormFieldAnnotFlags** (int Index)
- double **GetFormFieldBackgroundColor** (int Index, int ColorComponent)
- int **GetFormFieldBackgroundColorType** (int Index)
- double **GetFormFieldBorderColor** (int Index, int ColorComponent)
- int **GetFormFieldBorderColorType** (int Index)
- double **GetFormFieldBorderProperty** (int Index, int PropKey)
- int **GetFormFieldBorderStyle** (int Index)
- double **GetFormFieldBound** (int Index, int Edge)
- string **GetFormFieldCaption** (int Index)
- int **GetFormFieldCheckStyle** (int Index)
- string **GetFormFieldChildTitle** (int Index)
- int **GetFormFieldChoiceType** (int Index)
- double **GetFormFieldColor** (int Index, int ColorComponent)
- int **GetFormFieldComb** (int Index)
- string **GetFormFieldDefaultValue** (int Index)
- string **GetFormFieldDescription** (int Index)
- int **GetFormFieldFlags** (int Index)
- string **GetFormFieldFontName** (int Index)
- string **GetFormFieldJavaScript** (int Index, string ActionType)
- int **GetFormFieldKidCount** (int Index)
- int **GetFormFieldMaxLen** (int Index)
- int **GetFormFieldNoExport** (int Index)
- int **GetFormFieldPage** (int Index)
- int **GetFormFieldPrintable** (int Index)
- int **GetFormFieldReadOnly** (int Index)
- int **GetFormFieldRequired** (int Index)
- string **GetFormFieldRichTextString** (int Index, string Key)
- int **GetFormFieldRotation** (int Index)
- int **GetFormFieldSubCount** (int Index)
- string **GetFormFieldSubDisplayName** (int Index, int SubIndex)
- string **GetFormFieldSubName** (int Index, int SubIndex)
- int **GetFormFieldSubTempIndex** (int Index, int SubIndex)
- int **GetFormFieldTabOrder** (int Index)
- int **GetFormFieldTabOrderEx** (int Index, int Options)
- int **GetFormFieldTextFlags** (int Index, int ValueKey)
- double **GetFormFieldTextSize** (int Index)
- string **GetFormFieldTitle** (int Index)
- int **GetFormFieldType** (int Index)
- string **GetFormFieldValue** (int Index)
- string **GetFormFieldValueByTitle** (string Title)
- int **GetFormFieldVisible** (int Index)
- string **GetFormFieldWebLink** (int Index, string ActionType)
- int **GetFormFontCount** ()
- string **GetFormFontName** (int FontIndex)
- string **GetGlobalJavaScript** (string PackageName)
- double **GetHTMLTextHeight** (double Width, string HTMLText)
- int **GetHTMLTextLineCount** (double Width, string HTMLText)

- double **GetHTMLTextWidth** (double MaxWidth, string HTMLText)
- int **GetImageID** (int Index)
- int **GetImageListCount** (int ImageListID)
- byte[ ] **GetImageListItemDataToString** (int ImageListID, int ImageIndex, int Options)
- double **GetImageListItemDblProperty** (int ImageListID, int ImageIndex, int PropertyID)
- int **GetImageListItemIntProperty** (int ImageListID, int ImageIndex, int PropertyID)
- int **GetImageMeasureDict** ()
- int **GetImagePageCount** (string FileName)
- int **GetImagePageCountFromString** (byte[ ] Source)
- int **GetImagePtDataDict** ()
- string **GetInformation** (int Key)
- string **GetInstalledFontsByCharset** (int CharsetIndex, int Options)
- string **GetInstalledFontsByCodePage** (int CodePage, int Options)
- int **GetKerning** (string CharPair)
- string **GetLatestPrinterNames** ()
- int **GetMaxObjectNumber** ()
- int **GetMeasureDictBoundsCount** (int MeasureDictID)
- double **GetMeasureDictBoundsItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictCoordinateSystem** (int MeasureDictID)
- int **GetMeasureDictDCSDict** (int MeasureDictID)
- int **GetMeasureDictGCSDict** (int MeasureDictID)
- int **GetMeasureDictGPTSCount** (int MeasureDictID)
- double **GetMeasureDictGPTSItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictLPTSCount** (int MeasureDictID)
- double **GetMeasureDictLPTSItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictPDU** (int MeasureDictID, int UnitIndex)
- int **GetNamedDestination** (string DestName)
- int **GetNextOutline** (int OutlineID)
- int **GetObjectCount** ()
- int **GetObjectDecodeError** (int ObjectNumber)
- byte[ ] **GetObjectToString** (int ObjectNumber)
- int **GetOpenActionDestination** ()
- string **GetOpenActionJavaScript** ()
- int **GetOptionalContentConfigCount** ()
- int **GetOptionalContentConfigLocked** (int OptionalContentConfigID, int OptionalContentGroupID)
- int **GetOptionalContentConfigOrderCount** (int OptionalContentConfigID)
- int **GetOptionalContentConfigOrderItemID** (int OptionalContentConfigID, int ItemIndex)
- string **GetOptionalContentConfigOrderItemLabel** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigOrderItemLevel** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigOrderItemType** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigState** (int OptionalContentConfigID, int OptionalContentGroupID)
- int **GetOptionalContentGroupID** (int Index)
- string **GetOptionalContentGroupName** (int OptionalContentGroupID)
- int **GetOptionalContentGroupPrintable** (int OptionalContentGroupID)
- int **GetOptionalContentGroupVisible** (int OptionalContentGroupID)
- int **GetOrigin** ()
- int **GetOutlineActionID** (int OutlineID)
- double **GetOutlineColor** (int OutlineID, int ColorComponent)
- int **GetOutlineDest** (int OutlineID)
- int **GetOutlineID** (int Index)
- string **GetOutlineJavaScript** (int OutlineID)
- int **GetOutlineObjectNumber** (int OutlineID)
- string **GetOutlineOpenFile** (int OutlineID)
- int **GetOutlinePage** (int OutlineID)
- int **GetOutlineStyle** (int OutlineID)

- string **GetOutlineWebLink** (int OutlineID)
- double **GetPageBox** (int BoxType, int Dimension)
- string **GetPageColorSpaces** (int Options)
- byte[ ] **GetPageContentToString** ()
- int **GetPageImageList** (int Options)
- string **GetPageJavaScript** (string ActionType)
- string **GetPageLGIDictContent** (int DictIndex)
- int **GetPageLGIDictCount** ()
- string **GetPageLabel** (int Page)
- int **GetPageLayout** ()
- byte[ ] **GetPageMetricsToString** (int StartPage, int EndPage, int Options)
- int **GetPageMode** ()
- string **GetPageText** (int ExtractOptions)
- int **GetPageViewPortCount** ()
- int **GetPageViewPortID** (int Index)
- int **GetParentOutline** (int OutlineID)
- int **GetPrevOutline** (int OutlineID)
- string **GetPrinterBins** (string PrinterName)
- byte[ ] **GetPrinterDevModeToString** (string PrinterName)
- string **GetPrinterMediaTypes** (string PrinterName)
- string **GetPrinterNames** ()
- double **GetRenderScale** ()
- int **GetSignProcessByteRange** (int SignProcessID, int ArrayPosition)
- int **GetSignProcessResult** (int SignProcessID)
- int **GetStringListCount** (int StringListID)
- string **GetStringListItem** (int StringListID, int ItemIndex)
- double **GetTableCellDblProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- int **GetTableCellIntProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- string **GetTableCellStrProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- int **GetTableColumnCount** (int TableID)
- int **GetTableLastDrawnRow** (int TableID)
- int **GetTableRowCount** (int TableID)
- string **GetTempPath** ()
- double **GetTextAscent** ()
- double **GetTextBlockBound** (int TextBlockListID, int Index, int BoundIndex)
- double **GetTextBlockCharWidth** (int TextBlockListID, int Index, int CharIndex)
- double **GetTextBlockColor** (int TextBlockListID, int Index, int ColorComponent)
- int **GetTextBlockColorType** (int TextBlockListID, int Index)
- int **GetTextBlockCount** (int TextBlockListID)
- string **GetTextBlockFontName** (int TextBlockListID, int Index)
- double **GetTextBlockFontSize** (int TextBlockListID, int Index)
- string **GetTextBlockText** (int TextBlockListID, int Index)
- double **GetTextBound** (int Edge)
- double **GetTextDescent** ()
- double **GetTextHeight** ()
- double **GetTextSize** ()
- double **GetTextWidth** (string Text)
- string **GetUnicodeCharactersFromEncoding** (int Encoding)
- double **GetViewPortBBox** (int ViewPortID, int Dimension)
- int **GetViewPortMeasureDict** (int ViewPortID)
- string **GetViewPortName** (int ViewPortID)
- int **GetViewPortPtDataDict** (int ViewPortID)
- int **GetViewerPreferences** (int Option)
- string **GetWrappedText** (double Width, string Delimiter, string Text)
- string **GetWrappedTextBreakString** (double Width, string Delimiter, string Text)

- double **GetWrappedTextHeight** (double Width, string Text)
- int **GetWrappedTextLineCount** (double Width, string Text)
- int **GetXFAFormFieldCount** ()
- string **GetXFAFormFieldName** (int Index)
- string **GetXFAFormFieldNames** (string Delimiter)
- string **GetXFAFormFieldValue** (string XFAFieldName)
- byte[ ] **GetXFAToString** (int Options)
- int **GlobalJavaScriptCount** ()
- string **GlobalJavaScriptPackageName** (int Index)
- int **HasFontResources** ()
- int **HasPageBox** (int BoxType)
- int **HidePage** ()
- int **ImageCount** ()
- int **ImageFillColor** ()
- int **ImageHeight** ()
- int **ImageHorizontalResolution** ()
- int **ImageResolutionUnits** ()
- int **ImageType** ()
- int **ImageVerticalResolution** ()
- int **ImageWidth** ()
- int **ImportEMFFromFile** (string FileName, int FontOptions, int GeneralOptions)
- int **InsertPages** (int StartPage, int PageCount)
- int **InsertTableColumns** (int TableID, int Position, int NewColumnCount)
- int **InsertTableRows** (int TableID, int Position, int NewRowCount)
- int **IsAnnotFormField** (int Index)
- int **IsTaggedPDF** ()
- int **LastErrorCode** ()
- string **LastRenderError** ()
- string **LibraryVersion** ()
- string **LicenseInfo** ()
- int **Linearized** ()
- int **LoadFromCanvasDC** (double DPI, int Options)
- int **LoadFromFile** (string FileName, string Password)
- int **LoadFromString** (byte[ ] Source, string Password)
- int **LoadState** ()
- int **MergeDocument** (int DocumentID)
- int **MergeFileList** (string ListName, string OutputFileName)
- int **MergeFileListFast** (string ListName, string OutputFileName)
- int **MergeFiles** (string FirstFileName, string SecondFileName, string OutputFileName)
- int **MergeTableCells** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn)
- int **MoveContentStream** (int FromPosition, int ToPosition)
- int **MoveOutlineAfter** (int OutlineID, int SiblingID)
- int **MoveOutlineBefore** (int OutlineID, int SiblingID)
- int **MovePage** (int NewPosition)
- int **MovePath** (double NewX, double NewY)
- int **MultiplyScale** (double MultScaleBy)
- int **NewChildFormField** (int Index, string Title, int FieldType)
- int **NewContentStream** ()
- string **NewCustomPrinter** (string OriginalPrinterName)
- int **NewDestination** (int DestPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **NewDocument** ()
- int **NewFormField** (string Title, int FieldType)
- int **NewNamedDestination** (string DestName, int DestID)
- int **NewOptionalContentGroup** (string GroupName)

- int **NewOutline** (int Parent, string Title, int DestPage, double DestPosition)
- int **NewPage** ()
- int **NewPageFromCanvasDC** (int DPI, int Options)
- int **NewPages** (int PageCount)
- int **NewPostScriptXObject** (string PS)
- int **NewRGBAxialShader** (string ShaderName, double StartX, double StartY, double StartRed, double Start↩ Green, double StartBlue, double EndX, double EndY, double EndRed, double EndGreen, double EndBlue, int Extend)
- int **NewSignProcessFromFile** (string InputFile, string Password)
- int **NewSignProcessFromString** (byte[ ] Source, string Password)
- int **NewStaticOutline** (int Parent, string Title)
- int **NewTilingPatternFromCapturedPage** (string PatternName, int CaptureID)
- int **NoEmbedFontListAdd** (string FontName)
- int **NoEmbedFontListCount** ()
- string **NoEmbedFontListGet** (int Index)
- int **NoEmbedFontListRemoveAll** ()
- int **NoEmbedFontListRemoveIndex** (int Index)
- int **NoEmbedFontListRemoveName** (string FontName)
- int **NormalizePage** (int NormalizeOptions)
- int **OpenOutline** (int OutlineID)
- int **OptionalContentGroupCount** ()
- int **OutlineCount** ()
- string **OutlineTitle** (int OutlineID)
- int **PageCount** ()
- double **PageHeight** ()
- int **PageJavaScriptAction** (string ActionType, string JavaScript)
- int **PageRotation** ()
- double **PageWidth** ()
- int **PrintDocument** (string PrinterName, int StartPage, int EndPage, int Options)
- int **PrintDocumentToFile** (string PrinterName, int StartPage, int EndPage, int Options, string FileName)
- int **PrintOptions** (int PageScaling, int AutoRotateCenter, string Title)
- int **PrintPages** (string PrinterName, string PageRanges, int Options)
- int **PrintPagesToFile** (string PrinterName, string PageRanges, int Options, string FileName)
- int **ReleaseImage** (int ImageID)
- int **ReleaseImageList** (int ImageListID)
- int **ReleaseSignProcess** (int SignProcessID)
- int **ReleaseStringList** (int StringListID)
- int **ReleaseTextBlocks** (int TextBlockListID)
- int **RemoveAppearanceStream** (int Index)
- int **RemoveCustomInformation** (string Key)
- int **RemoveDocument** (int DocumentID)
- int **RemoveEmbeddedFile** (int Index)
- int **RemoveFormFieldBackgroundColor** (int Index)
- int **RemoveFormFieldBorderColor** (int Index)
- int **RemoveGlobalJavaScript** (string PackageName)
- int **RemoveOpenAction** ()
- int **RemoveOutline** (int OutlineID)
- int **RemoveSharedContentStreams** ()
- int **RemoveStyle** (string StyleName)
- int **RemoveUsageRights** ()
- int **RemoveXFAEntries** (int Options)
- int **RenderDocumentToFile** (int DPI, int StartPage, int EndPage, int Options, string FileName)
- int **RenderPageToDC** (int DPI, int Page, IntPtr DC)
- int **RenderPageToFile** (int DPI, int Page, int Options, string FileName)
- byte[ ] **RenderPageToString** (int DPI, int Page, int Options)

- int **ReplaceFonts** ()
- int **ReplaceImage** (int OriginalImageID, int NewImageID)
- int **ReplaceTag** (string Tag, string NewValue)
- int **RequestPrinterStatus** (int StatusCommand)
- int **RetrieveCustomDataToFile** (string Key, string FileName, int Location)
- byte[ ] **RetrieveCustomDataToString** (string Key, int Location)
- int **ReverseImage** (int Reset)
- int **RotatePage** (int PageRotation)
- int **SaveFontToFile** (string FileName)
- int **SaveImageListItemDataToFile** (int ImageListID, int ImageIndex, int Options, string ImageFileName)
- int **SaveImageToFile** (string FileName)
- byte[ ] **SaveImageToString** ()
- int **SaveState** ()
- int **SaveStyle** (string StyleName)
- int **SaveToFile** (string FileName)
- byte[ ] **SaveToString** ()
- int **SecurityInfo** (int SecurityItem)
- int **SelectContentStream** (int NewIndex)
- int **SelectDocument** (int DocumentID)
- int **SelectFont** (int FontID)
- int **SelectImage** (int ImageID)
- int **SelectPage** (int PageNumber)
- int **SelectRenderer** (int RendererID)
- int **SelectedDocument** ()
- int **SelectedFont** ()
- int **SelectedImage** ()
- int **SelectedPage** ()
- int **SetActionURL** (int ActionID, string NewURL)
- int **SetAnnotBorderColor** (int Index, double Red, double Green, double Blue)
- int **SetAnnotBorderStyle** (int Index, double Width, int Style, double DashOn, double DashOff)
- int **SetAnnotContents** (int Index, string NewContents)
- int **SetAnnotDblProperty** (int Index, int Tag, double NewValue)
- int **SetAnnotIntProperty** (int Index, int Tag, int NewValue)
- int **SetAnnotQuadPoints** (int Index, int QuadNumber, double X1, double Y1, double X2, double Y2, double X3, double Y3, double X4, double Y4)
- int **SetAnnotRect** (int Index, double Left, double Top, double Width, double Height)
- int **SetAnnotStrProperty** (int Index, int Tag, string NewValue)
- int **SetAnsiMode** (int NewAnsiMode)
- int **SetBaseURL** (string NewBaseURL)
- int **SetBlendMode** (int BlendMode)
- int **SetBreakString** (string NewBreakString)
- int **SetCSDictEPSG** (int CSDictID, int NewEPSG)
- int **SetCSDictType** (int CSDictID, int NewDictType)
- int **SetCSDictWKT** (int CSDictID, string NewWKT)
- int **SetCairoFileName** (string FileName)
- int **SetCapturedPageOptional** (int CaptureID, int OptionalContentGroupID)
- int **SetCapturedPageTransparencyGroup** (int CaptureID, int CS, int Isolate, int Knockout)
- int **SetCatalogInformation** (string Key, string NewValue)
- int **SetCharWidth** (int CharCode, int NewWidth)
- int **SetClippingPath** ()
- int **SetClippingPathEvenOdd** ()
- int **SetCompatibility** (int CompatibilityItem, int CompatibilityMode)
- int **SetContentStreamFromString** (byte[ ] Source)
- int **SetContentStreamOptional** (int OptionalContentGroupID)
- int **SetCropBox** (double Left, double Top, double Width, double Height)

- int **SetCustomInformation** (string Key, string NewValue)
- int **SetCustomLineDash** (string DashPattern, double DashPhase)
- int **SetDestProperties** (int DestID, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetDestValue** (int DestID, int ValueKey, double NewValue)
- int **SetDocumentMetadata** (string XMP)
- int **SetEmbeddedFileStrProperty** (int Index, int Tag, string NewValue)
- int **SetFillColor** (double Red, double Green, double Blue)
- int **SetFillColorCMYK** (double C, double M, double Y, double K)
- int **SetFillColorSep** (string ColorName, double Tint)
- int **SetFillShader** (string ShaderName)
- int **SetFillTilingPattern** (string PatternName)
- int **SetFindImagesMode** (int NewFindImagesMode)
- int **SetFontEncoding** (int Encoding)
- int **SetFontFlags** (int Fixed, int Serif, int Symbolic, int Script, int Italic, int AllCap, int SmallCap, int ForceBold)
- int **SetFormFieldAlignment** (int Index, int Alignment)
- int **SetFormFieldAnnotFlags** (int Index, int NewFlags)
- int **SetFormFieldBackgroundColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldBackgroundColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldBackgroundColorGray** (int Index, double Gray)
- int **SetFormFieldBackgroundColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldBorderColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldBorderColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldBorderColorGray** (int Index, double Gray)
- int **SetFormFieldBorderColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldBorderStyle** (int Index, double Width, int Style, double DashOn, double DashOff)
- int **SetFormFieldBounds** (int Index, double Left, double Top, double Width, double Height)
- int **SetFormFieldCalcOrder** (int Index, int Order)
- int **SetFormFieldCaption** (int Index, string NewCaption)
- int **SetFormFieldCheckStyle** (int Index, int CheckStyle, int Position)
- int **SetFormFieldChildTitle** (int Index, string NewTitle)
- int **SetFormFieldChoiceType** (int Index, int ChoiceType)
- int **SetFormFieldColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldComb** (int Index, int Comb)
- int **SetFormFieldDefaultValue** (int Index, string NewDefaultValue)
- int **SetFormFieldDescription** (int Index, string NewDescription)
- int **SetFormFieldFlags** (int Index, int NewFlags)
- int **SetFormFieldFont** (int Index, int FontIndex)
- int **SetFormFieldHighlightMode** (int Index, int NewMode)
- int **SetFormFieldIcon** (int Index, int IconType, int CaptureID)
- int **SetFormFieldIconStyle** (int Index, int Placement, int Scale, int ScaleType, int HorizontalShift, int Vertical↩ Shift)
- int **SetFormFieldMaxLen** (int Index, int NewMaxLen)
- int **SetFormFieldNoExport** (int Index, int NoExport)
- int **SetFormFieldOptional** (int Index, int OptionalContentGroupID)
- int **SetFormFieldPage** (int Index, int NewPage)
- int **SetFormFieldPrintable** (int Index, int Printable)
- int **SetFormFieldReadOnly** (int Index, int ReadOnly)
- int **SetFormFieldRequired** (int Index, int Required)
- int **SetFormFieldResetAction** (int Index, string ActionType)
- int **SetFormFieldRichTextString** (int Index, string Key, string NewValue)
- int **SetFormFieldRotation** (int Index, int Angle)
- int **SetFormFieldSignatureImage** (int Index, int ImageID, int Options)

- int **SetFormFieldStandardFont** (int Index, int StandardFontID)
- int **SetFormFieldSubmitAction** (int Index, string ActionType, string Link)
- int **SetFormFieldSubmitActionEx** (int Index, string ActionType, string Link, int Flags)
- int **SetFormFieldTabOrder** (int Index, int Order)
- int **SetFormFieldTextFlags** (int Index, int Multiline, int Password, int FileSelect, int DoNotSpellCheck, int DoNotScroll)
- int **SetFormFieldTextSize** (int Index, double NewTextSize)
- int **SetFormFieldValue** (int Index, string NewValue)
- int **SetFormFieldValueByTitle** (string Title, string NewValue)
- int **SetFormFieldVisible** (int Index, int Visible)
- int **SetGDIPlusFileName** (string DLLFileName)
- int **SetGDIPlusOptions** (int OptionID, int NewValue)
- int **SetHTMLBoldFont** (string FontSet, int FontID)
- int **SetHTMLBoldItalicFont** (string FontSet, int FontID)
- int **SetHTMLItalicFont** (string FontSet, int FontID)
- int **SetHTMLNormalFont** (string FontSet, int FontID)
- int **SetImageAsMask** (int MaskType)
- int **SetImageMask** (double FromRed, double FromGreen, double FromBlue, double ToRed, double ToGreen, double ToBlue)
- int **SetImageMaskCMYK** (double FromC, double FromM, double FromY, double FromK, double ToC, double ToM, double ToY, double ToK)
- int **SetImageMaskFromImage** (int ImageID)
- int **SetImageOptional** (int OptionalContentGroupID)
- int **SetImageResolution** (int Horizontal, int Vertical, int Units)
- int **SetInformation** (int Key, string NewValue)
- int **SetJPEGQuality** (int Quality)
- int **SetJavaScriptMode** (int JSMode)
- int **SetKerning** (string CharPair, int Adjustment)
- int **SetLineCap** (int LineCap)
- int **SetLineColor** (double Red, double Green, double Blue)
- int **SetLineColorCMYK** (double C, double M, double Y, double K)
- int **SetLineColorSep** (string ColorName, double Tint)
- int **SetLineDash** (double DashOn, double DashOff)
- int **SetLineDashEx** (string DashValues)
- int **SetLineJoin** (int LineJoin)
- int **SetLineShader** (string ShaderName)
- int **SetLineWidth** (double LineWidth)
- int **SetMarkupAnnotStyle** (int Index, double Red, double Green, double Blue, double Transparency)
- int **SetMeasureDictBoundsCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictBoundsItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictCoordinateSystem** (int MeasureDictID, int CoordinateSystemID)
- int **SetMeasureDictGPTSCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictGPTSItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictLPTSCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictLPTSItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictPDU** (int MeasureDictID, int LinearUnit, int AreaUnit, int AngularUnit)
- int **SetMeasurementUnits** (int MeasurementUnits)
- int **SetNeedAppearances** (int NewValue)
- int **SetObjectFromString** (int ObjectNumber, byte[ ] Source)
- int **SetOpenActionDestination** (int OpenPage, int Zoom)
- int **SetOpenActionDestinationFull** (int OpenPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetOpenActionJavaScript** (string JavaScript)
- int **SetOpenActionMenu** (string MenuItem)

- int **SetOptionalContentConfigLocked** (int OptionalContentConfigID, int OptionalContentGroupID, int New↩ Locked)
- int **SetOptionalContentConfigState** (int OptionalContentConfigID, int OptionalContentGroupID, int New↩ State)
- int **SetOptionalContentGroupName** (int OptionalContentGroupID, string NewGroupName)
- int **SetOptionalContentGroupPrintable** (int OptionalContentGroupID, int Printable)
- int **SetOptionalContentGroupVisible** (int OptionalContentGroupID, int Visible)
- int **SetOrigin** (int Origin)
- int **SetOutlineColor** (int OutlineID, double Red, double Green, double Blue)
- int **SetOutlineDestination** (int OutlineID, int DestPage, double DestPosition)
- int **SetOutlineDestinationFull** (int OutlineID, int DestPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetOutlineDestinationZoom** (int OutlineID, int DestPage, double DestPosition, int Zoom)
- int **SetOutlineJavaScript** (int OutlineID, string JavaScript)
- int **SetOutlineNamedDestination** (int OutlineID, string DestName)
- int **SetOutlineOpenFile** (int OutlineID, string FileName)
- int **SetOutlineRemoteDestination** (int OutlineID, string FileName, int OpenPage, int Zoom, int DestType, double PntLeft, double PntTop, double PntRight, double PntBottom, int NewWindow)
- int **SetOutlineStyle** (int OutlineID, int SetItalic, int SetBold)
- int **SetOutlineTitle** (int OutlineID, string NewTitle)
- int **SetOutlineWebLink** (int OutlineID, string Link)
- int **SetOverprint** (int StrokingOverprint, int OtherOverprint, int OverprintMode)
- int **SetPDFAMode** (int NewMode)
- int **SetPNGTransparencyColor** (int RedByte, int GreenByte, int BlueByte)
- int **SetPageActionMenu** (string MenuItem)
- int **SetPageBox** (int BoxType, double Left, double Top, double Width, double Height)
- int **SetPageContentFromString** (byte[ ] Source)
- int **SetPageDimensions** (double NewPageWidth, double NewPageHeight)
- int **SetPageLayout** (int NewPageLayout)
- int **SetPageMode** (int NewPageMode)
- int **SetPageSize** (string PaperName)
- int **SetPageThumbnail** ()
- int **SetPageTransparencyGroup** (int CS, int Isolate, int Knockout)
- int **SetPrecision** (int NewPrecision)
- int **SetPrinterDevModeFromString** (byte[ ] Source)
- int **SetRenderCropType** (int NewCropType)
- int **SetRenderDCErasePage** (int NewErasePage)
- int **SetRenderDCOffset** (int NewOffsetX, int NewOffsetY)
- int **SetRenderOptions** (int OptionID, int NewValue)
- int **SetRenderScale** (double NewScale)
- int **SetScale** (double NewScale)
- int **SetSignProcessField** (int SignProcessID, string SignatureFieldName)
- int **SetSignProcessFieldBounds** (int SignProcessID, double Left, double Top, double Width, double Height)
- int **SetSignProcessFieldImageFromFile** (int SignProcessID, string ImageFileName, int Options)
- int **SetSignProcessFieldPage** (int SignProcessID, int SignaturePage)
- int **SetSignProcessInfo** (int SignProcessID, string Reason, string Location, string ContactInfo)
- int **SetSignProcessPFXFromFile** (int SignProcessID, string PFXFileName, string PFXPassword)
- int **SetSignProcessPassthrough** (int SignProcessID, int SignatureLength)
- int **SetSignProcessSubFilter** (int SignProcessID, int SubFilter)
- int **SetTableBorderColor** (int TableID, int BorderIndex, double Red, double Green, double Blue)
- int **SetTableBorderColorCMYK** (int TableID, int BorderIndex, double C, double M, double Y, double K)
- int **SetTableBorderWidth** (int TableID, int BorderIndex, double NewWidth)
- int **SetTableCellAlignment** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int New↩ CellAlignment)

- int **SetTableCellBackgroundColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double Red, double Green, double Blue)
- int **SetTableCellBackgroundColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int Last↩ Column, double C, double M, double Y, double K)
- int **SetTableCellBorderColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double Red, double Green, double Blue)
- int **SetTableCellBorderColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double C, double M, double Y, double K)
- int **SetTableCellBorderWidth** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double NewWidth)
- int **SetTableCellContent** (int TableID, int RowNumber, int ColumnNumber, string HTMLText)
- int **SetTableCellPadding** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int Border↩ Index, double NewPadding)
- int **SetTableCellTextColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double Red, double Green, double Blue)
- int **SetTableCellTextColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double C, double M, double Y, double K)
- int **SetTableCellTextSize** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double NewTextSize)
- int **SetTableColumnWidth** (int TableID, int FirstColumn, int LastColumn, double NewWidth)
- int **SetTableRowHeight** (int TableID, int FirstRow, int LastRow, double NewHeight)
- int **SetTableThinBorders** (int TableID, int ThinBorders, double Red, double Green, double Blue)
- int **SetTableThinBordersCMYK** (int TableID, int ThinBorders, double C, double M, double Y, double K)
- int **SetTempFile** (string FileName)
- int **SetTempPath** (string NewPath)
- int **SetTextAlign** (int TextAlign)
- int **SetTextCharSpacing** (double CharSpacing)
- int **SetTextColor** (double Red, double Green, double Blue)
- int **SetTextColorCMYK** (double C, double M, double Y, double K)
- int **SetTextColorSep** (string ColorName, double Tint)
- int **SetTextExtractionArea** (double Left, double Top, double Width, double Height)
- int **SetTextExtractionOptions** (int OptionID, int NewValue)
- int **SetTextExtractionScaling** (int Options, double Horizontal, double Vertical)
- int **SetTextExtractionWordGap** (double NewWordGap)
- int **SetTextHighlight** (int Highlight)
- int **SetTextHighlightColor** (double Red, double Green, double Blue)
- int **SetTextHighlightColorCMYK** (double C, double M, double Y, double K)
- int **SetTextHighlightColorSep** (string ColorName, double Tint)
- int **SetTextMode** (int TextMode)
- int **SetTextRise** (double Rise)
- int **SetTextScaling** (double ScalePercentage)
- int **SetTextShader** (string ShaderName)
- int **SetTextSize** (double TextSize)
- int **SetTextSpacing** (double Spacing)
- int **SetTextUnderline** (int Underline)
- int **SetTextUnderlineColor** (double Red, double Green, double Blue)
- int **SetTextUnderlineColorCMYK** (double C, double M, double Y, double K)
- int **SetTextUnderlineColorSep** (string ColorName, double Tint)
- int **SetTextUnderlineCustomDash** (string DashPattern, double DashPhase)
- int **SetTextUnderlineDash** (double DashOn, double DashOff)
- int **SetTextUnderlineDistance** (double UnderlineDistance)
- int **SetTextUnderlineWidth** (double UnderlineWidth)
- int **SetTextWordSpacing** (double WordSpacing)
- int **SetTransparency** (int Transparency)
- int **SetViewerPreferences** (int Option, int NewValue)

- int **SetXFAFormFieldAccess** (string XFAFieldName, int NewAccess)
- int **SetXFAFormFieldBorderColor** (string XFAFieldName, double Red, double Green, double Blue)
- int **SetXFAFormFieldBorderPresence** (string XFAFieldName, int NewPresence)
- int **SetXFAFormFieldBorderWidth** (string XFAFieldName, double BorderWidth)
- int **SetXFAFormFieldValue** (string XFAFieldName, string NewValue)
- int **SetXFAFromString** (byte[ ] Source, int Options)
- int **SetupCustomPrinter** (string CustomPrinterName, int Setting, int NewValue)
- int **SignFile** (string InputFileName, string OpenPassword, string SignatureFieldName, string OutputFileName, string PFXFileName, string PFXPassword, string Reason, string Location, string ContactInfo)
- int **SplitPageText** (int Options)
- int **StartPath** (double StartX, double StartY)
- int **StoreCustomDataFromFile** (string Key, string FileName, int Location, int Options)
- int **StoreCustomDataFromString** (string Key, byte[ ] NewValue, int Location, int Options)
- int **StringResultLength** ()
- int **TestTempPath** ()
- int **TransformFile** (string InputFileName, string Password, string OutputFileName, int TransformType, int Options)
- int **UnlockKey** (string LicenseKey)
- int **Unlocked** ()
- int **UpdateAndFlattenFormField** (int Index)
- int **UpdateAppearanceStream** (int Index)
- int **UseKerning** (int Kern)
- int **UseUnsafeContentStreams** (int SafetyLevel)
- **PDFLibrary** (string dllFileName)
- bool **LibraryLoaded** ()
- void **ReleaseLibrary** ()
- int **AddArcToPath** (double CenterX, double CenterY, double TotalAngle)
- int **AddBoxToPath** (double Left, double Top, double Width, double Height)
- int **AddCJKFont** (int CJKFontID)
- int **AddCurveToPath** (double CtAX, double CtAY, double CtBX, double CtBY, double EndX, double EndY)
- int **AddEmbeddedFile** (string FileName, string MIMEType)
- int **AddFileAttachment** (string Title, int EmbeddedFileID)
- int **AddFormFieldChoiceSub** (int Index, string SubName, string DisplayName)
- int **AddFormFieldSub** (int Index, string SubName)
- int **AddFormFont** (int FontID)
- int **AddGlobalJavaScript** (string PackageName, string JavaScript)
- int **AddImageFromFile** (string FileName, int Options)
- int **AddImageFromFileOffset** (string FileName, int Offset, int DataLength, int Options)
- int **AddImageFromString** (byte[ ] Source, int Options)
- int **AddLGIDictToPage** (string DictContent)
- int **AddLineToPath** (double EndX, double EndY)
- int **AddLinkToDestination** (double Left, double Top, double Width, double Height, int DestID, int Options)
- int **AddLinkToEmbeddedFile** (double Left, double Top, double Width, double Height, int EmbeddedFileID, string Title, string Contents, int IconType, int Transpareny)
- int **AddLinkToFile** (double Left, double Top, double Width, double Height, string FileName, int Page, double Position, int NewWindow, int Options)
- int **AddLinkToJavaScript** (double Left, double Top, double Width, double Height, string JavaScript, int Options)
- int **AddLinkToLocalFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AddLinkToPage** (double Left, double Top, double Width, double Height, int Page, double Position, int Options)
- int **AddLinkToWeb** (double Left, double Top, double Width, double Height, string Link, int Options)
- int **AddNoteAnnotation** (double Left, double Top, int AnnotType, double PopupLeft, double PopupTop, double PopupWidth, double PopupHeight, string Title, string Contents, double Red, double Green, double Blue, int Open)

- int **AddOpenTypeFontFromFile** (string FileName, int Options)
- int **AddPageLabels** (int Start, int Style, int Offset, string Prefix)
- int **AddSVGAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AddSWFAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, string Title, int Options)
- int **AddSeparationColor** (string ColorName, double C, double M, double Y, double K, int Options)
- int **AddStampAnnotation** (double Left, double Top, double Width, double Height, int StampType, string Title, string Contents, double Red, double Green, double Blue, int Options)
- int **AddStandardFont** (int StandardFontID)
- int **AddSubsettedFont** (string FontName, int CharsetIndex, string SubsetChars)
- int **AddTextMarkupAnnotation** (int MarkupType, double Left, double Top, double Width, double Height)
- int **AddToFileList** (string ListName, string FileName)
- int **AddTrueTypeFont** (string FontName, int Embed)
- int **AddTrueTypeFontFromFile** (string FileName)
- int **AddTrueTypeSubsettedFont** (string FontName, string SubsetChars, int Options)
- int **AddType1Font** (string FileName)
- int **AddU3DAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AnalyseFile** (string InputFileName, string Password)
- int **AnnotationCount** ()
- int **AnsiStringResultLength** ()
- int **AppendSpace** (double RelativeSpace)
- int **AppendTableColumns** (int TableID, int NewColumnCount)
- int **AppendTableRows** (int TableID, int NewRowCount)
- int **AppendText** (string Text)
- int **ApplyStyle** (string StyleName)
- int **AttachAnnotToForm** (int Index)
- int **BalanceContentStream** ()
- int **BalancePageTree** (int Options)
- int **BeginPageUpdate** ()
- int **CapturePage** (int Page)
- int **CapturePageEx** (int Page, int Options)
- int **CharWidth** (int CharCode)
- int **CheckFileCompliance** (string InputFileName, string Password, int ComplianceTest, int Options)
- int **CheckObjects** ()
- int **CheckPageAnnots** ()
- int **ClearFileList** (string ListName)
- int **ClearImage** (int ImageID)
- int **ClearPageLabels** ()
- int **ClearTextFormatting** ()
- int **CloneOutlineAction** (int OutlineID)
- int **ClonePages** (int StartPage, int EndPage, int RepeatCount)
- int **CloseOutline** (int OutlineID)
- int **ClosePath** ()
- int **CombineContentStreams** ()
- int **CompareOutlines** (int FirstOutlineID, int SecondOutlineID)
- int **CompressContent** ()
- int **CompressFonts** (int Compress)
- int **CompressImages** (int Compress)
- int **CompressPage** ()
- int **ContentStreamCount** ()
- int **ContentStreamSafe** ()
- int **CopyPageRanges** (int DocumentID, string RangeList)
- int **CopyPageRangesEx** (int DocumentID, string RangeList, int Options)

- int **CreateNewObject** ()
- int **CreateTable** (int RowCount, int ColumnCount)
- int **DAAppendFile** (int FileHandle)
- int **DACapturePage** (int FileHandle, int PageRef)
- int **DACapturePageEx** (int FileHandle, int PageRef, int Options)
- int **DACloseFile** (int FileHandle)
- int **DADrawCapturedPage** (int FileHandle, int DACaptureID, int DestPageRef, double PntLeft, double Pnt↩ Bottom, double PntWidth, double PntHeight)
- int **DADrawRotatedCapturedPage** (int FileHandle, int DACaptureID, int DestPageRef, double PntLeft, double PntBottom, double PntWidth, double PntHeight, double Angle)
- int **DAEmbedFileStreams** (int FileHandle, string RootPath)
- string **DAExtractPageText** (int FileHandle, int PageRef, int Options)
- int **DAExtractPageTextBlocks** (int FileHandle, int PageRef, int ExtractOptions)
- int **DAFindPage** (int FileHandle, int Page)
- int **DAGetAnnotationCount** (int FileHandle, int PageRef)
- int **DAGetFormFieldCount** (int FileHandle)
- string **DAGetFormFieldTitle** (int FileHandle, int FieldIndex)
- string **DAGetFormFieldValue** (int FileHandle, int FieldIndex)
- byte[ ] **DAGetImageDataToString** (int FileHandle, int ImageListID, int ImageIndex)
- double **DAGetImageDblProperty** (int FileHandle, int ImageListID, int ImageIndex, int PropertyID)
- int **DAGetImageIntProperty** (int FileHandle, int ImageListID, int ImageIndex, int PropertyID)
- int **DAGetImageListCount** (int FileHandle, int ImageListID)
- string **DAGetInformation** (int FileHandle, string Key)
- int **DAGetObjectCount** (int FileHandle)
- byte[ ] **DAGetObjectToString** (int FileHandle, int ObjectNumber)
- double **DAGetPageBox** (int FileHandle, int PageRef, int BoxIndex, int Dimension)
- byte[ ] **DAGetPageContentToString** (int FileHandle, int PageRef)
- int **DAGetPageCount** (int FileHandle)
- double **DAGetPageHeight** (int FileHandle, int PageRef)
- int **DAGetPageImageList** (int FileHandle, int PageRef)
- double **DAGetPageWidth** (int FileHandle, int PageRef)
- double **DAGetTextBlockBound** (int TextBlockListID, int Index, int BoundIndex)
- double **DAGetTextBlockCharWidth** (int TextBlockListID, int Index, int CharIndex)
- double **DAGetTextBlockColor** (int TextBlockListID, int Index, int ColorComponent)
- int **DAGetTextBlockColorType** (int TextBlockListID, int Index)
- int **DAGetTextBlockCount** (int TextBlockListID)
- string **DAGetTextBlockFontName** (int TextBlockListID, int Index)
- double **DAGetTextBlockFontSize** (int TextBlockListID, int Index)
- string **DAGetTextBlockText** (int TextBlockListID, int Index)
- int **DAHasPageBox** (int FileHandle, int PageRef, int BoxIndex)
- int **DAHidePage** (int FileHandle, int PageRef)
- int **DAMovePage** (int FileHandle, int PageRef, int TargetPageRef, int Options)
- int **DANewPage** (int FileHandle)
- int **DANewPages** (int FileHandle, int PageCount)
- int **DAOpenFile** (string InputFileName, string Password)
- int **DAOpenFileReadOnly** (string InputFileName, string Password)
- int **DAPageRotation** (int FileHandle, int PageRef)
- int **DAReleaseImageList** (int FileHandle, int ImageListID)
- int **DAReleaseTextBlocks** (int TextBlockListID)
- int **DARemoveUsageRights** (int FileHandle)
- int **DARenderPageToDC** (int FileHandle, int PageRef, int DPI, IntPtr DC)
- int **DARenderPageToFile** (int FileHandle, int PageRef, int Options, int DPI, string FileName)
- byte[ ] **DARenderPageToString** (int FileHandle, int PageRef, int Options, int DPI)
- int **DARotatePage** (int FileHandle, int PageRef, int Angle, int Options)
- int **DASaveAsFile** (int FileHandle, string OutputFileName)

- int **DASaveImageDataToFile** (int FileHandle, int ImageListID, int ImageIndex, string ImageFileName)
- int **DASetInformation** (int FileHandle, string Key, string NewValue)
- int **DASetPageBox** (int FileHandle, int PageRef, int BoxIndex, double X1, double Y1, double X2, double Y2)
- int **DASetPageSize** (int FileHandle, int PageRef, double PntWidth, double PntHeight)
- int **DASetTextExtractionArea** (double Left, double Top, double Width, double Height)
- int **DASetTextExtractionOptions** (int OptionID, int NewValue)
- int **DASetTextExtractionScaling** (int Options, double Horizontal, double Vertical)
- int **DASetTextExtractionWordGap** (double NewWordGap)
- int **DAShiftedHeader** (int FileHandle)
- int **Decrypt** ()
- int **DecryptFile** (string InputFileName, string OutputFileName, string Password)
- int **DeleteAnalysis** (int AnalysisID)
- int **DeleteAnnotation** (int Index)
- int **DeleteContentStream** ()
- int **DeleteFormField** (int Index)
- int **DeleteOptionalContentGroup** (int OptionalContentGroupID)
- int **DeletePageLGIDict** (int DictIndex)
- int **DeletePages** (int StartPage, int PageCount)
- int **DocJavaScriptAction** (string ActionType, string JavaScript)
- int **DocumentCount** ()
- int **DrawArc** (double XPos, double YPos, double Radius, double StartAngle, double EndAngle, int Pie, int DrawOptions)
- int **DrawBarcode** (double Left, double Top, double Width, double Height, string Text, int Barcode, int Options)
- int **DrawBox** (double Left, double Top, double Width, double Height, int DrawOptions)
- int **DrawCapturedPage** (int CaptureID, double Left, double Top, double Width, double Height)
- int **DrawCapturedPageMatrix** (int CaptureID, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawCircle** (double XPos, double YPos, double Radius, int DrawOptions)
- int **DrawDataMatrixSymbol** (double Left, double Top, double ModuleSize, string Text, int Encoding, int SymbolSize, int Options)
- int **DrawEllipse** (double XPos, double YPos, double Width, double Height, int DrawOptions)
- int **DrawEllipticArc** (double XPos, double YPos, double Width, double Height, double StartAngle, double EndAngle, int Pie, int DrawOptions)
- int **DrawHTMLText** (double Left, double Top, double Width, string HTMLText)
- string **DrawHTMLTextBox** (double Left, double Top, double Width, double Height, string HTMLText)
- string **DrawHTMLTextBoxMatrix** (double Width, double Height, string HTMLText, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawImage** (double Left, double Top, double Width, double Height)
- int **DrawImageMatrix** (double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawIntelligentMailBarcode** (double Left, double Top, double BarWidth, double FullBarHeight, double TrackerHeight, double SpaceWidth, string BarcodeData, int Options)
- int **DrawLine** (double StartX, double StartY, double EndX, double EndY)
- int **DrawMultiLineText** (double XPos, double YPos, string Delimiter, string Text)
- int **DrawPDF417Symbol** (double Left, double Top, string Text, int Options)
- int **DrawPDF417SymbolEx** (double Left, double Top, string Text, int Options, int FixedColumns, int FixedRows, int ErrorLevel, double ModuleSize, double HeightWidthRatio)
- int **DrawPath** (int PathOptions)
- int **DrawPathEvenOdd** (int PathOptions)
- int **DrawPostScriptXObject** (int PSRef)
- int **DrawRotatedBox** (double Left, double Bottom, double Width, double Height, double Angle, int DrawOptions)
- int **DrawRotatedCapturedPage** (int CaptureID, double Left, double Bottom, double Width, double Height, double Angle)
- int **DrawRotatedImage** (double Left, double Bottom, double Width, double Height, double Angle)
- int **DrawRotatedMultiLineText** (double XPos, double YPos, double Angle, string Delimiter, string Text)

- int **DrawRotatedText** (double XPos, double YPos, double Angle, string Text)
- int **DrawRotatedTextBox** (double Left, double Top, double Width, double Height, double Angle, string Text, int Options)
- int **DrawRoundedBox** (double Left, double Top, double Width, double Height, double Radius, int Draw↩ Options)
- int **DrawScaledImage** (double Left, double Top, double Scale)
- int **DrawSpacedText** (double XPos, double YPos, double Spacing, string Text)
- double **DrawTableRows** (int TableID, double Left, double Top, double Height, int FirstRow, int LastRow)
- int **DrawText** (double XPos, double YPos, string Text)
- int **DrawTextArc** (double XPos, double YPos, double Radius, double Angle, string Text, int DrawOptions)
- int **DrawTextBox** (double Left, double Top, double Width, double Height, string Text, int Options)
- int **DrawTextBoxMatrix** (double Width, double Height, string Text, int Options, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawWrappedText** (double XPos, double YPos, double Width, string Text)
- int **EditableContentStream** ()
- int **EmbedFile** (string Title, string FileName, string MIMEType)
- int **EmbeddedFileCount** ()
- int **EncapsulateContentStream** ()
- int **EncodePermissions** (int CanPrint, int CanCopy, int CanChange, int CanAddNotes, int CanFillFields, int CanCopyAccess, int CanAssemble, int CanPrintFull)
- int **Encrypt** (string Owner, string User, int Strength, int Permissions)
- int **EncryptFile** (string InputFileName, string OutputFileName, string Owner, string User, int Strength, int Permissions)
- int **EncryptWithFingerprint** (string Fingerprint)
- int **EncryptionAlgorithm** ()
- int **EncryptionStatus** ()
- int **EncryptionStrength** ()
- int **EndPageUpdate** ()
- int **EndSignProcessToFile** (int SignProcessID, string OutputFile)
- byte[ ] **EndSignProcessToString** (int SignProcessID)
- byte[ ] **ExtractFilePageContentToString** (string InputFileName, string Password, int Page)
- string **ExtractFilePageText** (string InputFileName, string Password, int Page, int Options)
- int **ExtractFilePageTextBlocks** (string InputFileName, string Password, int Page, int Options)
- int **ExtractFilePages** (string InputFileName, string Password, string OutputFileName, string RangeList)
- int **ExtractFilePagesEx** (string InputFileName, string Password, string OutputFileName, string RangeList, int Options)
- int **ExtractPageRanges** (string RangeList)
- int **ExtractPageTextBlocks** (int ExtractOptions)
- int **ExtractPages** (int StartPage, int PageCount)
- int **FileListCount** (string ListName)
- string **FileListItem** (string ListName, int Index)
- int **FindFonts** ()
- int **FindFormFieldByTitle** (string Title)
- int **FindImages** ()
- int **FitImage** (double Left, double Top, double Width, double Height, int HAlign, int VAlign, int Rotate)
- int **FitRotatedTextBox** (double Left, double Top, double Width, double Height, double Angle, string Text, int Options)
- int **FitTextBox** (double Left, double Top, double Width, double Height, string Text, int Options)
- int **FlattenAnnot** (int Index, int Options)
- int **FlattenFormField** (int Index)
- int **FontCount** ()
- string **FontFamily** ()
- int **FontHasKerning** ()
- string **FontName** ()
- string **FontReference** ()

- int **FontSize** ()
- int **FontType** ()
- int **FormFieldCount** ()
- int **FormFieldHasParent** (int Index)
- int **FormFieldJavaScriptAction** (int Index, string ActionType, string JavaScript)
- int **FormFieldWebLinkAction** (int Index, string ActionType, string Link)
- int **GetActionDest** (int ActionID)
- string **GetActionType** (int ActionID)
- string **GetActionURL** (int ActionID)
- string **GetAnalysisInfo** (int AnalysisID, int AnalysisItem)
- int **GetAnnotActionID** (int Index)
- double **GetAnnotDblProperty** (int Index, int Tag)
- int **GetAnnotDest** (int Index)
- int **GetAnnotIntProperty** (int Index, int Tag)
- int **GetAnnotQuadCount** (int Index)
- double **GetAnnotQuadPoints** (int Index, int QuadNumber, int PointNumber)
- int **GetAnnotSoundToFile** (int Index, int Options, string SoundFileName)
- byte[ ] **GetAnnotSoundToString** (int Index, int Options)
- string **GetAnnotStrProperty** (int Index, int Tag)
- double **GetBarcodeWidth** (double NominalWidth, string Text, int Barcode)
- string **GetBaseURL** ()
- int **GetCSDictEPSG** (int CSDictID)
- int **GetCSDictType** (int CSDictID)
- string **GetCSDictWKT** (int CSDictID)
- IntPtr **GetCanvasDC** (int CanvasWidth, int CanvasHeight)
- string **GetCatalogInformation** (string Key)
- byte[ ] **GetContentStreamToString** ()
- string **GetCustomInformation** (string Key)
- string **GetCustomKeys** (int Location)
- string **GetDefaultPrinterName** ()
- string **GetDestName** (int DestID)
- int **GetDestPage** (int DestID)
- int **GetDestType** (int DestID)
- double **GetDestValue** (int DestID, int ValueKey)
- string **GetDocJavaScript** (string ActionType)
- string **GetDocumentFileName** ()
- int **GetDocumentFileSize** ()
- int **GetDocumentID** (int Index)
- string **GetDocumentIdentifier** (int Part, int Options)
- string **GetDocumentMetadata** ()
- int **GetDocumentRepaired** ()
- string **GetDocumentResourceList** ()
- int **GetEmbeddedFileContentToFile** (int Index, string FileName)
- byte[ ] **GetEmbeddedFileContentToString** (int Index)
- int **GetEmbeddedFileID** (int Index)
- int **GetEmbeddedFileIntProperty** (int Index, int Tag)
- string **GetEmbeddedFileStrProperty** (int Index, int Tag)
- string **GetEncryptionFingerprint** ()
- string **GetFileMetadata** (string InputFileName, string Password)
- int **GetFirstChildOutline** (int OutlineID)
- int **GetFirstOutline** ()
- int **GetFontEncoding** ()
- int **GetFontFlags** (int FontFlagItemID)
- int **GetFontID** (int Index)
- int **GetFontObjectNumber** ()

- int **GetFormFieldActionID** (int Index, string TriggerEvent)
- int **GetFormFieldAlignment** (int Index)
- int **GetFormFieldAnnotFlags** (int Index)
- double **GetFormFieldBackgroundColor** (int Index, int ColorComponent)
- int **GetFormFieldBackgroundColorType** (int Index)
- double **GetFormFieldBorderColor** (int Index, int ColorComponent)
- int **GetFormFieldBorderColorType** (int Index)
- double **GetFormFieldBorderProperty** (int Index, int PropKey)
- int **GetFormFieldBorderStyle** (int Index)
- double **GetFormFieldBound** (int Index, int Edge)
- string **GetFormFieldCaption** (int Index)
- int **GetFormFieldCheckStyle** (int Index)
- string **GetFormFieldChildTitle** (int Index)
- int **GetFormFieldChoiceType** (int Index)
- double **GetFormFieldColor** (int Index, int ColorComponent)
- int **GetFormFieldComb** (int Index)
- string **GetFormFieldDefaultValue** (int Index)
- string **GetFormFieldDescription** (int Index)
- int **GetFormFieldFlags** (int Index)
- string **GetFormFieldFontName** (int Index)
- string **GetFormFieldJavaScript** (int Index, string ActionType)
- int **GetFormFieldKidCount** (int Index)
- int **GetFormFieldMaxLen** (int Index)
- int **GetFormFieldNoExport** (int Index)
- int **GetFormFieldPage** (int Index)
- int **GetFormFieldPrintable** (int Index)
- int **GetFormFieldReadOnly** (int Index)
- int **GetFormFieldRequired** (int Index)
- string **GetFormFieldRichTextString** (int Index, string Key)
- int **GetFormFieldRotation** (int Index)
- int **GetFormFieldSubCount** (int Index)
- string **GetFormFieldSubDisplayName** (int Index, int SubIndex)
- string **GetFormFieldSubName** (int Index, int SubIndex)
- int **GetFormFieldSubTempIndex** (int Index, int SubIndex)
- int **GetFormFieldTabOrder** (int Index)
- int **GetFormFieldTabOrderEx** (int Index, int Options)
- int **GetFormFieldTextFlags** (int Index, int ValueKey)
- double **GetFormFieldTextSize** (int Index)
- string **GetFormFieldTitle** (int Index)
- int **GetFormFieldType** (int Index)
- string **GetFormFieldValue** (int Index)
- string **GetFormFieldValueByTitle** (string Title)
- int **GetFormFieldVisible** (int Index)
- string **GetFormFieldWebLink** (int Index, string ActionType)
- int **GetFormFontCount** ()
- string **GetFormFontName** (int FontIndex)
- string **GetGlobalJavaScript** (string PackageName)
- double **GetHTMLTextHeight** (double Width, string HTMLText)
- int **GetHTMLTextLineCount** (double Width, string HTMLText)
- double **GetHTMLTextWidth** (double MaxWidth, string HTMLText)
- int **GetImageID** (int Index)
- int **GetImageListCount** (int ImageListID)
- byte[ ] **GetImageListItemDataToString** (int ImageListID, int ImageIndex, int Options)
- double **GetImageListItemDblProperty** (int ImageListID, int ImageIndex, int PropertyID)
- int **GetImageListItemIntProperty** (int ImageListID, int ImageIndex, int PropertyID)

- int **GetImageMeasureDict** ()
- int **GetImagePageCount** (string FileName)
- int **GetImagePageCountFromString** (byte[ ] Source)
- int **GetImagePtDataDict** ()
- string **GetInformation** (int Key)
- string **GetInstalledFontsByCharset** (int CharsetIndex, int Options)
- string **GetInstalledFontsByCodePage** (int CodePage, int Options)
- int **GetKerning** (string CharPair)
- string **GetLatestPrinterNames** ()
- int **GetMaxObjectNumber** ()
- int **GetMeasureDictBoundsCount** (int MeasureDictID)
- double **GetMeasureDictBoundsItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictCoordinateSystem** (int MeasureDictID)
- int **GetMeasureDictDCSDict** (int MeasureDictID)
- int **GetMeasureDictGCSDict** (int MeasureDictID)
- int **GetMeasureDictGPTSCount** (int MeasureDictID)
- double **GetMeasureDictGPTSItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictLPTSCount** (int MeasureDictID)
- double **GetMeasureDictLPTSItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictPDU** (int MeasureDictID, int UnitIndex)
- int **GetNamedDestination** (string DestName)
- int **GetNextOutline** (int OutlineID)
- int **GetObjectCount** ()
- int **GetObjectDecodeError** (int ObjectNumber)
- byte[ ] **GetObjectToString** (int ObjectNumber)
- int **GetOpenActionDestination** ()
- string **GetOpenActionJavaScript** ()
- int **GetOptionalContentConfigCount** ()
- int **GetOptionalContentConfigLocked** (int OptionalContentConfigID, int OptionalContentGroupID)
- int **GetOptionalContentConfigOrderCount** (int OptionalContentConfigID)
- int **GetOptionalContentConfigOrderItemID** (int OptionalContentConfigID, int ItemIndex)
- string **GetOptionalContentConfigOrderItemLabel** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigOrderItemLevel** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigOrderItemType** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigState** (int OptionalContentConfigID, int OptionalContentGroupID)
- int **GetOptionalContentGroupID** (int Index)
- string **GetOptionalContentGroupName** (int OptionalContentGroupID)
- int **GetOptionalContentGroupPrintable** (int OptionalContentGroupID)
- int **GetOptionalContentGroupVisible** (int OptionalContentGroupID)
- int **GetOrigin** ()
- int **GetOutlineActionID** (int OutlineID)
- double **GetOutlineColor** (int OutlineID, int ColorComponent)
- int **GetOutlineDest** (int OutlineID)
- int **GetOutlineID** (int Index)
- string **GetOutlineJavaScript** (int OutlineID)
- int **GetOutlineObjectNumber** (int OutlineID)
- string **GetOutlineOpenFile** (int OutlineID)
- int **GetOutlinePage** (int OutlineID)
- int **GetOutlineStyle** (int OutlineID)
- string **GetOutlineWebLink** (int OutlineID)
- double **GetPageBox** (int BoxType, int Dimension)
- string **GetPageColorSpaces** (int Options)
- byte[ ] **GetPageContentToString** ()
- int **GetPageImageList** (int Options)
- string **GetPageJavaScript** (string ActionType)

- string **GetPageLGIDictContent** (int DictIndex)
- int **GetPageLGIDictCount** ()
- string **GetPageLabel** (int Page)
- int **GetPageLayout** ()
- byte[ ] **GetPageMetricsToString** (int StartPage, int EndPage, int Options)
- int **GetPageMode** ()
- string **GetPageText** (int ExtractOptions)
- int **GetPageViewPortCount** ()
- int **GetPageViewPortID** (int Index)
- int **GetParentOutline** (int OutlineID)
- int **GetPrevOutline** (int OutlineID)
- string **GetPrinterBins** (string PrinterName)
- byte[ ] **GetPrinterDevModeToString** (string PrinterName)
- string **GetPrinterMediaTypes** (string PrinterName)
- string **GetPrinterNames** ()
- double **GetRenderScale** ()
- int **GetSignProcessByteRange** (int SignProcessID, int ArrayPosition)
- int **GetSignProcessResult** (int SignProcessID)
- int **GetStringListCount** (int StringListID)
- string **GetStringListItem** (int StringListID, int ItemIndex)
- double **GetTableCellDblProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- int **GetTableCellIntProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- string **GetTableCellStrProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- int **GetTableColumnCount** (int TableID)
- int **GetTableLastDrawnRow** (int TableID)
- int **GetTableRowCount** (int TableID)
- string **GetTempPath** ()
- double **GetTextAscent** ()
- double **GetTextBlockBound** (int TextBlockListID, int Index, int BoundIndex)
- double **GetTextBlockCharWidth** (int TextBlockListID, int Index, int CharIndex)
- double **GetTextBlockColor** (int TextBlockListID, int Index, int ColorComponent)
- int **GetTextBlockColorType** (int TextBlockListID, int Index)
- int **GetTextBlockCount** (int TextBlockListID)
- string **GetTextBlockFontName** (int TextBlockListID, int Index)
- double **GetTextBlockFontSize** (int TextBlockListID, int Index)
- string **GetTextBlockText** (int TextBlockListID, int Index)
- double **GetTextBound** (int Edge)
- double **GetTextDescent** ()
- double **GetTextHeight** ()
- double **GetTextSize** ()
- double **GetTextWidth** (string Text)
- string **GetUnicodeCharactersFromEncoding** (int Encoding)
- double **GetViewPortBBox** (int ViewPortID, int Dimension)
- int **GetViewPortMeasureDict** (int ViewPortID)
- string **GetViewPortName** (int ViewPortID)
- int **GetViewPortPtDataDict** (int ViewPortID)
- int **GetViewerPreferences** (int Option)
- string **GetWrappedText** (double Width, string Delimiter, string Text)
- string **GetWrappedTextBreakString** (double Width, string Delimiter, string Text)
- double **GetWrappedTextHeight** (double Width, string Text)
- int **GetWrappedTextLineCount** (double Width, string Text)
- int **GetXFAFormFieldCount** ()
- string **GetXFAFormFieldName** (int Index)
- string **GetXFAFormFieldNames** (string Delimiter)
- string **GetXFAFormFieldValue** (string XFAFieldName)

- byte[ ] **GetXFAToString** (int Options)
- int **GlobalJavaScriptCount** ()
- string **GlobalJavaScriptPackageName** (int Index)
- int **HasFontResources** ()
- int **HasPageBox** (int BoxType)
- int **HidePage** ()
- int **ImageCount** ()
- int **ImageFillColor** ()
- int **ImageHeight** ()
- int **ImageHorizontalResolution** ()
- int **ImageResolutionUnits** ()
- int **ImageType** ()
- int **ImageVerticalResolution** ()
- int **ImageWidth** ()
- int **ImportEMFFromFile** (string FileName, int FontOptions, int GeneralOptions)
- int **InsertPages** (int StartPage, int PageCount)
- int **InsertTableColumns** (int TableID, int Position, int NewColumnCount)
- int **InsertTableRows** (int TableID, int Position, int NewRowCount)
- int **IsAnnotFormField** (int Index)
- int **IsTaggedPDF** ()
- int **LastErrorCode** ()
- string **LastRenderError** ()
- string **LibraryVersion** ()
- string **LicenseInfo** ()
- int **Linearized** ()
- int **LoadFromCanvasDC** (double DPI, int Options)
- int **LoadFromFile** (string FileName, string Password)
- int **LoadFromString** (byte[ ] Source, string Password)
- int **LoadState** ()
- int **MergeDocument** (int DocumentID)
- int **MergeFileList** (string ListName, string OutputFileName)
- int **MergeFileListFast** (string ListName, string OutputFileName)
- int **MergeFiles** (string FirstFileName, string SecondFileName, string OutputFileName)
- int **MergeTableCells** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn)
- int **MoveContentStream** (int FromPosition, int ToPosition)
- int **MoveOutlineAfter** (int OutlineID, int SiblingID)
- int **MoveOutlineBefore** (int OutlineID, int SiblingID)
- int **MovePage** (int NewPosition)
- int **MovePath** (double NewX, double NewY)
- int **MultiplyScale** (double MultScaleBy)
- int **NewChildFormField** (int Index, string Title, int FieldType)
- int **NewContentStream** ()
- string **NewCustomPrinter** (string OriginalPrinterName)
- int **NewDestination** (int DestPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **NewDocument** ()
- int **NewFormField** (string Title, int FieldType)
- int **NewNamedDestination** (string DestName, int DestID)
- int **NewOptionalContentGroup** (string GroupName)
- int **NewOutline** (int Parent, string Title, int DestPage, double DestPosition)
- int **NewPage** ()
- int **NewPageFromCanvasDC** (int DPI, int Options)
- int **NewPages** (int PageCount)
- int **NewPostScriptXObject** (string PS)

- int **NewRGBAxialShader** (string ShaderName, double StartX, double StartY, double StartRed, double StartGreen, double StartBlue, double EndX, double EndY, double EndRed, double EndGreen, double EndBlue, int Extend)
- int **NewSignProcessFromFile** (string InputFile, string Password)
- int **NewSignProcessFromString** (byte[ ] Source, string Password)
- int **NewStaticOutline** (int Parent, string Title)
- int **NewTilingPatternFromCapturedPage** (string PatternName, int CaptureID)
- int **NoEmbedFontListAdd** (string FontName)
- int **NoEmbedFontListCount** ()
- string **NoEmbedFontListGet** (int Index)
- int **NoEmbedFontListRemoveAll** ()
- int **NoEmbedFontListRemoveIndex** (int Index)
- int **NoEmbedFontListRemoveName** (string FontName)
- int **NormalizePage** (int NormalizeOptions)
- int **OpenOutline** (int OutlineID)
- int **OptionalContentGroupCount** ()
- int **OutlineCount** ()
- string **OutlineTitle** (int OutlineID)
- int **PageCount** ()
- double **PageHeight** ()
- int **PageJavaScriptAction** (string ActionType, string JavaScript)
- int **PageRotation** ()
- double **PageWidth** ()
- int **PrintDocument** (string PrinterName, int StartPage, int EndPage, int Options)
- int **PrintDocumentToFile** (string PrinterName, int StartPage, int EndPage, int Options, string FileName)
- int **PrintOptions** (int PageScaling, int AutoRotateCenter, string Title)
- int **PrintPages** (string PrinterName, string PageRanges, int Options)
- int **PrintPagesToFile** (string PrinterName, string PageRanges, int Options, string FileName)
- int **ReleaseImage** (int ImageID)
- int **ReleaseImageList** (int ImageListID)
- int **ReleaseSignProcess** (int SignProcessID)
- int **ReleaseStringList** (int StringListID)
- int **ReleaseTextBlocks** (int TextBlockListID)
- int **RemoveAppearanceStream** (int Index)
- int **RemoveCustomInformation** (string Key)
- int **RemoveDocument** (int DocumentID)
- int **RemoveEmbeddedFile** (int Index)
- int **RemoveFormFieldBackgroundColor** (int Index)
- int **RemoveFormFieldBorderColor** (int Index)
- int **RemoveGlobalJavaScript** (string PackageName)
- int **RemoveOpenAction** ()
- int **RemoveOutline** (int OutlineID)
- int **RemoveSharedContentStreams** ()
- int **RemoveStyle** (string StyleName)
- int **RemoveUsageRights** ()
- int **RemoveXFAEntries** (int Options)
- int **RenderDocumentToFile** (int DPI, int StartPage, int EndPage, int Options, string FileName)
- int **RenderPageToDC** (int DPI, int Page, IntPtr DC)
- int **RenderPageToFile** (int DPI, int Page, int Options, string FileName)
- byte[ ] **RenderPageToString** (int DPI, int Page, int Options)
- int **ReplaceFonts** ()
- int **ReplaceImage** (int OriginalImageID, int NewImageID)
- int **ReplaceTag** (string Tag, string NewValue)
- int **RequestPrinterStatus** (int StatusCommand)
- int **RetrieveCustomDataToFile** (string Key, string FileName, int Location)

- byte[ ] **RetrieveCustomDataToString** (string Key, int Location)
- int **ReverseImage** (int Reset)
- int **RotatePage** (int PageRotation)
- int **SaveFontToFile** (string FileName)
- int **SaveImageListItemDataToFile** (int ImageListID, int ImageIndex, int Options, string ImageFileName)
- int **SaveImageToFile** (string FileName)
- byte[ ] **SaveImageToString** ()
- int **SaveState** ()
- int **SaveStyle** (string StyleName)
- int **SaveToFile** (string FileName)
- byte[ ] **SaveToString** ()
- int **SecurityInfo** (int SecurityItem)
- int **SelectContentStream** (int NewIndex)
- int **SelectDocument** (int DocumentID)
- int **SelectFont** (int FontID)
- int **SelectImage** (int ImageID)
- int **SelectPage** (int PageNumber)
- int **SelectRenderer** (int RendererID)
- int **SelectedDocument** ()
- int **SelectedFont** ()
- int **SelectedImage** ()
- int **SelectedPage** ()
- int **SetActionURL** (int ActionID, string NewURL)
- int **SetAnnotBorderColor** (int Index, double Red, double Green, double Blue)
- int **SetAnnotBorderStyle** (int Index, double Width, int Style, double DashOn, double DashOff)
- int **SetAnnotContents** (int Index, string NewContents)
- int **SetAnnotDblProperty** (int Index, int Tag, double NewValue)
- int **SetAnnotIntProperty** (int Index, int Tag, int NewValue)
- int **SetAnnotQuadPoints** (int Index, int QuadNumber, double X1, double Y1, double X2, double Y2, double X3, double Y3, double X4, double Y4)
- int **SetAnnotRect** (int Index, double Left, double Top, double Width, double Height)
- int **SetAnnotStrProperty** (int Index, int Tag, string NewValue)
- int **SetAnsiMode** (int NewAnsiMode)
- int **SetBaseURL** (string NewBaseURL)
- int **SetBlendMode** (int BlendMode)
- int **SetBreakString** (string NewBreakString)
- int **SetCSDictEPSG** (int CSDictID, int NewEPSG)
- int **SetCSDictType** (int CSDictID, int NewDictType)
- int **SetCSDictWKT** (int CSDictID, string NewWKT)
- int **SetCairoFileName** (string FileName)
- int **SetCapturedPageOptional** (int CaptureID, int OptionalContentGroupID)
- int **SetCapturedPageTransparencyGroup** (int CaptureID, int CS, int Isolate, int Knockout)
- int **SetCatalogInformation** (string Key, string NewValue)
- int **SetCharWidth** (int CharCode, int NewWidth)
- int **SetClippingPath** ()
- int **SetClippingPathEvenOdd** ()
- int **SetCompatibility** (int CompatibilityItem, int CompatibilityMode)
- int **SetContentStreamFromString** (byte[ ] Source)
- int **SetContentStreamOptional** (int OptionalContentGroupID)
- int **SetCropBox** (double Left, double Top, double Width, double Height)
- int **SetCustomInformation** (string Key, string NewValue)
- int **SetCustomLineDash** (string DashPattern, double DashPhase)
- int **SetDestProperties** (int DestID, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetDestValue** (int DestID, int ValueKey, double NewValue)

- int **SetDocumentMetadata** (string XMP)
- int **SetEmbeddedFileStrProperty** (int Index, int Tag, string NewValue)
- int **SetFillColor** (double Red, double Green, double Blue)
- int **SetFillColorCMYK** (double C, double M, double Y, double K)
- int **SetFillColorSep** (string ColorName, double Tint)
- int **SetFillShader** (string ShaderName)
- int **SetFillTilingPattern** (string PatternName)
- int **SetFindImagesMode** (int NewFindImagesMode)
- int **SetFontEncoding** (int Encoding)
- int **SetFontFlags** (int Fixed, int Serif, int Symbolic, int Script, int Italic, int AllCap, int SmallCap, int ForceBold)
- int **SetFormFieldAlignment** (int Index, int Alignment)
- int **SetFormFieldAnnotFlags** (int Index, int NewFlags)
- int **SetFormFieldBackgroundColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldBackgroundColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldBackgroundColorGray** (int Index, double Gray)
- int **SetFormFieldBackgroundColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldBorderColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldBorderColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldBorderColorGray** (int Index, double Gray)
- int **SetFormFieldBorderColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldBorderStyle** (int Index, double Width, int Style, double DashOn, double DashOff)
- int **SetFormFieldBounds** (int Index, double Left, double Top, double Width, double Height)
- int **SetFormFieldCalcOrder** (int Index, int Order)
- int **SetFormFieldCaption** (int Index, string NewCaption)
- int **SetFormFieldCheckStyle** (int Index, int CheckStyle, int Position)
- int **SetFormFieldChildTitle** (int Index, string NewTitle)
- int **SetFormFieldChoiceType** (int Index, int ChoiceType)
- int **SetFormFieldColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldComb** (int Index, int Comb)
- int **SetFormFieldDefaultValue** (int Index, string NewDefaultValue)
- int **SetFormFieldDescription** (int Index, string NewDescription)
- int **SetFormFieldFlags** (int Index, int NewFlags)
- int **SetFormFieldFont** (int Index, int FontIndex)
- int **SetFormFieldHighlightMode** (int Index, int NewMode)
- int **SetFormFieldIcon** (int Index, int IconType, int CaptureID)
- int **SetFormFieldIconStyle** (int Index, int Placement, int Scale, int ScaleType, int HorizontalShift, int Vertical↩ Shift)
- int **SetFormFieldMaxLen** (int Index, int NewMaxLen)
- int **SetFormFieldNoExport** (int Index, int NoExport)
- int **SetFormFieldOptional** (int Index, int OptionalContentGroupID)
- int **SetFormFieldPage** (int Index, int NewPage)
- int **SetFormFieldPrintable** (int Index, int Printable)
- int **SetFormFieldReadOnly** (int Index, int ReadOnly)
- int **SetFormFieldRequired** (int Index, int Required)
- int **SetFormFieldResetAction** (int Index, string ActionType)
- int **SetFormFieldRichTextString** (int Index, string Key, string NewValue)
- int **SetFormFieldRotation** (int Index, int Angle)
- int **SetFormFieldSignatureImage** (int Index, int ImageID, int Options)
- int **SetFormFieldStandardFont** (int Index, int StandardFontID)
- int **SetFormFieldSubmitAction** (int Index, string ActionType, string Link)
- int **SetFormFieldSubmitActionEx** (int Index, string ActionType, string Link, int Flags)
- int **SetFormFieldTabOrder** (int Index, int Order)

- int **SetFormFieldTextFlags** (int Index, int Multiline, int Password, int FileSelect, int DoNotSpellCheck, int DoNotScroll)
- int **SetFormFieldTextSize** (int Index, double NewTextSize)
- int **SetFormFieldValue** (int Index, string NewValue)
- int **SetFormFieldValueByTitle** (string Title, string NewValue)
- int **SetFormFieldVisible** (int Index, int Visible)
- int **SetGDIPlusFileName** (string DLLFileName)
- int **SetGDIPlusOptions** (int OptionID, int NewValue)
- int **SetHTMLBoldFont** (string FontSet, int FontID)
- int **SetHTMLBoldItalicFont** (string FontSet, int FontID)
- int **SetHTMLItalicFont** (string FontSet, int FontID)
- int **SetHTMLNormalFont** (string FontSet, int FontID)
- int **SetImageAsMask** (int MaskType)
- int **SetImageMask** (double FromRed, double FromGreen, double FromBlue, double ToRed, double ToGreen, double ToBlue)
- int **SetImageMaskCMYK** (double FromC, double FromM, double FromY, double FromK, double ToC, double ToM, double ToY, double ToK)
- int **SetImageMaskFromImage** (int ImageID)
- int **SetImageOptional** (int OptionalContentGroupID)
- int **SetImageResolution** (int Horizontal, int Vertical, int Units)
- int **SetInformation** (int Key, string NewValue)
- int **SetJPEGQuality** (int Quality)
- int **SetJavaScriptMode** (int JSMode)
- int **SetKerning** (string CharPair, int Adjustment)
- int **SetLineCap** (int LineCap)
- int **SetLineColor** (double Red, double Green, double Blue)
- int **SetLineColorCMYK** (double C, double M, double Y, double K)
- int **SetLineColorSep** (string ColorName, double Tint)
- int **SetLineDash** (double DashOn, double DashOff)
- int **SetLineDashEx** (string DashValues)
- int **SetLineJoin** (int LineJoin)
- int **SetLineShader** (string ShaderName)
- int **SetLineWidth** (double LineWidth)
- int **SetMarkupAnnotStyle** (int Index, double Red, double Green, double Blue, double Transparency)
- int **SetMeasureDictBoundsCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictBoundsItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictCoordinateSystem** (int MeasureDictID, int CoordinateSystemID)
- int **SetMeasureDictGPTSCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictGPTSItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictLPTSCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictLPTSItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictPDU** (int MeasureDictID, int LinearUnit, int AreaUnit, int AngularUnit)
- int **SetMeasurementUnits** (int MeasurementUnits)
- int **SetNeedAppearances** (int NewValue)
- int **SetObjectFromString** (int ObjectNumber, byte[ ] Source)
- int **SetOpenActionDestination** (int OpenPage, int Zoom)
- int **SetOpenActionDestinationFull** (int OpenPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetOpenActionJavaScript** (string JavaScript)
- int **SetOpenActionMenu** (string MenuItem)
- int **SetOptionalContentConfigLocked** (int OptionalContentConfigID, int OptionalContentGroupID, int New←Locked)
- int **SetOptionalContentConfigState** (int OptionalContentConfigID, int OptionalContentGroupID, int New←State)
- int **SetOptionalContentGroupName** (int OptionalContentGroupID, string NewGroupName)

- int **SetOptionalContentGroupPrintable** (int OptionalContentGroupID, int Printable)
- int **SetOptionalContentGroupVisible** (int OptionalContentGroupID, int Visible)
- int **SetOrigin** (int Origin)
- int **SetOutlineColor** (int OutlineID, double Red, double Green, double Blue)
- int **SetOutlineDestination** (int OutlineID, int DestPage, double DestPosition)
- int **SetOutlineDestinationFull** (int OutlineID, int DestPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetOutlineDestinationZoom** (int OutlineID, int DestPage, double DestPosition, int Zoom)
- int **SetOutlineJavaScript** (int OutlineID, string JavaScript)
- int **SetOutlineNamedDestination** (int OutlineID, string DestName)
- int **SetOutlineOpenFile** (int OutlineID, string FileName)
- int **SetOutlineRemoteDestination** (int OutlineID, string FileName, int OpenPage, int Zoom, int DestType, double PntLeft, double PntTop, double PntRight, double PntBottom, int NewWindow)
- int **SetOutlineStyle** (int OutlineID, int SetItalic, int SetBold)
- int **SetOutlineTitle** (int OutlineID, string NewTitle)
- int **SetOutlineWebLink** (int OutlineID, string Link)
- int **SetOverprint** (int StrokingOverprint, int OtherOverprint, int OverprintMode)
- int **SetPDFAMode** (int NewMode)
- int **SetPNGTransparencyColor** (int RedByte, int GreenByte, int BlueByte)
- int **SetPageActionMenu** (string MenuItem)
- int **SetPageBox** (int BoxType, double Left, double Top, double Width, double Height)
- int **SetPageContentFromString** (byte[ ] Source)
- int **SetPageDimensions** (double NewPageWidth, double NewPageHeight)
- int **SetPageLayout** (int NewPageLayout)
- int **SetPageMode** (int NewPageMode)
- int **SetPageSize** (string PaperName)
- int **SetPageThumbnail** ()
- int **SetPageTransparencyGroup** (int CS, int Isolate, int Knockout)
- int **SetPrecision** (int NewPrecision)
- int **SetPrinterDevModeFromString** (byte[ ] Source)
- int **SetRenderCropType** (int NewCropType)
- int **SetRenderDCErasePage** (int NewErasePage)
- int **SetRenderDCOffset** (int NewOffsetX, int NewOffsetY)
- int **SetRenderOptions** (int OptionID, int NewValue)
- int **SetRenderScale** (double NewScale)
- int **SetScale** (double NewScale)
- int **SetSignProcessField** (int SignProcessID, string SignatureFieldName)
- int **SetSignProcessFieldBounds** (int SignProcessID, double Left, double Top, double Width, double Height)
- int **SetSignProcessFieldImageFromFile** (int SignProcessID, string ImageFileName, int Options)
- int **SetSignProcessFieldPage** (int SignProcessID, int SignaturePage)
- int **SetSignProcessInfo** (int SignProcessID, string Reason, string Location, string ContactInfo)
- int **SetSignProcessPFXFromFile** (int SignProcessID, string PFXFileName, string PFXPassword)
- int **SetSignProcessPassthrough** (int SignProcessID, int SignatureLength)
- int **SetSignProcessSubFilter** (int SignProcessID, int SubFilter)
- int **SetTableBorderColor** (int TableID, int BorderIndex, double Red, double Green, double Blue)
- int **SetTableBorderColorCMYK** (int TableID, int BorderIndex, double C, double M, double Y, double K)
- int **SetTableBorderWidth** (int TableID, int BorderIndex, double NewWidth)
- int **SetTableCellAlignment** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int New↵ CellAlignment)
- int **SetTableCellBackgroundColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double Red, double Green, double Blue)
- int **SetTableCellBackgroundColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int Last↵ Column, double C, double M, double Y, double K)
- int **SetTableCellBorderColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double Red, double Green, double Blue)

- int **SetTableCellBorderColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double C, double M, double Y, double K)
- int **SetTableCellBorderWidth** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double NewWidth)
- int **SetTableCellContent** (int TableID, int RowNumber, int ColumnNumber, string HTMLText)
- int **SetTableCellPadding** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int Border↩Index, double NewPadding)
- int **SetTableCellTextColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double Red, double Green, double Blue)
- int **SetTableCellTextColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double C, double M, double Y, double K)
- int **SetTableCellTextSize** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double NewTextSize)
- int **SetTableColumnWidth** (int TableID, int FirstColumn, int LastColumn, double NewWidth)
- int **SetTableRowHeight** (int TableID, int FirstRow, int LastRow, double NewHeight)
- int **SetTableThinBorders** (int TableID, int ThinBorders, double Red, double Green, double Blue)
- int **SetTableThinBordersCMYK** (int TableID, int ThinBorders, double C, double M, double Y, double K)
- int **SetTempFile** (string FileName)
- int **SetTempPath** (string NewPath)
- int **SetTextAlign** (int TextAlign)
- int **SetTextCharSpacing** (double CharSpacing)
- int **SetTextColor** (double Red, double Green, double Blue)
- int **SetTextColorCMYK** (double C, double M, double Y, double K)
- int **SetTextColorSep** (string ColorName, double Tint)
- int **SetTextExtractionArea** (double Left, double Top, double Width, double Height)
- int **SetTextExtractionOptions** (int OptionID, int NewValue)
- int **SetTextExtractionScaling** (int Options, double Horizontal, double Vertical)
- int **SetTextExtractionWordGap** (double NewWordGap)
- int **SetTextHighlight** (int Highlight)
- int **SetTextHighlightColor** (double Red, double Green, double Blue)
- int **SetTextHighlightColorCMYK** (double C, double M, double Y, double K)
- int **SetTextHighlightColorSep** (string ColorName, double Tint)
- int **SetTextMode** (int TextMode)
- int **SetTextRise** (double Rise)
- int **SetTextScaling** (double ScalePercentage)
- int **SetTextShader** (string ShaderName)
- int **SetTextSize** (double TextSize)
- int **SetTextSpacing** (double Spacing)
- int **SetTextUnderline** (int Underline)
- int **SetTextUnderlineColor** (double Red, double Green, double Blue)
- int **SetTextUnderlineColorCMYK** (double C, double M, double Y, double K)
- int **SetTextUnderlineColorSep** (string ColorName, double Tint)
- int **SetTextUnderlineCustomDash** (string DashPattern, double DashPhase)
- int **SetTextUnderlineDash** (double DashOn, double DashOff)
- int **SetTextUnderlineDistance** (double UnderlineDistance)
- int **SetTextUnderlineWidth** (double UnderlineWidth)
- int **SetTextWordSpacing** (double WordSpacing)
- int **SetTransparency** (int Transparency)
- int **SetViewerPreferences** (int Option, int NewValue)
- int **SetXFAFormFieldAccess** (string XFAFieldName, int NewAccess)
- int **SetXFAFormFieldBorderColor** (string XFAFieldName, double Red, double Green, double Blue)
- int **SetXFAFormFieldBorderPresence** (string XFAFieldName, int NewPresence)
- int **SetXFAFormFieldBorderWidth** (string XFAFieldName, double BorderWidth)
- int **SetXFAFormFieldValue** (string XFAFieldName, string NewValue)
- int **SetXFAFromString** (byte[ ] Source, int Options)

- int **SetupCustomPrinter** (string CustomPrinterName, int Setting, int NewValue)
- int **SignFile** (string InputFileName, string OpenPassword, string SignatureFieldName, string OutputFileName, string PFXFileName, string PFXPassword, string Reason, string Location, string ContactInfo)
- int **SplitPageText** (int Options)
- int **StartPath** (double StartX, double StartY)
- int **StoreCustomDataFromFile** (string Key, string FileName, int Location, int Options)
- int **StoreCustomDataFromString** (string Key, byte[ ] NewValue, int Location, int Options)
- int **StringResultLength** ()
- int **TestTempPath** ()
- int **TransformFile** (string InputFileName, string Password, string OutputFileName, int TransformType, int Options)
- int **UnlockKey** (string LicenseKey)
- int **Unlocked** ()
- int **UpdateAndFlattenFormField** (int Index)
- int **UpdateAppearanceStream** (int Index)
- int **UseKerning** (int Kern)
- int **UseUnsafeContentStreams** (int SafetyLevel)
- PDFLibrary ()

  Constructor modified by e-Docs USA to initialize the licence key.
- **PDFLibrary** (string dllFileName)
- bool **LibraryLoaded** ()
- void **ReleaseLibrary** ()
- int **AddArcToPath** (double CenterX, double CenterY, double TotalAngle)
- int **AddBoxToPath** (double Left, double Top, double Width, double Height)
- int **AddCJKFont** (int CJKFontID)
- int **AddCurveToPath** (double CtAX, double CtAY, double CtBX, double CtBY, double EndX, double EndY)
- int **AddEmbeddedFile** (string FileName, string MIMEType)
- int **AddFileAttachment** (string Title, int EmbeddedFileID)
- int **AddFormFieldChoiceSub** (int Index, string SubName, string DisplayName)
- int **AddFormFieldSub** (int Index, string SubName)
- int **AddFormFont** (int FontID)
- int **AddGlobalJavaScript** (string PackageName, string JavaScript)
- int **AddImageFromFile** (string FileName, int Options)
- int **AddImageFromFileOffset** (string FileName, int Offset, int DataLength, int Options)
- int **AddImageFromString** (byte[ ] Source, int Options)
- int **AddLGIDictToPage** (string DictContent)
- int **AddLineToPath** (double EndX, double EndY)
- int **AddLinkToDestination** (double Left, double Top, double Width, double Height, int DestID, int Options)
- int **AddLinkToEmbeddedFile** (double Left, double Top, double Width, double Height, int EmbeddedFileID, string Title, string Contents, int IconType, int Transpareny)
- int **AddLinkToFile** (double Left, double Top, double Width, double Height, string FileName, int Page, double Position, int NewWindow, int Options)
- int **AddLinkToJavaScript** (double Left, double Top, double Width, double Height, string JavaScript, int Options)
- int **AddLinkToLocalFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AddLinkToPage** (double Left, double Top, double Width, double Height, int Page, double Position, int Options)
- int **AddLinkToWeb** (double Left, double Top, double Width, double Height, string Link, int Options)
- int **AddNoteAnnotation** (double Left, double Top, int AnnotType, double PopupLeft, double PopupTop, double PopupWidth, double PopupHeight, string Title, string Contents, double Red, double Green, double Blue, int Open)
- int **AddOpenTypeFontFromFile** (string FileName, int Options)
- int **AddPageLabels** (int Start, int Style, int Offset, string Prefix)
- int **AddSVGAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, int Options)

- int **AddSWFAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, string Title, int Options)
- int **AddSeparationColor** (string ColorName, double C, double M, double Y, double K, int Options)
- int **AddStampAnnotation** (double Left, double Top, double Width, double Height, int StampType, string Title, string Contents, double Red, double Green, double Blue, int Options)
- int **AddStandardFont** (int StandardFontID)
- int **AddSubsettedFont** (string FontName, int CharsetIndex, string SubsetChars)
- int **AddTextMarkupAnnotation** (int MarkupType, double Left, double Top, double Width, double Height)
- int **AddToFileList** (string ListName, string FileName)
- int **AddTrueTypeFont** (string FontName, int Embed)
- int **AddTrueTypeFontFromFile** (string FileName)
- int **AddTrueTypeSubsettedFont** (string FontName, string SubsetChars, int Options)
- int **AddType1Font** (string FileName)
- int **AddU3DAnnotationFromFile** (double Left, double Top, double Width, double Height, string FileName, int Options)
- int **AnalyseFile** (string InputFileName, string Password)
- int **AnnotationCount** ()
- int **AnsiStringResultLength** ()
- int **AppendSpace** (double RelativeSpace)
- int **AppendTableColumns** (int TableID, int NewColumnCount)
- int **AppendTableRows** (int TableID, int NewRowCount)
- int **AppendText** (string Text)
- int **ApplyStyle** (string StyleName)
- int **AttachAnnotToForm** (int Index)
- int **BalanceContentStream** ()
- int **BalancePageTree** (int Options)
- int **BeginPageUpdate** ()
- int **CapturePage** (int Page)
- int **CapturePageEx** (int Page, int Options)
- int **CharWidth** (int CharCode)
- int **CheckFileCompliance** (string InputFileName, string Password, int ComplianceTest, int Options)
- int **CheckObjects** ()
- int **CheckPageAnnots** ()
- int **ClearFileList** (string ListName)
- int **ClearImage** (int ImageID)
- int **ClearPageLabels** ()
- int **ClearTextFormatting** ()
- int **CloneOutlineAction** (int OutlineID)
- int **ClonePages** (int StartPage, int EndPage, int RepeatCount)
- int **CloseOutline** (int OutlineID)
- int **ClosePath** ()
- int **CombineContentStreams** ()
- int **CompareOutlines** (int FirstOutlineID, int SecondOutlineID)
- int **CompressContent** ()
- int **CompressFonts** (int Compress)
- int **CompressImages** (int Compress)
- int **CompressPage** ()
- int **ContentStreamCount** ()
- int **ContentStreamSafe** ()
- int **CopyPageRanges** (int DocumentID, string RangeList)
- int **CopyPageRangesEx** (int DocumentID, string RangeList, int Options)
- int **CreateNewObject** ()
- int **CreateTable** (int RowCount, int ColumnCount)
- int **DAAppendFile** (int FileHandle)
- int **DACapturePage** (int FileHandle, int PageRef)

- int **DACapturePageEx** (int FileHandle, int PageRef, int Options)
- int **DACloseFile** (int FileHandle)
- int **DADrawCapturedPage** (int FileHandle, int DACaptureID, int DestPageRef, double PntLeft, double Pnt↩ Bottom, double PntWidth, double PntHeight)
- int **DADrawRotatedCapturedPage** (int FileHandle, int DACaptureID, int DestPageRef, double PntLeft, double PntBottom, double PntWidth, double PntHeight, double Angle)
- int **DAEmbedFileStreams** (int FileHandle, string RootPath)
- string **DAExtractPageText** (int FileHandle, int PageRef, int Options)
- int **DAExtractPageTextBlocks** (int FileHandle, int PageRef, int ExtractOptions)
- int **DAFindPage** (int FileHandle, int Page)
- int **DAGetAnnotationCount** (int FileHandle, int PageRef)
- int **DAGetFormFieldCount** (int FileHandle)
- string **DAGetFormFieldTitle** (int FileHandle, int FieldIndex)
- string **DAGetFormFieldValue** (int FileHandle, int FieldIndex)
- byte[ ] **DAGetImageDataToString** (int FileHandle, int ImageListID, int ImageIndex)
- double **DAGetImageDblProperty** (int FileHandle, int ImageListID, int ImageIndex, int PropertyID)
- int **DAGetImageIntProperty** (int FileHandle, int ImageListID, int ImageIndex, int PropertyID)
- int **DAGetImageListCount** (int FileHandle, int ImageListID)
- string **DAGetInformation** (int FileHandle, string Key)
- int **DAGetObjectCount** (int FileHandle)
- byte[ ] **DAGetObjectToString** (int FileHandle, int ObjectNumber)
- double **DAGetPageBox** (int FileHandle, int PageRef, int BoxIndex, int Dimension)
- byte[ ] **DAGetPageContentToString** (int FileHandle, int PageRef)
- int **DAGetPageCount** (int FileHandle)
- double **DAGetPageHeight** (int FileHandle, int PageRef)
- int **DAGetPageImageList** (int FileHandle, int PageRef)
- double **DAGetPageWidth** (int FileHandle, int PageRef)
- double **DAGetTextBlockBound** (int TextBlockListID, int Index, int BoundIndex)
- double **DAGetTextBlockCharWidth** (int TextBlockListID, int Index, int CharIndex)
- double **DAGetTextBlockColor** (int TextBlockListID, int Index, int ColorComponent)
- int **DAGetTextBlockColorType** (int TextBlockListID, int Index)
- int **DAGetTextBlockCount** (int TextBlockListID)
- string **DAGetTextBlockFontName** (int TextBlockListID, int Index)
- double **DAGetTextBlockFontSize** (int TextBlockListID, int Index)
- string **DAGetTextBlockText** (int TextBlockListID, int Index)
- int **DAHasPageBox** (int FileHandle, int PageRef, int BoxIndex)
- int **DAHidePage** (int FileHandle, int PageRef)
- int **DAMovePage** (int FileHandle, int PageRef, int TargetPageRef, int Options)
- int **DANewPage** (int FileHandle)
- int **DANewPages** (int FileHandle, int PageCount)
- int **DAOpenFile** (string InputFileName, string Password)
- int **DAOpenFileReadOnly** (string InputFileName, string Password)
- int **DAPageRotation** (int FileHandle, int PageRef)
- int **DAReleaseImageList** (int FileHandle, int ImageListID)
- int **DAReleaseTextBlocks** (int TextBlockListID)
- int **DARemoveUsageRights** (int FileHandle)
- int **DARenderPageToDC** (int FileHandle, int PageRef, int DPI, IntPtr DC)
- int **DARenderPageToFile** (int FileHandle, int PageRef, int Options, int DPI, string FileName)
- byte[ ] **DARenderPageToString** (int FileHandle, int PageRef, int Options, int DPI)
- int **DARotatePage** (int FileHandle, int PageRef, int Angle, int Options)
- int **DASaveAsFile** (int FileHandle, string OutputFileName)
- int **DASaveImageDataToFile** (int FileHandle, int ImageListID, int ImageIndex, string ImageFileName)
- int **DASetInformation** (int FileHandle, string Key, string NewValue)
- int **DASetPageBox** (int FileHandle, int PageRef, int BoxIndex, double X1, double Y1, double X2, double Y2)
- int **DASetPageSize** (int FileHandle, int PageRef, double PntWidth, double PntHeight)

- int **DASetTextExtractionArea** (double Left, double Top, double Width, double Height)
- int **DASetTextExtractionOptions** (int OptionID, int NewValue)
- int **DASetTextExtractionScaling** (int Options, double Horizontal, double Vertical)
- int **DASetTextExtractionWordGap** (double NewWordGap)
- int **DAShiftedHeader** (int FileHandle)
- int **Decrypt** ()
- int **DecryptFile** (string InputFileName, string OutputFileName, string Password)
- int **DeleteAnalysis** (int AnalysisID)
- int **DeleteAnnotation** (int Index)
- int **DeleteContentStream** ()
- int **DeleteFormField** (int Index)
- int **DeleteOptionalContentGroup** (int OptionalContentGroupID)
- int **DeletePageLGIDict** (int DictIndex)
- int **DeletePages** (int StartPage, int PageCount)
- int **DocJavaScriptAction** (string ActionType, string JavaScript)
- int **DocumentCount** ()
- int **DrawArc** (double XPos, double YPos, double Radius, double StartAngle, double EndAngle, int Pie, int DrawOptions)
- int **DrawBarcode** (double Left, double Top, double Width, double Height, string Text, int Barcode, int Options)
- int **DrawBox** (double Left, double Top, double Width, double Height, int DrawOptions)
- int **DrawCapturedPage** (int CaptureID, double Left, double Top, double Width, double Height)
- int **DrawCapturedPageMatrix** (int CaptureID, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawCircle** (double XPos, double YPos, double Radius, int DrawOptions)
- int **DrawDataMatrixSymbol** (double Left, double Top, double ModuleSize, string Text, int Encoding, int SymbolSize, int Options)
- int **DrawEllipse** (double XPos, double YPos, double Width, double Height, int DrawOptions)
- int **DrawEllipticArc** (double XPos, double YPos, double Width, double Height, double StartAngle, double EndAngle, int Pie, int DrawOptions)
- int **DrawHTMLText** (double Left, double Top, double Width, string HTMLText)
- string **DrawHTMLTextBox** (double Left, double Top, double Width, double Height, string HTMLText)
- string **DrawHTMLTextBoxMatrix** (double Width, double Height, string HTMLText, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawImage** (double Left, double Top, double Width, double Height)
- int **DrawImageMatrix** (double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawIntelligentMailBarcode** (double Left, double Top, double BarWidth, double FullBarHeight, double TrackerHeight, double SpaceWidth, string BarcodeData, int Options)
- int **DrawLine** (double StartX, double StartY, double EndX, double EndY)
- int **DrawMultiLineText** (double XPos, double YPos, string Delimiter, string Text)
- int **DrawPDF417Symbol** (double Left, double Top, string Text, int Options)
- int **DrawPDF417SymbolEx** (double Left, double Top, string Text, int Options, int FixedColumns, int Fixed↩ Rows, int ErrorLevel, double ModuleSize, double HeightWidthRatio)
- int **DrawPath** (int PathOptions)
- int **DrawPathEvenOdd** (int PathOptions)
- int **DrawPostScriptXObject** (int PSRef)
- int **DrawRotatedBox** (double Left, double Bottom, double Width, double Height, double Angle, int Draw↩ Options)
- int **DrawRotatedCapturedPage** (int CaptureID, double Left, double Bottom, double Width, double Height, double Angle)
- int **DrawRotatedImage** (double Left, double Bottom, double Width, double Height, double Angle)
- int **DrawRotatedMultiLineText** (double XPos, double YPos, double Angle, string Delimiter, string Text)
- int **DrawRotatedText** (double XPos, double YPos, double Angle, string Text)
- int **DrawRotatedTextBox** (double Left, double Top, double Width, double Height, double Angle, string Text, int Options)

- int **DrawRoundedBox** (double Left, double Top, double Width, double Height, double Radius, int Draw↵ Options)
- int **DrawScaledImage** (double Left, double Top, double Scale)
- int **DrawSpacedText** (double XPos, double YPos, double Spacing, string Text)
- double **DrawTableRows** (int TableID, double Left, double Top, double Height, int FirstRow, int LastRow)
- int **DrawText** (double XPos, double YPos, string Text)
- int **DrawTextArc** (double XPos, double YPos, double Radius, double Angle, string Text, int DrawOptions)
- int **DrawTextBox** (double Left, double Top, double Width, double Height, string Text, int Options)
- int **DrawTextBoxMatrix** (double Width, double Height, string Text, int Options, double M11, double M12, double M21, double M22, double MDX, double MDY)
- int **DrawWrappedText** (double XPos, double YPos, double Width, string Text)
- int **EditableContentStream** ()
- int **EmbedFile** (string Title, string FileName, string MIMEType)
- int **EmbeddedFileCount** ()
- int **EncapsulateContentStream** ()
- int **EncodePermissions** (int CanPrint, int CanCopy, int CanChange, int CanAddNotes, int CanFillFields, int CanCopyAccess, int CanAssemble, int CanPrintFull)
- int **Encrypt** (string Owner, string User, int Strength, int Permissions)
- int **EncryptFile** (string InputFileName, string OutputFileName, string Owner, string User, int Strength, int Permissions)
- int **EncryptWithFingerprint** (string Fingerprint)
- int **EncryptionAlgorithm** ()
- int **EncryptionStatus** ()
- int **EncryptionStrength** ()
- int **EndPageUpdate** ()
- int **EndSignProcessToFile** (int SignProcessID, string OutputFile)
- byte[ ] **EndSignProcessToString** (int SignProcessID)
- byte[ ] **ExtractFilePageContentToString** (string InputFileName, string Password, int Page)
- string **ExtractFilePageText** (string InputFileName, string Password, int Page, int Options)
- int **ExtractFilePageTextBlocks** (string InputFileName, string Password, int Page, int Options)
- int **ExtractFilePages** (string InputFileName, string Password, string OutputFileName, string RangeList)
- int **ExtractFilePagesEx** (string InputFileName, string Password, string OutputFileName, string RangeList, int Options)
- int **ExtractPageRanges** (string RangeList)
- int **ExtractPageTextBlocks** (int ExtractOptions)
- int **ExtractPages** (int StartPage, int PageCount)
- int **FileListCount** (string ListName)
- string **FileListItem** (string ListName, int Index)
- int **FindFonts** ()
- int **FindFormFieldByTitle** (string Title)
- int **FindImages** ()
- int **FitImage** (double Left, double Top, double Width, double Height, int HAlign, int VAlign, int Rotate)
- int **FitRotatedTextBox** (double Left, double Top, double Width, double Height, double Angle, string Text, int Options)
- int **FitTextBox** (double Left, double Top, double Width, double Height, string Text, int Options)
- int **FlattenAnnot** (int Index, int Options)
- int **FlattenFormField** (int Index)
- int **FontCount** ()
- string **FontFamily** ()
- int **FontHasKerning** ()
- string **FontName** ()
- string **FontReference** ()
- int **FontSize** ()
- int **FontType** ()
- int **FormFieldCount** ()

- int **FormFieldHasParent** (int Index)
- int **FormFieldJavaScriptAction** (int Index, string ActionType, string JavaScript)
- int **FormFieldWebLinkAction** (int Index, string ActionType, string Link)
- int **GetActionDest** (int ActionID)
- string **GetActionType** (int ActionID)
- string **GetActionURL** (int ActionID)
- string **GetAnalysisInfo** (int AnalysisID, int AnalysisItem)
- int **GetAnnotActionID** (int Index)
- double **GetAnnotDblProperty** (int Index, int Tag)
- int **GetAnnotDest** (int Index)
- int **GetAnnotIntProperty** (int Index, int Tag)
- int **GetAnnotQuadCount** (int Index)
- double **GetAnnotQuadPoints** (int Index, int QuadNumber, int PointNumber)
- int **GetAnnotSoundToFile** (int Index, int Options, string SoundFileName)
- byte[ ] **GetAnnotSoundToString** (int Index, int Options)
- string **GetAnnotStrProperty** (int Index, int Tag)
- double **GetBarcodeWidth** (double NominalWidth, string Text, int Barcode)
- string **GetBaseURL** ()
- int **GetCSDictEPSG** (int CSDictID)
- int **GetCSDictType** (int CSDictID)
- string **GetCSDictWKT** (int CSDictID)
- IntPtr **GetCanvasDC** (int CanvasWidth, int CanvasHeight)
- string **GetCatalogInformation** (string Key)
- byte[ ] **GetContentStreamToString** ()
- string **GetCustomInformation** (string Key)
- string **GetCustomKeys** (int Location)
- string **GetDefaultPrinterName** ()
- string **GetDestName** (int DestID)
- int **GetDestPage** (int DestID)
- int **GetDestType** (int DestID)
- double **GetDestValue** (int DestID, int ValueKey)
- string **GetDocJavaScript** (string ActionType)
- string **GetDocumentFileName** ()
- int **GetDocumentFileSize** ()
- int **GetDocumentID** (int Index)
- string **GetDocumentIdentifier** (int Part, int Options)
- string **GetDocumentMetadata** ()
- int **GetDocumentRepaired** ()
- string **GetDocumentResourceList** ()
- int **GetEmbeddedFileContentToFile** (int Index, string FileName)
- byte[ ] **GetEmbeddedFileContentToString** (int Index)
- int **GetEmbeddedFileID** (int Index)
- int **GetEmbeddedFileIntProperty** (int Index, int Tag)
- string **GetEmbeddedFileStrProperty** (int Index, int Tag)
- string **GetEncryptionFingerprint** ()
- string **GetFileMetadata** (string InputFileName, string Password)
- int **GetFirstChildOutline** (int OutlineID)
- int **GetFirstOutline** ()
- int **GetFontEncoding** ()
- int **GetFontFlags** (int FontFlagItemID)
- int **GetFontID** (int Index)
- int **GetFontObjectNumber** ()
- int **GetFormFieldActionID** (int Index, string TriggerEvent)
- int **GetFormFieldAlignment** (int Index)
- int **GetFormFieldAnnotFlags** (int Index)

- double **GetFormFieldBackgroundColor** (int Index, int ColorComponent)
- int **GetFormFieldBackgroundColorType** (int Index)
- double **GetFormFieldBorderColor** (int Index, int ColorComponent)
- int **GetFormFieldBorderColorType** (int Index)
- double **GetFormFieldBorderProperty** (int Index, int PropKey)
- int **GetFormFieldBorderStyle** (int Index)
- double **GetFormFieldBound** (int Index, int Edge)
- string **GetFormFieldCaption** (int Index)
- int **GetFormFieldCheckStyle** (int Index)
- string **GetFormFieldChildTitle** (int Index)
- int **GetFormFieldChoiceType** (int Index)
- double **GetFormFieldColor** (int Index, int ColorComponent)
- int **GetFormFieldComb** (int Index)
- string **GetFormFieldDefaultValue** (int Index)
- string **GetFormFieldDescription** (int Index)
- int **GetFormFieldFlags** (int Index)
- string **GetFormFieldFontName** (int Index)
- string **GetFormFieldJavaScript** (int Index, string ActionType)
- int **GetFormFieldKidCount** (int Index)
- int **GetFormFieldMaxLen** (int Index)
- int **GetFormFieldNoExport** (int Index)
- int **GetFormFieldPage** (int Index)
- int **GetFormFieldPrintable** (int Index)
- int **GetFormFieldReadOnly** (int Index)
- int **GetFormFieldRequired** (int Index)
- string **GetFormFieldRichTextString** (int Index, string Key)
- int **GetFormFieldRotation** (int Index)
- int **GetFormFieldSubCount** (int Index)
- string **GetFormFieldSubDisplayName** (int Index, int SubIndex)
- string **GetFormFieldSubName** (int Index, int SubIndex)
- int **GetFormFieldSubTempIndex** (int Index, int SubIndex)
- int **GetFormFieldTabOrder** (int Index)
- int **GetFormFieldTabOrderEx** (int Index, int Options)
- int **GetFormFieldTextFlags** (int Index, int ValueKey)
- double **GetFormFieldTextSize** (int Index)
- string **GetFormFieldTitle** (int Index)
- int **GetFormFieldType** (int Index)
- string **GetFormFieldValue** (int Index)
- string **GetFormFieldValueByTitle** (string Title)
- int **GetFormFieldVisible** (int Index)
- string **GetFormFieldWebLink** (int Index, string ActionType)
- int **GetFormFontCount** ()
- string **GetFormFontName** (int FontIndex)
- string **GetGlobalJavaScript** (string PackageName)
- double **GetHTMLTextHeight** (double Width, string HTMLText)
- int **GetHTMLTextLineCount** (double Width, string HTMLText)
- double **GetHTMLTextWidth** (double MaxWidth, string HTMLText)
- int **GetImageID** (int Index)
- int **GetImageListCount** (int ImageListID)
- byte[ ] **GetImageListItemDataToString** (int ImageListID, int ImageIndex, int Options)
- double **GetImageListItemDblProperty** (int ImageListID, int ImageIndex, int PropertyID)
- int **GetImageListItemIntProperty** (int ImageListID, int ImageIndex, int PropertyID)
- int **GetImageMeasureDict** ()
- int **GetImagePageCount** (string FileName)
- int **GetImagePageCountFromString** (byte[ ] Source)

- int **GetImagePtDataDict** ()
- string **GetInformation** (int Key)
- string **GetInstalledFontsByCharset** (int CharsetIndex, int Options)
- string **GetInstalledFontsByCodePage** (int CodePage, int Options)
- int **GetKerning** (string CharPair)
- string **GetLatestPrinterNames** ()
- int **GetMaxObjectNumber** ()
- int **GetMeasureDictBoundsCount** (int MeasureDictID)
- double **GetMeasureDictBoundsItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictCoordinateSystem** (int MeasureDictID)
- int **GetMeasureDictDCSDict** (int MeasureDictID)
- int **GetMeasureDictGCSDict** (int MeasureDictID)
- int **GetMeasureDictGPTSCount** (int MeasureDictID)
- double **GetMeasureDictGPTSItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictLPTSCount** (int MeasureDictID)
- double **GetMeasureDictLPTSItem** (int MeasureDictID, int ItemIndex)
- int **GetMeasureDictPDU** (int MeasureDictID, int UnitIndex)
- int **GetNamedDestination** (string DestName)
- int **GetNextOutline** (int OutlineID)
- int **GetObjectCount** ()
- int **GetObjectDecodeError** (int ObjectNumber)
- byte[ ] **GetObjectToString** (int ObjectNumber)
- int **GetOpenActionDestination** ()
- string **GetOpenActionJavaScript** ()
- int **GetOptionalContentConfigCount** ()
- int **GetOptionalContentConfigLocked** (int OptionalContentConfigID, int OptionalContentGroupID)
- int **GetOptionalContentConfigOrderCount** (int OptionalContentConfigID)
- int **GetOptionalContentConfigOrderItemID** (int OptionalContentConfigID, int ItemIndex)
- string **GetOptionalContentConfigOrderItemLabel** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigOrderItemLevel** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigOrderItemType** (int OptionalContentConfigID, int ItemIndex)
- int **GetOptionalContentConfigState** (int OptionalContentConfigID, int OptionalContentGroupID)
- int **GetOptionalContentGroupID** (int Index)
- string **GetOptionalContentGroupName** (int OptionalContentGroupID)
- int **GetOptionalContentGroupPrintable** (int OptionalContentGroupID)
- int **GetOptionalContentGroupVisible** (int OptionalContentGroupID)
- int **GetOrigin** ()
- int **GetOutlineActionID** (int OutlineID)
- double **GetOutlineColor** (int OutlineID, int ColorComponent)
- int **GetOutlineDest** (int OutlineID)
- int **GetOutlineID** (int Index)
- string **GetOutlineJavaScript** (int OutlineID)
- int **GetOutlineObjectNumber** (int OutlineID)
- string **GetOutlineOpenFile** (int OutlineID)
- int **GetOutlinePage** (int OutlineID)
- int **GetOutlineStyle** (int OutlineID)
- string **GetOutlineWebLink** (int OutlineID)
- double **GetPageBox** (int BoxType, int Dimension)
- string **GetPageColorSpaces** (int Options)
- byte[ ] **GetPageContentToString** ()
- int **GetPageImageList** (int Options)
- string **GetPageJavaScript** (string ActionType)
- string **GetPageLGIDictContent** (int DictIndex)
- int **GetPageLGIDictCount** ()
- string **GetPageLabel** (int Page)

- int **GetPageLayout** ()
- byte[ ] **GetPageMetricsToString** (int StartPage, int EndPage, int Options)
- int **GetPageMode** ()
- string **GetPageText** (int ExtractOptions)
- int **GetPageViewPortCount** ()
- int **GetPageViewPortID** (int Index)
- int **GetParentOutline** (int OutlineID)
- int **GetPrevOutline** (int OutlineID)
- string **GetPrinterBins** (string PrinterName)
- byte[ ] **GetPrinterDevModeToString** (string PrinterName)
- string **GetPrinterMediaTypes** (string PrinterName)
- string **GetPrinterNames** ()
- double **GetRenderScale** ()
- int **GetSignProcessByteRange** (int SignProcessID, int ArrayPosition)
- int **GetSignProcessResult** (int SignProcessID)
- int **GetStringListCount** (int StringListID)
- string **GetStringListItem** (int StringListID, int ItemIndex)
- double **GetTableCellDblProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- int **GetTableCellIntProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- string **GetTableCellStrProperty** (int TableID, int RowNumber, int ColumnNumber, int Tag)
- int **GetTableColumnCount** (int TableID)
- int **GetTableLastDrawnRow** (int TableID)
- int **GetTableRowCount** (int TableID)
- string **GetTempPath** ()
- double **GetTextAscent** ()
- double **GetTextBlockBound** (int TextBlockListID, int Index, int BoundIndex)
- double **GetTextBlockCharWidth** (int TextBlockListID, int Index, int CharIndex)
- double **GetTextBlockColor** (int TextBlockListID, int Index, int ColorComponent)
- int **GetTextBlockColorType** (int TextBlockListID, int Index)
- int **GetTextBlockCount** (int TextBlockListID)
- string **GetTextBlockFontName** (int TextBlockListID, int Index)
- double **GetTextBlockFontSize** (int TextBlockListID, int Index)
- string **GetTextBlockText** (int TextBlockListID, int Index)
- double **GetTextBound** (int Edge)
- double **GetTextDescent** ()
- double **GetTextHeight** ()
- double **GetTextSize** ()
- double **GetTextWidth** (string Text)
- string **GetUnicodeCharactersFromEncoding** (int Encoding)
- double **GetViewPortBBox** (int ViewPortID, int Dimension)
- int **GetViewPortMeasureDict** (int ViewPortID)
- string **GetViewPortName** (int ViewPortID)
- int **GetViewPortPtDataDict** (int ViewPortID)
- int **GetViewerPreferences** (int Option)
- string **GetWrappedText** (double Width, string Delimiter, string Text)
- string **GetWrappedTextBreakString** (double Width, string Delimiter, string Text)
- double **GetWrappedTextHeight** (double Width, string Text)
- int **GetWrappedTextLineCount** (double Width, string Text)
- int **GetXFAFormFieldCount** ()
- string **GetXFAFormFieldName** (int Index)
- string **GetXFAFormFieldNames** (string Delimiter)
- string **GetXFAFormFieldValue** (string XFAFieldName)
- byte[ ] **GetXFAToString** (int Options)
- int **GlobalJavaScriptCount** ()
- string **GlobalJavaScriptPackageName** (int Index)

- int **HasFontResources** ()
- int **HasPageBox** (int BoxType)
- int **HidePage** ()
- int **ImageCount** ()
- int **ImageFillColor** ()
- int **ImageHeight** ()
- int **ImageHorizontalResolution** ()
- int **ImageResolutionUnits** ()
- int **ImageType** ()
- int **ImageVerticalResolution** ()
- int **ImageWidth** ()
- int **ImportEMFFromFile** (string FileName, int FontOptions, int GeneralOptions)
- int **InsertPages** (int StartPage, int PageCount)
- int **InsertTableColumns** (int TableID, int Position, int NewColumnCount)
- int **InsertTableRows** (int TableID, int Position, int NewRowCount)
- int **IsAnnotFormField** (int Index)
- int **IsTaggedPDF** ()
- int **LastErrorCode** ()
- string **LastRenderError** ()
- string **LibraryVersion** ()
- string **LicenseInfo** ()
- int **Linearized** ()
- int **LoadFromCanvasDC** (double DPI, int Options)
- int **LoadFromFile** (string FileName, string Password)
- int **LoadFromString** (byte[ ] Source, string Password)
- int **LoadState** ()
- int **MergeDocument** (int DocumentID)
- int **MergeFileList** (string ListName, string OutputFileName)
- int **MergeFileListFast** (string ListName, string OutputFileName)
- int **MergeFiles** (string FirstFileName, string SecondFileName, string OutputFileName)
- int **MergeTableCells** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn)
- int **MoveContentStream** (int FromPosition, int ToPosition)
- int **MoveOutlineAfter** (int OutlineID, int SiblingID)
- int **MoveOutlineBefore** (int OutlineID, int SiblingID)
- int **MovePage** (int NewPosition)
- int **MovePath** (double NewX, double NewY)
- int **MultiplyScale** (double MultScaleBy)
- int **NewChildFormField** (int Index, string Title, int FieldType)
- int **NewContentStream** ()
- string **NewCustomPrinter** (string OriginalPrinterName)
- int **NewDestination** (int DestPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **NewDocument** ()
- int **NewFormField** (string Title, int FieldType)
- int **NewNamedDestination** (string DestName, int DestID)
- int **NewOptionalContentGroup** (string GroupName)
- int **NewOutline** (int Parent, string Title, int DestPage, double DestPosition)
- int **NewPage** ()
- int **NewPageFromCanvasDC** (int DPI, int Options)
- int **NewPages** (int PageCount)
- int **NewPostScriptXObject** (string PS)
- int **NewRGBAxialShader** (string ShaderName, double StartX, double StartY, double StartRed, double Start↩ Green, double StartBlue, double EndX, double EndY, double EndRed, double EndGreen, double EndBlue, int Extend)
- int **NewSignProcessFromFile** (string InputFile, string Password)

- int **NewSignProcessFromString** (byte[ ] Source, string Password)
- int **NewStaticOutline** (int Parent, string Title)
- int **NewTilingPatternFromCapturedPage** (string PatternName, int CaptureID)
- int **NoEmbedFontListAdd** (string FontName)
- int **NoEmbedFontListCount** ()
- string **NoEmbedFontListGet** (int Index)
- int **NoEmbedFontListRemoveAll** ()
- int **NoEmbedFontListRemoveIndex** (int Index)
- int **NoEmbedFontListRemoveName** (string FontName)
- int **NormalizePage** (int NormalizeOptions)
- int **OpenOutline** (int OutlineID)
- int **OptionalContentGroupCount** ()
- int **OutlineCount** ()
- string **OutlineTitle** (int OutlineID)
- int **PageCount** ()
- double **PageHeight** ()
- int **PageJavaScriptAction** (string ActionType, string JavaScript)
- int **PageRotation** ()
- double **PageWidth** ()
- int **PrintDocument** (string PrinterName, int StartPage, int EndPage, int Options)
- int **PrintDocumentToFile** (string PrinterName, int StartPage, int EndPage, int Options, string FileName)
- int **PrintOptions** (int PageScaling, int AutoRotateCenter, string Title)
- int **PrintPages** (string PrinterName, string PageRanges, int Options)
- int **PrintPagesToFile** (string PrinterName, string PageRanges, int Options, string FileName)
- int **ReleaseImage** (int ImageID)
- int **ReleaseImageList** (int ImageListID)
- int **ReleaseSignProcess** (int SignProcessID)
- int **ReleaseStringList** (int StringListID)
- int **ReleaseTextBlocks** (int TextBlockListID)
- int **RemoveAppearanceStream** (int Index)
- int **RemoveCustomInformation** (string Key)
- int **RemoveDocument** (int DocumentID)
- int **RemoveEmbeddedFile** (int Index)
- int **RemoveFormFieldBackgroundColor** (int Index)
- int **RemoveFormFieldBorderColor** (int Index)
- int **RemoveGlobalJavaScript** (string PackageName)
- int **RemoveOpenAction** ()
- int **RemoveOutline** (int OutlineID)
- int **RemoveSharedContentStreams** ()
- int **RemoveStyle** (string StyleName)
- int **RemoveUsageRights** ()
- int **RemoveXFAEntries** (int Options)
- int **RenderDocumentToFile** (int DPI, int StartPage, int EndPage, int Options, string FileName)
- int **RenderPageToDC** (int DPI, int Page, IntPtr DC)
- int **RenderPageToFile** (int DPI, int Page, int Options, string FileName)
- byte[ ] **RenderPageToString** (int DPI, int Page, int Options)
- int **ReplaceFonts** ()
- int **ReplaceImage** (int OriginalImageID, int NewImageID)
- int **ReplaceTag** (string Tag, string NewValue)
- int **RequestPrinterStatus** (int StatusCommand)
- int **RetrieveCustomDataToFile** (string Key, string FileName, int Location)
- byte[ ] **RetrieveCustomDataToString** (string Key, int Location)
- int **ReverseImage** (int Reset)
- int **RotatePage** (int PageRotation)
- int **SaveFontToFile** (string FileName)

- int **SaveImageListItemDataToFile** (int ImageListID, int ImageIndex, int Options, string ImageFileName)
- int **SaveImageToFile** (string FileName)
- byte[ ] **SaveImageToString** ()
- int **SaveState** ()
- int **SaveStyle** (string StyleName)
- int **SaveToFile** (string FileName)
- byte[ ] **SaveToString** ()
- int **SecurityInfo** (int SecurityItem)
- int **SelectContentStream** (int NewIndex)
- int **SelectDocument** (int DocumentID)
- int **SelectFont** (int FontID)
- int **SelectImage** (int ImageID)
- int **SelectPage** (int PageNumber)
- int **SelectRenderer** (int RendererID)
- int **SelectedDocument** ()
- int **SelectedFont** ()
- int **SelectedImage** ()
- int **SelectedPage** ()
- int **SetActionURL** (int ActionID, string NewURL)
- int **SetAnnotBorderColor** (int Index, double Red, double Green, double Blue)
- int **SetAnnotBorderStyle** (int Index, double Width, int Style, double DashOn, double DashOff)
- int **SetAnnotContents** (int Index, string NewContents)
- int **SetAnnotDblProperty** (int Index, int Tag, double NewValue)
- int **SetAnnotIntProperty** (int Index, int Tag, int NewValue)
- int **SetAnnotQuadPoints** (int Index, int QuadNumber, double X1, double Y1, double X2, double Y2, double X3, double Y3, double X4, double Y4)
- int **SetAnnotRect** (int Index, double Left, double Top, double Width, double Height)
- int **SetAnnotStrProperty** (int Index, int Tag, string NewValue)
- int **SetAnsiMode** (int NewAnsiMode)
- int **SetBaseURL** (string NewBaseURL)
- int **SetBlendMode** (int BlendMode)
- int **SetBreakString** (string NewBreakString)
- int **SetCSDictEPSG** (int CSDictID, int NewEPSG)
- int **SetCSDictType** (int CSDictID, int NewDictType)
- int **SetCSDictWKT** (int CSDictID, string NewWKT)
- int **SetCairoFileName** (string FileName)
- int **SetCapturedPageOptional** (int CaptureID, int OptionalContentGroupID)
- int **SetCapturedPageTransparencyGroup** (int CaptureID, int CS, int Isolate, int Knockout)
- int **SetCatalogInformation** (string Key, string NewValue)
- int **SetCharWidth** (int CharCode, int NewWidth)
- int **SetClippingPath** ()
- int **SetClippingPathEvenOdd** ()
- int **SetCompatibility** (int CompatibilityItem, int CompatibilityMode)
- int **SetContentStreamFromString** (byte[ ] Source)
- int **SetContentStreamOptional** (int OptionalContentGroupID)
- int **SetCropBox** (double Left, double Top, double Width, double Height)
- int **SetCustomInformation** (string Key, string NewValue)
- int **SetCustomLineDash** (string DashPattern, double DashPhase)
- int **SetDestProperties** (int DestID, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetDestValue** (int DestID, int ValueKey, double NewValue)
- int **SetDocumentMetadata** (string XMP)
- int **SetEmbeddedFileStrProperty** (int Index, int Tag, string NewValue)
- int **SetFillColor** (double Red, double Green, double Blue)
- int **SetFillColorCMYK** (double C, double M, double Y, double K)

- int **SetFillColorSep** (string ColorName, double Tint)
- int **SetFillShader** (string ShaderName)
- int **SetFillTilingPattern** (string PatternName)
- int **SetFindImagesMode** (int NewFindImagesMode)
- int **SetFontEncoding** (int Encoding)
- int **SetFontFlags** (int Fixed, int Serif, int Symbolic, int Script, int Italic, int AllCap, int SmallCap, int ForceBold)
- int **SetFormFieldAlignment** (int Index, int Alignment)
- int **SetFormFieldAnnotFlags** (int Index, int NewFlags)
- int **SetFormFieldBackgroundColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldBackgroundColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldBackgroundColorGray** (int Index, double Gray)
- int **SetFormFieldBackgroundColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldBorderColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldBorderColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldBorderColorGray** (int Index, double Gray)
- int **SetFormFieldBorderColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldBorderStyle** (int Index, double Width, int Style, double DashOn, double DashOff)
- int **SetFormFieldBounds** (int Index, double Left, double Top, double Width, double Height)
- int **SetFormFieldCalcOrder** (int Index, int Order)
- int **SetFormFieldCaption** (int Index, string NewCaption)
- int **SetFormFieldCheckStyle** (int Index, int CheckStyle, int Position)
- int **SetFormFieldChildTitle** (int Index, string NewTitle)
- int **SetFormFieldChoiceType** (int Index, int ChoiceType)
- int **SetFormFieldColor** (int Index, double Red, double Green, double Blue)
- int **SetFormFieldColorCMYK** (int Index, double C, double M, double Y, double K)
- int **SetFormFieldColorSep** (int Index, string ColorName, double Tint)
- int **SetFormFieldComb** (int Index, int Comb)
- int **SetFormFieldDefaultValue** (int Index, string NewDefaultValue)
- int **SetFormFieldDescription** (int Index, string NewDescription)
- int **SetFormFieldFlags** (int Index, int NewFlags)
- int **SetFormFieldFont** (int Index, int FontIndex)
- int **SetFormFieldHighlightMode** (int Index, int NewMode)
- int **SetFormFieldIcon** (int Index, int IconType, int CaptureID)
- int **SetFormFieldIconStyle** (int Index, int Placement, int Scale, int ScaleType, int HorizontalShift, int Vertical↩ Shift)
- int **SetFormFieldMaxLen** (int Index, int NewMaxLen)
- int **SetFormFieldNoExport** (int Index, int NoExport)
- int **SetFormFieldOptional** (int Index, int OptionalContentGroupID)
- int **SetFormFieldPage** (int Index, int NewPage)
- int **SetFormFieldPrintable** (int Index, int Printable)
- int **SetFormFieldReadOnly** (int Index, int ReadOnly)
- int **SetFormFieldRequired** (int Index, int Required)
- int **SetFormFieldResetAction** (int Index, string ActionType)
- int **SetFormFieldRichTextString** (int Index, string Key, string NewValue)
- int **SetFormFieldRotation** (int Index, int Angle)
- int **SetFormFieldSignatureImage** (int Index, int ImageID, int Options)
- int **SetFormFieldStandardFont** (int Index, int StandardFontID)
- int **SetFormFieldSubmitAction** (int Index, string ActionType, string Link)
- int **SetFormFieldSubmitActionEx** (int Index, string ActionType, string Link, int Flags)
- int **SetFormFieldTabOrder** (int Index, int Order)
- int **SetFormFieldTextFlags** (int Index, int Multiline, int Password, int FileSelect, int DoNotSpellCheck, int DoNotScroll)
- int **SetFormFieldTextSize** (int Index, double NewTextSize)
- int **SetFormFieldValue** (int Index, string NewValue)
- int **SetFormFieldValueByTitle** (string Title, string NewValue)

- int **SetFormFieldVisible** (int Index, int Visible)
- int **SetGDIPlusFileName** (string DLLFileName)
- int **SetGDIPlusOptions** (int OptionID, int NewValue)
- int **SetHTMLBoldFont** (string FontSet, int FontID)
- int **SetHTMLBoldItalicFont** (string FontSet, int FontID)
- int **SetHTMLItalicFont** (string FontSet, int FontID)
- int **SetHTMLNormalFont** (string FontSet, int FontID)
- int **SetImageAsMask** (int MaskType)
- int **SetImageMask** (double FromRed, double FromGreen, double FromBlue, double ToRed, double ToGreen, double ToBlue)
- int **SetImageMaskCMYK** (double FromC, double FromM, double FromY, double FromK, double ToC, double ToM, double ToY, double ToK)
- int **SetImageMaskFromImage** (int ImageID)
- int **SetImageOptional** (int OptionalContentGroupID)
- int **SetImageResolution** (int Horizontal, int Vertical, int Units)
- int **SetInformation** (int Key, string NewValue)
- int **SetJPEGQuality** (int Quality)
- int **SetJavaScriptMode** (int JSMode)
- int **SetKerning** (string CharPair, int Adjustment)
- int **SetLineCap** (int LineCap)
- int **SetLineColor** (double Red, double Green, double Blue)
- int **SetLineColorCMYK** (double C, double M, double Y, double K)
- int **SetLineColorSep** (string ColorName, double Tint)
- int **SetLineDash** (double DashOn, double DashOff)
- int **SetLineDashEx** (string DashValues)
- int **SetLineJoin** (int LineJoin)
- int **SetLineShader** (string ShaderName)
- int **SetLineWidth** (double LineWidth)
- int **SetMarkupAnnotStyle** (int Index, double Red, double Green, double Blue, double Transparency)
- int **SetMeasureDictBoundsCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictBoundsItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictCoordinateSystem** (int MeasureDictID, int CoordinateSystemID)
- int **SetMeasureDictGPTSCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictGPTSItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictLPTSCount** (int MeasureDictID, int NewCount)
- int **SetMeasureDictLPTSItem** (int MeasureDictID, int ItemIndex, double NewValue)
- int **SetMeasureDictPDU** (int MeasureDictID, int LinearUnit, int AreaUnit, int AngularUnit)
- int **SetMeasurementUnits** (int MeasurementUnits)
- int **SetNeedAppearances** (int NewValue)
- int **SetObjectFromString** (int ObjectNumber, byte[ ] Source)
- int **SetOpenActionDestination** (int OpenPage, int Zoom)
- int **SetOpenActionDestinationFull** (int OpenPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetOpenActionJavaScript** (string JavaScript)
- int **SetOpenActionMenu** (string MenuItem)
- int **SetOptionalContentConfigLocked** (int OptionalContentConfigID, int OptionalContentGroupID, int New←Locked)
- int **SetOptionalContentConfigState** (int OptionalContentConfigID, int OptionalContentGroupID, int New←State)
- int **SetOptionalContentGroupName** (int OptionalContentGroupID, string NewGroupName)
- int **SetOptionalContentGroupPrintable** (int OptionalContentGroupID, int Printable)
- int **SetOptionalContentGroupVisible** (int OptionalContentGroupID, int Visible)
- int **SetOrigin** (int Origin)
- int **SetOutlineColor** (int OutlineID, double Red, double Green, double Blue)
- int **SetOutlineDestination** (int OutlineID, int DestPage, double DestPosition)

- int **SetOutlineDestinationFull** (int OutlineID, int DestPage, int Zoom, int DestType, double Left, double Top, double Right, double Bottom)
- int **SetOutlineDestinationZoom** (int OutlineID, int DestPage, double DestPosition, int Zoom)
- int **SetOutlineJavaScript** (int OutlineID, string JavaScript)
- int **SetOutlineNamedDestination** (int OutlineID, string DestName)
- int **SetOutlineOpenFile** (int OutlineID, string FileName)
- int **SetOutlineRemoteDestination** (int OutlineID, string FileName, int OpenPage, int Zoom, int DestType, double PntLeft, double PntTop, double PntRight, double PntBottom, int NewWindow)
- int **SetOutlineStyle** (int OutlineID, int SetItalic, int SetBold)
- int **SetOutlineTitle** (int OutlineID, string NewTitle)
- int **SetOutlineWebLink** (int OutlineID, string Link)
- int **SetOverprint** (int StrokingOverprint, int OtherOverprint, int OverprintMode)
- int **SetPDFAMode** (int NewMode)
- int **SetPNGTransparencyColor** (int RedByte, int GreenByte, int BlueByte)
- int **SetPageActionMenu** (string MenuItem)
- int **SetPageBox** (int BoxType, double Left, double Top, double Width, double Height)
- int **SetPageContentFromString** (byte[ ] Source)
- int **SetPageDimensions** (double NewPageWidth, double NewPageHeight)
- int **SetPageLayout** (int NewPageLayout)
- int **SetPageMode** (int NewPageMode)
- int **SetPageSize** (string PaperName)
- int **SetPageThumbnail** ()
- int **SetPageTransparencyGroup** (int CS, int Isolate, int Knockout)
- int **SetPrecision** (int NewPrecision)
- int **SetPrinterDevModeFromString** (byte[ ] Source)
- int **SetRenderCropType** (int NewCropType)
- int **SetRenderDCErasePage** (int NewErasePage)
- int **SetRenderDCOffset** (int NewOffsetX, int NewOffsetY)
- int **SetRenderOptions** (int OptionID, int NewValue)
- int **SetRenderScale** (double NewScale)
- int **SetScale** (double NewScale)
- int **SetSignProcessField** (int SignProcessID, string SignatureFieldName)
- int **SetSignProcessFieldBounds** (int SignProcessID, double Left, double Top, double Width, double Height)
- int **SetSignProcessFieldImageFromFile** (int SignProcessID, string ImageFileName, int Options)
- int **SetSignProcessFieldPage** (int SignProcessID, int SignaturePage)
- int **SetSignProcessInfo** (int SignProcessID, string Reason, string Location, string ContactInfo)
- int **SetSignProcessPFXFromFile** (int SignProcessID, string PFXFileName, string PFXPassword)
- int **SetSignProcessPassthrough** (int SignProcessID, int SignatureLength)
- int **SetSignProcessSubFilter** (int SignProcessID, int SubFilter)
- int **SetTableBorderColor** (int TableID, int BorderIndex, double Red, double Green, double Blue)
- int **SetTableBorderColorCMYK** (int TableID, int BorderIndex, double C, double M, double Y, double K)
- int **SetTableBorderWidth** (int TableID, int BorderIndex, double NewWidth)
- int **SetTableCellAlignment** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int New↩ CellAlignment)
- int **SetTableCellBackgroundColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double Red, double Green, double Blue)
- int **SetTableCellBackgroundColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int Last↩ Column, double C, double M, double Y, double K)
- int **SetTableCellBorderColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double Red, double Green, double Blue)
- int **SetTableCellBorderColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double C, double M, double Y, double K)
- int **SetTableCellBorderWidth** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double NewWidth)
- int **SetTableCellContent** (int TableID, int RowNumber, int ColumnNumber, string HTMLText)

- int **SetTableCellPadding** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, int BorderIndex, double NewPadding)
- int **SetTableCellTextColor** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double Red, double Green, double Blue)
- int **SetTableCellTextColorCMYK** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double C, double M, double Y, double K)
- int **SetTableCellTextSize** (int TableID, int FirstRow, int FirstColumn, int LastRow, int LastColumn, double NewTextSize)
- int **SetTableColumnWidth** (int TableID, int FirstColumn, int LastColumn, double NewWidth)
- int **SetTableRowHeight** (int TableID, int FirstRow, int LastRow, double NewHeight)
- int **SetTableThinBorders** (int TableID, int ThinBorders, double Red, double Green, double Blue)
- int **SetTableThinBordersCMYK** (int TableID, int ThinBorders, double C, double M, double Y, double K)
- int **SetTempFile** (string FileName)
- int **SetTempPath** (string NewPath)
- int **SetTextAlign** (int TextAlign)
- int **SetTextCharSpacing** (double CharSpacing)
- int **SetTextColor** (double Red, double Green, double Blue)
- int **SetTextColorCMYK** (double C, double M, double Y, double K)
- int **SetTextColorSep** (string ColorName, double Tint)
- int **SetTextExtractionArea** (double Left, double Top, double Width, double Height)
- int **SetTextExtractionOptions** (int OptionID, int NewValue)
- int **SetTextExtractionScaling** (int Options, double Horizontal, double Vertical)
- int **SetTextExtractionWordGap** (double NewWordGap)
- int **SetTextHighlight** (int Highlight)
- int **SetTextHighlightColor** (double Red, double Green, double Blue)
- int **SetTextHighlightColorCMYK** (double C, double M, double Y, double K)
- int **SetTextHighlightColorSep** (string ColorName, double Tint)
- int **SetTextMode** (int TextMode)
- int **SetTextRise** (double Rise)
- int **SetTextScaling** (double ScalePercentage)
- int **SetTextShader** (string ShaderName)
- int **SetTextSize** (double TextSize)
- int **SetTextSpacing** (double Spacing)
- int **SetTextUnderline** (int Underline)
- int **SetTextUnderlineColor** (double Red, double Green, double Blue)
- int **SetTextUnderlineColorCMYK** (double C, double M, double Y, double K)
- int **SetTextUnderlineColorSep** (string ColorName, double Tint)
- int **SetTextUnderlineCustomDash** (string DashPattern, double DashPhase)
- int **SetTextUnderlineDash** (double DashOn, double DashOff)
- int **SetTextUnderlineDistance** (double UnderlineDistance)
- int **SetTextUnderlineWidth** (double UnderlineWidth)
- int **SetTextWordSpacing** (double WordSpacing)
- int **SetTransparency** (int Transparency)
- int **SetViewerPreferences** (int Option, int NewValue)
- int **SetXFAFormFieldAccess** (string XFAFieldName, int NewAccess)
- int **SetXFAFormFieldBorderColor** (string XFAFieldName, double Red, double Green, double Blue)
- int **SetXFAFormFieldBorderPresence** (string XFAFieldName, int NewPresence)
- int **SetXFAFormFieldBorderWidth** (string XFAFieldName, double BorderWidth)
- int **SetXFAFormFieldValue** (string XFAFieldName, string NewValue)
- int **SetXFAFromString** (byte[ ] Source, int Options)
- int **SetupCustomPrinter** (string CustomPrinterName, int Setting, int NewValue)
- int **SignFile** (string InputFileName, string OpenPassword, string SignatureFieldName, string OutputFileName, string PFXFileName, string PFXPassword, string Reason, string Location, string ContactInfo)
- int **SplitPageText** (int Options)
- int **StartPath** (double StartX, double StartY)

- int **StoreCustomDataFromFile** (string Key, string FileName, int Location, int Options)
- int **StoreCustomDataFromString** (string Key, byte[ ] NewValue, int Location, int Options)
- int **StringResultLength** ()
- int **TestTempPath** ()
- int **TransformFile** (string InputFileName, string Password, string OutputFileName, int TransformType, int Options)
- int **UnlockKey** (string LicenseKey)
- int **Unlocked** ()
- int **UpdateAndFlattenFormField** (int Index)
- int **UpdateAppearanceStream** (int Index)
- int **UseKerning** (int Kern)
- int **UseUnsafeContentStreams** (int SafetyLevel)
- PDFLibrary ()

    Constructor modified by e-Docs USA to initialize the licence key.

5.25.1 Constructor & Destructor Documentation

5.25.1.1 DebenuPDFLibraryDLL0915.PDFLibrary.PDFLibrary ( )

Constructor modified by e-Docs USA to initialize the licence key.

5.25.1.2 DebenuPDFLibraryDLL0915.PDFLibrary.PDFLibrary ( )

Constructor modified by e-Docs USA to initialize the licence key.

5.25.1.3 DebenuPDFLibraryDLL0915.PDFLibrary.PDFLibrary ( )

Constructor modified by e-Docs USA to initialize the licence key.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire/bin/Debug/ThirdParty/DebenuPdfLibrary915.cs

5.26 Scanquire.Public.SQPhotoAlbumFilesystemArchiver.PhotoAlbumInputDialog Class Reference

Inheritance diagram for Scanquire.Public.SQPhotoAlbumFilesystemArchiver.PhotoAlbumInputDialog:

Public Member Functions

- void **Clear** ()

Public Attributes

- ValidatingTextBox< string > **AlbumName**
- CheckBox **LowQualityCopyCheckBox**

Protected Member Functions

- override void **OnShown** (EventArgs e)

Properties

- bool **LowQualityCopy** `[get, set]`

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQArchiver/SQPhotoAlbumFilesystemArchiver.cs

5.27 Scanquire.Public.UserControls.ProgressDialog Class Reference

Inheritance diagram for Scanquire.Public.UserControls.ProgressDialog:

Public Attributes

- ProgressMonitor **ProgressMonitor**
- System.Windows.Forms.Label **CaptionLabel**

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

5.27.1 Member Function Documentation

5.27.1.1 override void Scanquire.Public.UserControls.ProgressDialog.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

The documentation for this class was generated from the following files:

- C:/Code/Scanquire 4/Scanquire.Public/UserControls/ProgressDialog.cs
- C:/Code/Scanquire 4/Scanquire.Public/UserControls/ProgressDialog.Designer.cs

5.28 Scanquire.Public.UserControls.ProgressMonitor Class Reference

Progress bar to monitor progress from a Progress and respond to CancelationToken requests.

Inheritance diagram for Scanquire.Public.UserControls.ProgressMonitor:

Public Types

- enum **StopMonitoringReason** { **None**, **ProcessCompleted**, **ProcessFailed**, **ProcessCanceled** }

Public Member Functions

- void StartMonitoring (Microsoft.Progress< ProgressEventArgs > progress, CancellationTokenSource cancelSource)
- void StartMonitoring< T > (Progress< ProgressEventArgs< T >> progress, CancellationTokenSource cancelSource)
- void **StopMonitoring** (Progress< ProgressEventArgs > progress, StopMonitoringReason reason=Stop↩ MonitoringReason.None)
- void **StopMonitoring**< **T** > (Progress< ProgressEventArgs< T >> progress, StopMonitoringReason reason=StopMonitoringReason.None)

Public Attributes

- BusyStatus **BusyStatus** = new BusyStatus()

Protected Member Functions

- void **OnSupportsCancellationChanged** ()
- void **SetProgressBarCompletedColor** (StopMonitoringReason reason)
- override void Dispose (bool disposing)

  Clean up any resources being used.

Protected Attributes

- CancellationTokenSource **CancellationTokenSource** = null
- bool **_SupportsCancellation** = false

Properties

- bool SupportsCancellation `[get, set]`

  If true, monitors the the cancelation token.

5.28.1 Detailed Description

Progress bar to monitor progress from a Progress and respond to CancelationToken requests.

5.28.2 Member Function Documentation

5.28.2.1 override void Scanquire.Public.UserControls.ProgressMonitor.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

5.28.2.2 void Scanquire.Public.UserControls.ProgressMonitor.StartMonitoring (Microsoft.Progress< ProgressEventArgs > *progress,* CancellationTokenSource *cancelSource*)

Must be called before you can monitor anything else./>

5.28.2.3 void **Scanquire.Public.UserControls.ProgressMonitor.StartMonitoring**< T > (Progress< ProgressEventArgs< T >> *progress,* CancellationTokenSource *cancelSource*)

Must be called before you can monitor anything else./>

5.28.3 Property Documentation

5.28.3.1 bool Scanquire.Public.UserControls.ProgressMonitor.SupportsCancellation [get], [set]

If true, monitors the the cancelation token.

The documentation for this class was generated from the following files:

- C:/Code/Scanquire 4/Scanquire.Public/UserControls/ProgressMonitor.cs
- C:/Code/Scanquire 4/Scanquire.Public/UserControls/ProgressMonitor.Designer.cs

5.29 Scanquire.Public.UserControls.ProgressMonitorToolstripItem Class Reference

Wrapper to allow a ProgressMonitor to be placed on a ToolStrip.

Inheritance diagram for Scanquire.Public.UserControls.ProgressMonitorToolstripItem:

Public Member Functions

- void **StartMonitoring** (Progress< ProgressEventArgs > progress, CancellationTokenSource cancelSource)
- void **StartMonitoring**< **T** > (Progress< ProgressEventArgs< T >> progress, CancellationTokenSource cancelSource)
- void **StopMonitoring** (Progress< ProgressEventArgs > progress, ProgressMonitor.StopMonitoringReason reason=ProgressMonitor.StopMonitoringReason.None)
- void **StopMonitoring**< **T** > (Progress< ProgressEventArgs< T >> progress, ProgressMonitor.Stop↩MonitoringReason reason=ProgressMonitor.StopMonitoringReason.None)

Public Attributes

- ProgressMonitor **Value**

Protected Member Functions

- override void **OnHostedControlResize** (EventArgs e)

5.29.1 Detailed Description

Wrapper to allow a ProgressMonitor to be placed on a ToolStrip.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/UserControls/ProgressMonitorToolstripItem.cs

5.30 Scanquire.Public.SQFilesystemConnector.SaveFileResult Class Reference

Properties

- bool **Success** `[get, set]`
- int **VersionNumber** `[get, set]`
- bool **RetryRequested** `[get, set]`
- string **AbsolutePath** `[get, set]`
- string **RelativePath** `[get, set]`
- Uri **AbsoluteUri** `[get, set]`
- string **RelativeUriPath** `[get, set]`

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFilesystemConnector/SQFilesystemConnector.cs

5.31 Scanquire.Properties.Scanquire Class Reference

Inheritance diagram for Scanquire.Properties.Scanquire:

Properties

- static Scanquire **Default** `[get]`
- string **SessionRecoveryFilePath** `[get, set]`
- string **PreviousArchiverName** `[get, set]`
- bool **PortableMode** `[get]`

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire/Properties/Scanquire.Designer.cs

5.32 Scanquire.Public.SerializedObjectDictionary< T > Class Template Reference

Serializes a directory of object to a local directory.

Public Member Functions

- SerializedObjectDictionary ()
- **SerializedObjectDictionary** (string directoryPath)
- void EnsureKeyExists (string key)

Ensure that the provided key exists in the dictionary.

- bool **ContainsKey** (string key)
- bool Remove (string key)

  Remove an object from the dictionary.
- bool TryGetValue (string key, out T value)

  Try to get a value from the dictionary.
- Dictionary< string, T >.Enumerator **GetEnumerator** ()
- void Refresh ()

  Clear all entries from the dictionary and reload it from the directory.
- string GetFilePathFromKey (string key)

  Returns

  The absolute path to the file corresponding to the provided dictionary key.

Protected Member Functions

- void **NotifyCollectionUpdated** ()
- string GetKeyFromFilePath (string path)

  Returns

  The dictionary key for a given file.
- void Serialize (string key, T value)

  Serialize an object to a file.
- bool TryDeserialize (string path, out T value)

  Attempt to deserialize an individual object file.

Properties

- virtual string DirectoryPath `[get, set]`

  Path to the directory to serialize the objects to.
- T this[string key] `[get, set]`

  Retrieve the value that corresponds to the provided key.
- ICollection< string > **Keys** `[get]`
- ICollection< T > **Values** `[get]`

Events

- EventHandler **CollectionUpdated**

5.32.1 Detailed Description

Serializes a directory of object to a local directory.

Does not handle updates to the individual objects.

Type Constraints

T* : *class

5.32.2 Constructor & Destructor Documentation

5.32.2.1 Scanquire.Public.SerializedObjectDictionary< T >.SerializedObjectDictionary ( )

Intended for initialization from serializer only. For manual creation, use one of the other constructors.

5.32.3 Member Function Documentation

5.32.3.1 void Scanquire.Public.SerializedObjectDictionary< T >.EnsureKeyExists (string *key*)

Ensure that the provided key exists in the dictionary.

Exceptions

KeyNotFoundException	If the key does not exist in the dictionary.

5.32.3.2 void Scanquire.Public.SerializedObjectDictionary< T >.Refresh ( )

Clear all entries from the dictionary and reload it from the directory.

5.32.3.3 bool Scanquire.Public.SerializedObjectDictionary< T >.Remove (string *key*)

Remove an object from the dictionary.

Parameters

key	The dictionary key of the object to remove.

Returns

On success: true. On failure (key not found, or error deleting): false.

Also deletes the corresponding file.

5.32.3.4 void Scanquire.Public.SerializedObjectDictionary< T >.Serialize (string *key*, T *value*) [protected]

Serialize an object to a file.

Parameters

key	The dictionary key for the object (also the serialized file name).
value	The object to serialize

5.32.3.5 bool Scanquire.Public.SerializedObjectDictionary< T >.TryDeserialize (string *path*, out T *value*) [protected]

Attempt to deserialize an individual object file.

Parameters

path	Path to the file to deserialize.
value	On success: The serialized file. On failure: null.

Returns

On success: true. On failure: false.

5.32.3.6 bool Scanquire.Public.SerializedObjectDictionary< T >.TryGetValue (string *key*, out T *value*)

Try to get a value from the dictionary.

Parameters

key	Dictionary key of the value.
value	On success: The value of the provided key. On failure: null.

Returns

On success: True. On failure (key not found): False.

5.32.4 Property Documentation

5.32.4.1 virtual string Scanquire.Public.SerializedObjectDictionary< T >.DirectoryPath [get], [set]

Path to the directory to serialize the objects to.

Directory should be dedicated to objects of the provided type.

5.32.4.2 T Scanquire.Public.SerializedObjectDictionary< T >.this[string key] [get], [set]

Retrieve the value that corresponds to the provided key.

Returns

If the key exists: the value that corresponds to the key. If the key does not exist: null.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SerializedObjectDictionary.cs

5.33 Scanquire.Public.SerializedObjectDictionaryCollection< T > Class Template Reference

Serializes a dictionary of SerializedObjectDictionary to a directory.

Public Member Functions

- SerializedObjectDictionaryCollection ()
- **SerializedObjectDictionaryCollection** (string directoryPath)
- bool **ContainsKey** (string key)
- void EnsureKeyExists (string key)

  Ensure that the provided key exists in the dictionary.
- bool Remove (string key)

  Remove an object from the dictionary.
- bool TryGetValue (string key, out SerializedObjectDictionary< T > value)

  Try to get a value from the dictionary.
- Dictionary< string, SerializedObjectDictionary< T > >.Enumerator **GetEnumerator** ()
- void Refresh ()

  Clear all entries and reload from the directory.

Protected Member Functions

- void **NotifyCollectionUpdated** ()
- bool TryDeserialize (string directory, out SerializedObjectDictionary< T > value)

  Attempt to deserialize an individual directory
- string **GetKeyFromDirectory** (string path)
- string **GetDirectoryFromKey** (string key)

Properties

- virtual string DirectoryPath `[get, set]`

  Path to the directory to serialize the objects to.
- SerializedObjectDictionary< T > **this[string key]** `[get]`
- ICollection< string > **Keys** `[get]`
- ICollection< SerializedObjectDictionary< T > > **Values** `[get]`

Events

- EventHandler CollectionUpdated

  Raised when an item is added or removed from the dictionary.

5.33.1 Detailed Description

Serializes a dictionary of SerializedObjectDictionary to a directory.

Does not handle updates to the individual objects.

Assume you want to organize a group of Students by Family

Type Constraints

T* : *class

5.33.2 Constructor & Destructor Documentation

5.33.2.1 Scanquire.Public.SerializedObjectDictionaryCollection< T >.SerializedObjectDictionaryCollection ( )

Intended for initialization from serializer only. For manual creation, use one of the other constructors.

5.33.3 Member Function Documentation

5.33.3.1 void Scanquire.Public.SerializedObjectDictionaryCollection< T >.EnsureKeyExists (string *key*)

Ensure that the provided key exists in the dictionary.

Exceptions

KeyNotFoundException	If the key does not exist in the dictionary.

5.33.3.2 void Scanquire.Public.SerializedObjectDictionaryCollection< T >.Refresh ( )

Clear all entries and reload from the directory.

5.33.3.3 bool Scanquire.Public.SerializedObjectDictionaryCollection< T >.Remove (string *key*)

Remove an object from the dictionary.

Parameters

key	The dictionary key of the object to remove.

Returns

On success: true. On failure (key not found, or error deleting): false.

Also deletes the corresponding file.

5.33.3.4 bool Scanquire.Public.SerializedObjectDictionaryCollection< T >.TryDeserialize (string *directory,* out SerializedObjectDictionary< T > *value*) [protected]

Attempt to deserialize an individual directory

Parameters

directory	Directory to deserialize.
value	On success: A SerializedObjectDictionary for the specified directory. On failure: null.

Returns

On success: true. On failure (Directory not found or deserialize error): false.

5.33.3.5 bool Scanquire.Public.SerializedObjectDictionaryCollection< T >.TryGetValue (string *key,* out SerializedObjectDictionary< T > *value*)

Try to get a value from the dictionary.

Parameters

key	Dictionary key of the value.
value	On success: The value of the provided key. On failure: null.

Returns

On success: True. On failure (key not found): False.

5.33.4 Property Documentation

5.33.4.1 virtual string Scanquire.Public.SerializedObjectDictionaryCollection< T >.DirectoryPath [get], [set]

Path to the directory to serialize the objects to.

Directory should be dedicated to objects of the provided type.

5.33.5 Event Documentation

5.33.5.1 EventHandler Scanquire.Public.SerializedObjectDictionaryCollection< T >.CollectionUpdated

Raised when an item is added or removed from the dictionary.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SerializedObjectDictionaryCollection.cs

5.34 Scanquire.Public.UserControls.SessionLogViewer Class Reference

Dialog to view the Trace history for a selected TextWriterTraceListener.

Inheritance diagram for Scanquire.Public.UserControls.SessionLogViewer:

Protected Member Functions

- async override void **OnShown** (EventArgs e)
- async Task LoadListeners ()

  Populate Listeners with all TextWriterTraceListeners that output to files.
- bool TryGetListenerFilePath (TraceListener listener, out string path)

  Attempt to get the output filepath for a TraceListener.
- async Task LoadSessionLog (string traceFilePath)

  Populate SessionLogTextBox with the contents of the specified trace file.
- override void Dispose (bool disposing)

  Clean up any resources being used.

Protected Attributes

- readonly SynchronizedBindingList< KeyValuePair< string, string > > FileListeners

  List of available TextWriterTraceListeners that are writing to a FileStream.

5.34.1 Detailed Description

Dialog to view the Trace history for a selected TextWriterTraceListener.

5.34.2 Member Function Documentation

5.34.2.1 override void Scanquire.Public.UserControls.SessionLogViewer.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

5.34.2.2 async Task Scanquire.Public.UserControls.SessionLogViewer.LoadListeners ( ) `[protected]`

Populate Listeners with all TextWriterTraceListeners that output to files.

5.34.2.3 async Task Scanquire.Public.UserControls.SessionLogViewer.LoadSessionLog (string *traceFilePath*) `[protected]`

Populate SessionLogTextBox with the contents of the specified trace file.

File is opened with read only access, so it should not interfere with continued tracing operations.

5.34.2.4 bool Scanquire.Public.UserControls.SessionLogViewer.TryGetListenerFilePath (TraceListener *listener,* out string *path*) [protected]

Attempt to get the output filepath for a TraceListener.

Parameters

path	On False: null. On True: The absolute path to the file being written to by the specified Trace↩ Listener.

Returns

False if listener is not a TextWriterTraceListener that is outputting to a file. True if listener is a TextWriter↩ TraceListener that is outputting to a file.

5.34.3 Member Data Documentation

5.34.3.1 readonly SynchronizedBindingList<KeyValuePair<string, string> > Scanquire.Public.UserControls.SessionLog↩ Viewer.FileListeners [protected]

List of available TextWriterTraceListeners that are writing to a FileStream.

The documentation for this class was generated from the following files:

- C:/Code/Scanquire 4/Scanquire.Public/UserControls/SessionLogViewer.cs
- C:/Code/Scanquire 4/Scanquire.Public/UserControls/SessionLogViewer.Designer.cs

5.35 Scanquire.Public.ThirdParty.ShareFileV3 Class Reference

Static Public Member Functions

- static OAuth2Token Authenticate (string hostname, string clientId, string clientSecret, string username, string password)

  Authenticate via username/password
- static void **addAuthorizationHeader** (HttpWebRequest request, OAuth2Token token)
- static string **GetHostname** (OAuth2Token token)
- static void GetRoot (OAuth2Token token, bool getChildren=false)

  Get the root level Item for the provided user. To retrieve Children the $expand=Children parameter can be added.
- static string **GetSubFolder** (OAuth2Token token, string rPath)
- static string **GetRootSharedFolder** (OAuth2Token token)
- static string **GetRootId** (OAuth2Token token)
- static void GetItemById (OAuth2Token token, string id)

  Get a single Item by Id.
- static void GetFolderWithQueryParameters (OAuth2Token token, string id)

  Get a folder using some of the common query parameters that are available. This will add the expand, select parameters. The following are used: expand=Children to get any Children of the folder select=Id,Name,Children/↩ Id,Children/Name,Children/CreationDate to get the Id, Name of the folder and the Id, Name, CreationDate of any Children
- static string CreateFolder (OAuth2Token token, string parentId, string name, string description)

  Create a new folder in the given parent folder.
- static void UpdateItem (OAuth2Token token, string itemId, string name, string description)

  Update the name and description of an Item.
- static void DeleteItem (OAuth2Token token, string itemId)

  Delete an Item by Id.
- static void DownloadItem (OAuth2Token token, string itemId, string localPath)

  Downloads a single Item.
- static void UploadFile (OAuth2Token token, string parentId, string fileName, byte[ ] data)

  Uploads a File using the Standard upload method with a multipart/form mime encoded POST.
- static void UploadMultiPartFile (string parameterName, string uploadUrl, string fileName, byte[ ] data)

  Does a multipart form post upload of a file to a url.

5.35.1 Member Function Documentation

5.35.1.1 static **OAuth2Token** Scanquire.Public.ThirdParty.ShareFileV3.Authenticate (string *hostname,* string *clientId,* string *clientSecret,* string *username,* string *password*) [static]

Authenticate via username/password

Parameters

hostname	hostname like "myaccount.sharefile.com"
clientId	my client id
clientSecret	my client secret
username	my@user.name
password	mypassword

Returns

5.35.1.2 static string Scanquire.Public.ThirdParty.ShareFileV3.CreateFolder (**OAuth2Token** *token,* string *parentId,* string *name,* string *description*) [static]

Create a new folder in the given parent folder.

Parameters

token	the OAuth2Token returned from Authenticate
parentId	the parent folder in which to create the new folder
name	the folder name
description	the folder description

5.35.1.3 static void Scanquire.Public.ThirdParty.ShareFileV3.DeleteItem (**OAuth2Token** *token,* string *itemId*) [static]

Delete an Item by Id.

Parameters

token	the OAuth2Token returned from Authenticate
itemId	the id of the item to delete

5.35.1.4 static void Scanquire.Public.ThirdParty.ShareFileV3.DownloadItem (**OAuth2Token** *token,* string *itemId,* string *localPath*) [static]

Downloads a single Item.

Parameters

token	the OAuth2Token returned from Authenticate
itemId	the id of the item to download
localPath	where to download the item to, like "c:\\path\\to\\the.file". If downloading a folder the localPath name should end in .zip.

5.35.1.5 static void Scanquire.Public.ThirdParty.ShareFileV3.GetFolderWithQueryParameters (OAuth2Token *token*, string *id*) [static]

Get a folder using some of the common query parameters that are available. This will add the expand, select parameters. The following are used: expand=Children to get any Children of the folder select=Id,Name,Children/↩Id,Children/Name,Children/CreationDate to get the Id, Name of the folder and the Id, Name, CreationDate of any Children

Parameters

token	the OAuth2Token returned from Authenticate
id	a folder id

5.35.1.6 static void Scanquire.Public.ThirdParty.ShareFileV3.GetItemById (OAuth2Token *token*, string *id*) [static]

Get a single Item by Id.

Parameters

token	the OAuth2Token returned from Authenticate
id	an item id

5.35.1.7 static void Scanquire.Public.ThirdParty.ShareFileV3.GetRoot (OAuth2Token *token*, bool *getChildren* = false) [static]

Get the root level Item for the provided user. To retrieve Children the $expand=Children parameter can be added.

Parameters

token	the OAuth2Token returned from Authenticate
getChildren	retrieve Children Items if true, default is false

5.35.1.8 static void Scanquire.Public.ThirdParty.ShareFileV3.UpdateItem (OAuth2Token *token*, string *itemId*, string *name*, string *description*) [static]

Update the name and description of an Item.

Parameters

token	the OAuth2Token returned from Authenticate
itemId	the id of the item to update
name	the item name
description	the item description

5.35.1.9 static void Scanquire.Public.ThirdParty.ShareFileV3.UploadFile (OAuth2Token *token*, string *parentId*, string *fileName*, byte[ ] *data*) [static]

Uploads a File using the Standard upload method with a multipart/form mime encoded POST.

Parameters

token	the OAuth2Token returned from Authenticate
parentId	where to upload the file
localPath	the full path of the file to upload, like "c:\\path\\to\\file.name"

5.35.1.10 static void Scanquire.Public.ThirdParty.ShareFileV3.UploadMultiPartFile (string *parameterName,* string *uploadUrl,* string *fileName,* byte[ ] *data*) [static]

Does a multipart form post upload of a file to a url.

Parameters

parameterName	multipart parameter name. File1 for a standard upload.
file	the FileInfo to upload
uploadUrl	the url of the server to upload to

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/ThirdParty/ShareFile.cs

5.36 Scanquire.Public.Sharepoint.SharepointConnector Class Reference

Description of SharepointConnector.

Public Member Functions

- Byte[ ] **RetrieveFile** (string camlQuery)
- void **SendFile** (byte[ ] fileContents, string fileName, Dictionary< string, object > fieldValues)
- object **GetUserObject** (string loginName)

Static Public Member Functions

- static string **UnspaceName** (string visibleName)

Public Attributes

- SP.Web **Site** = null

Properties

- long **Timeout** [get, set]
- IWin32Window **APPHwnd** [get, set]
- string **Domain** [get, set]
- string **UserName** [get, set]
- string **Password** [get, set]
- string **Host** [get, set]
- SP.ClientAuthenticationMode **AuthMode** [get, set]
- string **SiteName** [get, set]
- string **LibraryName** [get, set]

5.36.1 Detailed Description

Description of SharepointConnector.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Sharepoint/SharepointConnector.cs

5.37 Scanquire.Public.Sharepoint.SharepointCredentials Class Reference

Inheritance diagram for Scanquire.Public.Sharepoint.SharepointCredentials:

Public Types

- enum **ValidationResult**

Public Member Functions

- override bool **ValidateChildren** ()

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Properties

- Timeout **Timeout** `[get, set]`
- string **ServerAddress** `[get, set]`
- string **UserName** `[get, set]`
- string **Password** `[get, set]`
- string **Domain** `[get, set]`
- bool **ShowDomain** `[get, set]`

5.37.1 Member Function Documentation

5.37.1.1 override void Scanquire.Public.Sharepoint.SharepointCredentials.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

The documentation for this class was generated from the following files:

- C:/Code/Scanquire 4/Scanquire.Public/Sharepoint/SharepointCredentials.cs
- C:/Code/Scanquire 4/Scanquire.Public/Sharepoint/SharepointCredentials.Designer.cs

5.38 Scanquire.Public.Sharepoint.SharepointCredentialsDialog Class Reference

Protected Member Functions

- override void Dispose (bool disposing)
  Clean up any resources being used.

5.38.1 Member Function Documentation

5.38.1.1 override void Scanquire.Public.Sharepoint.SharepointCredentialsDialog.Dispose (bool *disposing*) [protected]

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Sharepoint/SharepointCredentialsDialog.Designer.cs

5.39 Scanquire.Public.Sharepoint.SharepointCredentialsInputDialog Class Reference

Inheritance diagram for Scanquire.Public.Sharepoint.SharepointCredentialsInputDialog:

Public Attributes

- ValidatingTextBox< string > **Host**
- ValidatingTextBox< string > **Domain**
- ValidatingTextBox< string > **User**
- ValidatingTextBox< string > **Password**

Properties

- SP.ClientAuthenticationMode **AuthenticationMode** [get, set]

5.39.1 Member Data Documentation

5.39.1.1 ValidatingTextBox<string> Scanquire.Public.Sharepoint.SharepointCredentialsInputDialog.Domain

Initial value:

```
= new ValidatingTextBox<string>()
        { RequiresValue = true }
```

5.39.1.2 ValidatingTextBox<string> Scanquire.Public.Sharepoint.SharepointCredentialsInputDialog.Host

Initial value:

```
= new ValidatingTextBox<string>()
        { RequiresValue = true }
```

5.39.1.3 ValidatingTextBox<string> Scanquire.Public.Sharepoint.SharepointCredentialsInputDialog.Password

Initial value:

```
= new ValidatingTextBox<string>()
        {
            RequiresValue = true,
            PasswordChar = '*'
        }
```

5.39.1.4 ValidatingTextBox<string> Scanquire.Public.Sharepoint.SharepointCredentialsInputDialog.User

Initial value:

```
= new ValidatingTextBox<string>()
        { RequiresValue = true }
```

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Sharepoint/SharepointCredentialsInputDialog.cs

5.40 Scanquire.Public.Sharepoint.SharepointCredentialsManager Class Reference

Inheritance diagram for Scanquire.Public.Sharepoint.SharepointCredentialsManager:

Protected Member Functions

- bool **HasTimedOut** ()
- void **SetLastActionTime** ()
- override void Dispose (bool disposing)

  Clean up any resources being used.

Protected Attributes

- DateTime **LastActionTime** = DateTime.MinValue

Properties

- string **ServerAddress** `[get, set]`
- string **UserName** `[get, set]`
- string **Password** `[get, set]`
- string **Domain** `[get, set]`
- bool **ShowDomain** `[get, set]`
- bool **EnableTimeout** `[get, set]`
- int **Timeout** `[get, set]`

5.40.1 Member Function Documentation

5.40.1.1 override void Scanquire.Public.Sharepoint.SharepointCredentialsManager.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

The documentation for this class was generated from the following files:

- C:/Code/Scanquire 4/Scanquire.Public/Sharepoint/SharepointCredentialsManager.cs
- C:/Code/Scanquire 4/Scanquire.Public/Sharepoint/SharepointCredentialsManager.Designer.cs

5.41 Scanquire.Public.SQArchiverBase Class Reference

A base implementation of an ISQArchiver, that handles the majority of common cases.

Inheritance diagram for Scanquire.Public.SQArchiverBase:

Public Member Functions

- virtual IEnumerable< Task< SQImage > > Acquire (SQAcquireIntent intent, SQAcquireSource source, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

    Handles retrieving documents from an archiver and feeding them as images into Scanquire
- virtual IEnumerable< Task< SQImage > > **AcquireForNew** (SQAcquireSource source, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireForAppend** (SQAcquireSource source, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireForInsert** (SQAcquireSource source, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > AcquireFromScanner (IProgress< ProgressEventArgs > progress, CancellationToken cToken)

    Default method for acquiring from the scanner
- virtual IEnumerable< Task< SQImage > > **AcquireFromScannerForNew** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireFromScannerForAppend** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireFromScannerForInsert** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- abstract IEnumerable< Task< SQImage > > AcquireFromFile (IProgress< ProgressEventArgs > progress, CancellationToken cToken)

    Acquiring from file depends on the type of archive, so defer to custom archiver implementation
- virtual IEnumerable< Task< SQImage > > **AcquireFromFileForNew** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireFromFileForAppend** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireFromFileForInsert** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireFromCommand** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireFromCommandForNew** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireFromCommandForAppend** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireFromCommandForInsert** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)

- virtual IEnumerable< Task< SQImage > > **AcquireFromCustom** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireFromCustomForNew** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireFromCustomForAppend** (IProgress< Progress↩ EventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **AcquireFromCustomForInsert** (IProgress< ProgressEvent↩ Args > progress, CancellationToken cToken)
- virtual async Task< IList< ISQCommand > > **ReadCommands** (SQImage image, int documentNumber, int pageNumber, CancellationToken cToken)
- virtual async Task< IList< ISQCommand > > TranslateCommands (IEnumerable< ISQCommand > commands, int documentNumber, int pageNumber, CancellationToken cToken)

  Translate a set of commands.</summar>
  To ignore a command, convert it to SQCommand_Null
- virtual async Task Send (IEnumerable< SQImage > images, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

  Send the specified images to the destination archive.
- virtual async Task Send (SQDocument document, int documentNumber, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

  Sends a single document to the archive.
- abstract Task Send (SQFile file, int fileNumber, IProgress< ProgressEventArgs > progress, Cancellation↩ Token cToken)

  Sends a single file (encoded document) to the archiver.

Protected Attributes

- string **_FileWriterName** = "PDF"

Properties

- virtual string ScannerName `[get, set]`

  If set, will specify the scanner's device name to automatically use.
- virtual string ScannerSetting `[get, set]`

  If set will specify the pre-defined custom setting to apply to the scan.
- virtual string CommandReaderName `[get, set]`

  Common name of the desired SQCommandReader.
- ISQCommandReader CommandReader `[get, set]`

  Command Reader to read commands from images during a save operation.
- virtual string FileWriterName `[get, set]`

  Name of the defined ISQFileWriter to use when compiling files.
- ISQFileWriter FileWriter `[get, set]`

  File Writer to compile files during a save operation.
- virtual string CommandImageBuilderName `[get, set]`

  Name of the defined ISQCommandImageBuilder to use when SQCommandImages are requested.
- ISQCommandImageBuilder CommandImageBuilder `[get, set]`

  The ISQCommandImageBuilder to use when SQCommandImages are requested.

5.41.1 Detailed Description

A base implementation of an ISQArchiver, that handles the majority of common cases.

5.41.2 Member Function Documentation

5.41.2.1 virtual IEnumerable<Task<SQImage> > Scanquire.Public.SQArchiverBase.Acquire (SQAcquireIntent *intent,* SQAcquireSource *source,* IProgress< ProgressEventArgs > *progress,* CancellationToken *cancelToken*) [virtual]

Handles retrieving documents from an archiver and feeding them as images into Scanquire

Parameters

intent	The intent of the retrieval (new doc, append, insert, etc)
source	The expected source (scanner, file, etc) to retrieve the document from.

Implements Scanquire.Public.ISQArchiver.

5.41.2.2 abstract IEnumerable<Task<SQImage> > Scanquire.Public.SQArchiverBase.AcquireFromFile (IProgress< ProgressEventArgs > *progress*, CancellationToken *cToken*) [pure virtual]

Acquiring from file depends on the type of archive, so defer to custom archiver implementation

Implemented in Scanquire.Public.SQFilesystemArchiver.

5.41.2.3 virtual IEnumerable<Task<SQImage> > Scanquire.Public.SQArchiverBase.AcquireFromScanner (IProgress< ProgressEventArgs > *progress*, CancellationToken *cToken*) [virtual]

Default method for acquiring from the scanner

5.41.2.4 virtual async Task Scanquire.Public.SQArchiverBase.Send (IEnumerable< SQImage > *images*, IProgress< ProgressEventArgs > *progress*, CancellationToken *cToken*) [virtual]

Send the specified images to the destination archive.

Implements Scanquire.Public.ISQArchiver.

5.41.2.5 virtual async Task Scanquire.Public.SQArchiverBase.Send (SQDocument *document*, int *documentNumber*, IProgress< ProgressEventArgs > *progress*, CancellationToken *cToken*) [virtual]

Sends a single document to the archive.

Parameters

document	The document to archive.
document↩ Number	For sending multiple documents, the index of the document in the overall listing.

5.41.2.6 abstract Task Scanquire.Public.SQArchiverBase.Send (SQFile *file*, int *fileNumber*, IProgress< ProgressEventArgs > *progress*, CancellationToken *cToken*) [pure virtual]

Sends a single file (encoded document) to the archiver.

Parameters

file	The file to archive.
fileNumber	For sending multiple files, the index of the file in the overall listing.
progress	
cToken	

Returns

Implements Scanquire.Public.ISQArchiver.

Implemented in Scanquire.Public.SQFilesystemArchiver.

5.41.3 Property Documentation

5.41.3.1 ISQCommandImageBuilder Scanquire.Public.SQArchiverBase.CommandImageBuilder [get], [set]

The ISQCommandImageBuilder to use when SQCommandImages are requested.

5.41.3.2 virtual string Scanquire.Public.SQArchiverBase.CommandImageBuilderName [get], [set]

Name of the defined ISQCommandImageBuilder to use when SQCommandImages are requested.

5.41.3.3 ISQCommandReader Scanquire.Public.SQArchiverBase.CommandReader [get], [set]

Command Reader to read commands from images during a save operation.

5.41.3.4 virtual string Scanquire.Public.SQArchiverBase.CommandReaderName [get], [set]

Common name of the desired SQCommandReader.

5.41.3.5 ISQFileWriter Scanquire.Public.SQArchiverBase.FileWriter [get], [set]

File Writer to compile files during a save operation.

5.41.3.6 virtual string Scanquire.Public.SQArchiverBase.FileWriterName [get], [set]

Name of the defined ISQFileWriter to use when compiling files.

5.41.3.7 virtual string Scanquire.Public.SQArchiverBase.ScannerName [get], [set]

If set, will specify the scanner's device name to automatically use.

This must be verbatim to the name that appears in the Twain scanner source menu.

5.41.3.8 virtual string Scanquire.Public.SQArchiverBase.ScannerSetting [get], [set]

If set will specify the pre-defined custom setting to apply to the scan.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQArchiver/SQArchiverBase.cs

5.42 Scanquire.Public.SQArchivers Class Reference

Singleton collection of available SQArchivers.

Inheritance diagram for Scanquire.Public.SQArchivers:

Properties

- override string **DirectoryPath** [get]
- static SQArchivers **Instance** [get]

Additional Inherited Members

5.42.1 Detailed Description

Singleton collection of available SQArchivers.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQArchiver/SQArchivers.cs

5.43 Scanquire.Public.UserControls.SQArchiverSelector Class Reference

User dropdown control to select a single SQArchiver from all configured SQArchivers.

Inheritance diagram for Scanquire.Public.UserControls.SQArchiverSelector:

Public Member Functions

- bool **TryGetSelectedArchiver** (out ISQArchiver archiver)
- bool **TryGetSelectedArchiverName** (out string name)
- bool **TryGetSelectedArchiverInfo** (out string name, out ISQArchiver archiver)
- async Task LoadAllArchivers ()

  Refresh the listing of available Archivers from SQArchivers.Instance

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Properties

- SynchronizedBindingList< KeyValuePair< string, ISQArchiver > > **Archivers** `[get]`
- ISQArchiver **SelectedArchiver** `[get, set]`
- string **SelectedArchiverName** `[get, set]`

5.43.1 Detailed Description

User dropdown control to select a single SQArchiver from all configured SQArchivers.

5.43.2 Member Function Documentation

5.43.2.1 override void Scanquire.Public.UserControls.SQArchiverSelector.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

5.43.2.2 async Task Scanquire.Public.UserControls.SQArchiverSelector.LoadAllArchivers ( )

Refresh the listing of available Archivers from SQArchivers.Instance

The documentation for this class was generated from the following files:

- C:/Code/Scanquire 4/Scanquire.Public/UserControls/SQArchiverSelector.cs
- C:/Code/Scanquire 4/Scanquire.Public/UserControls/SQArchiverSelector.Designer.cs

5.44 Scanquire.Public.SQBarcodeEncoder Class Reference

The default barcode encoder. Barcode commands are encoded to SQBC{Numeric CommandId}{Value Separator}{Command Text}

Inheritance diagram for Scanquire.Public.SQBarcodeEncoder:

Public Types

- enum CommandId {
TerminateDocument = 001, **Page_Bookmark** = 002, **Page_Exclude** = 003, **Document_IndexField** = 004, **Document_Password** = 005, **Document_OwnerPassword** = 006, **Document_UserPassword** = 007, **Page_Size** = 008,
Page_DrawImage = 009, **Page_DrawText** = 010 }

    Mapping of commands to a numeric command id

Public Member Functions

- virtual bool TryEncode (ISQCommand command, out string value, out string caption)

    Attempt to encode a command to a barcode value.
- bool TryParse (string barcodeText, out int commandID, out string value)

    Attempt to separate an encoded barcode into a commnandID and encoded value string
- virtual bool TryParse (string barcodeText, out IEnumerable< ISQCommand > commands)

    Attempt to decode an encoded barcode string into a series of ISQCommands

Protected Member Functions

- virtual string **EncodeValue** (CommandId commandId, string value)
- virtual string EncodeValue (int commandId, string value)

    Base encoding method.
- virtual bool TryEncodeCustom (ISQCommand command, out string value, out string caption)

    Placeholder for sub-classes to handle encoding of commands not accounted for.
- virtual bool TryEncode_Document_IndexField (SQCommand_Document_IndexField command, out string value, out string caption)

    Attempt to encode a Document_IndexField command.
- virtual bool TryEncode_Document_OwnerPassword (SQCommand_Document_OwnerPassword command, out string value, out string caption)

Attempt to encode a Document_OwnerPassword command.

- virtual bool TryEncode_Document_Password (SQCommand_Document_Password command, out string value, out string caption)

  Attempt to encode a Document_Password command.
- virtual bool TryEncode_Document_UserPassword (SQCommand_Document_UserPassword command, out string value, out string caption)

  Attempt to encode a Document_UserPassword command.
- virtual bool TryEncode_Page_Bookmark (SQCommand_Page_Bookmark command, out string value, out string caption)

  Attempt to encode a Page_Bookmark command.
- virtual bool TryEncode_Page_Exclude (SQCommand_Page_Exclude command, out string value, out string caption)

  Attempt to encode a Page_Exclude command.
- virtual bool TryEncode_TerminateDocument (SQCommand_TerminateDocument command, out string value, out string caption)

  Attempt to encode a TerminateDocument command.
- virtual bool TryParseCustom (int commandId, string barcodeValue, out IEnumerable< ISQCommand > commands)

  Placeholder for subclasses to handle unacounted for decoding.
- virtual bool TryParseCustom (string barcodeText, out IEnumerable< ISQCommand > commands)

  Placeholder for subclasses to handle unacounted for decoding.
- virtual bool TryParse_Page_Bookmark (string barcodeValue, out IEnumerable< ISQCommand > commands)

  Attempt to decode a Page_Bookmark command.
- virtual bool TryParse_Document_OwnerPassword (string barcodeValue, out IEnumerable< ISQCommand > commands)

  Attempt to decode a Document_OwnerPassword command.
- virtual bool TryParse_Document_UserPassword (string barcodeValue, out IEnumerable< ISQCommand > commands)

  Attempt to decode a Document_UserPassword command.
- virtual bool TryParse_Document_Password (string barcodeValue, out IEnumerable< ISQCommand > commands)

  Attempt to decode a Document_Password command.
- virtual bool TryParse_Document_IndexField (string barcodeValue, out IEnumerable< ISQCommand > commands)

  Attempt to decode a Document_IndexField command

Properties

- string CommandPrefix `[get, set]`

  Prefix to apply to barcode values to differentiate between barcode sources.
- int CommandIdPadding `[get, set]`

  Number of characters to pad the command id to.
- char ValueSeparator `[get, set]`

  Character separator to apply between the commandid and the command value.
- char IndexFieldSeparator `[get, set]`

  For index field commands, the character to apply between the field namd and field value.

5.44.1 Detailed Description

The default barcode encoder. Barcode commands are encoded to SQBC{Numeric CommandId}{Value Separator}{Command Text}

5.44.2 Member Enumeration Documentation

5.44.2.1 enum Scanquire.Public.SQBarcodeEncoder.CommandId

Mapping of commands to a numeric command id

5.44.3 Member Function Documentation

5.44.3.1 virtual string Scanquire.Public.SQBarcodeEncoder.EncodeValue (int *commandId,* string *value*) [protected], [virtual]

Base encoding method.

Parameters

commandId	Mapped command id of the command type to encode.
value	Encoded value to apply.

Returns

The full encoded barcode text.

5.44.3.2 virtual bool Scanquire.Public.SQBarcodeEncoder.TryEncode (ISQCommand *command,* out string *value,* out string *caption*) [virtual]

Attempt to encode a command to a barcode value.

Parameters

command	The command to encode.
value	If successful, the encoded barcode text.
caption	If successful, the encoded caption text.

Returns

True on success, false on failure.

Implements Scanquire.Public.ISQBarcodeEncoder.

5.44.3.3 virtual bool Scanquire.Public.SQBarcodeEncoder.TryEncode_Document_IndexField (SQCommand↩ _Document_IndexField *command,* out string *value,* out string *caption*) [protected], [virtual]

Attempt to encode a Document_IndexField command.

Parameters

command	The command to encode
value	If successful, the encoded barcode text.
caption	If successful, the encoded caption text.

Returns

True on success, false on failure.

5.44.3.4 virtual bool Scanquire.Public.SQBarcodeEncoder.TryEncode_Document_OwnerPassword (**SQCommand_Document_OwnerPassword** *command,* out string *value,* out string *caption*) `[protected]`, `[virtual]`

Attempt to encode a Document_OwnerPassword command.

Parameters

command	The command to encode
value	If successful, the encoded barcode text.
caption	If successful, the encoded caption text.

Returns

True on success, false on failure.

5.44.3.5 virtual bool Scanquire.Public.SQBarcodeEncoder.TryEncode_Document_Password (SQCommand↩ _Document_Password *command,* out string *value,* out string *caption*) [protected], [virtual]

Attempt to encode a Document_Password command.

Parameters

command	The command to encode
value	If successful, the encoded barcode text.
caption	If successful, the encoded caption text.

Returns

True on success, false on failure.

5.44.3.6 virtual bool Scanquire.Public.SQBarcodeEncoder.TryEncode_Document_UserPassword (SQCommand↩ _Document_UserPassword *command,* out string *value,* out string *caption*) [protected], [virtual]

Attempt to encode a Document_UserPassword command.

Parameters

command	The command to encode
value	If successful, the encoded barcode text.
caption	If successful, the encoded caption text.

Returns

True on success, false on failure.

5.44.3.7 virtual bool Scanquire.Public.SQBarcodeEncoder.TryEncode_Page_Bookmark (SQCommand_Page_Bookmark *command,* out string *value,* out string *caption*) [protected], [virtual]

Attempt to encode a Page_Bookmark command.

Parameters

command	The command to encode
value	If successful, the encoded barcode text.
caption	If successful, the encoded caption text.

Returns

True on success, false on failure.

5.44.3.8 virtual bool Scanquire.Public.SQBarcodeEncoder.TryEncode_Page_Exclude (SQCommand_Page_Exclude *command,* out string *value,* out string *caption*) [protected], [virtual]

Attempt to encode a Page_Exclude command.

Parameters

command	The command to encode
value	If successful, the encoded barcode text.
caption	If successful, the encoded caption text.

Returns

True on success, false on failure.

5.44.3.9 virtual bool Scanquire.Public.SQBarcodeEncoder.TryEncode_TerminateDocument (SQCommand↩ _TerminateDocument *command,* out string *value,* out string *caption*) [protected], [virtual]

Attempt to encode a TerminateDocument command.

Parameters

command	The command to encode
value	If successful, the encoded barcode text.
caption	If successful, the encoded caption text.

Returns

True on success, false on failure.

5.44.3.10 virtual bool Scanquire.Public.SQBarcodeEncoder.TryEncodeCustom (ISQCommand *command,* out string *value,* out string *caption*) [protected], [virtual]

Placeholder for sub-classes to handle encoding of commands not accounted for.

Parameters

command	The command to encode
value	If successful, the encoded barcode text.
caption	If successful, the encoded caption text.

Returns

True on success, false on failure.

5.44.3.11 bool Scanquire.Public.SQBarcodeEncoder.TryParse (string *barcodeText,* out int *commandID,* out string *value*)

Attempt to separate an encoded barcode into a commnandID and encoded value string

Parameters

barcodeText	The barcode string to decode.
commandID	If successful, the comand id specified by the barcode text.
value	If sucessful, the encoded value specified by the barcode text.

Returns

True on success, false on failure.

5.44.3.12 virtual bool Scanquire.Public.SQBarcodeEncoder.TryParse (string *barcodeText,* out IEnumerable< **ISQCommand** > *commands*) [virtual]

Attempt to decode an encoded barcode string into a series of ISQCommands

Parameters

barcodeText	The encoded barcode text to decode.
commands	If successful, the decoded commands.

Returns

True on success, false on failure.

Implements Scanquire.Public.ISQBarcodeEncoder.

5.44.3.13 virtual bool Scanquire.Public.SQBarcodeEncoder.TryParse_Document_IndexField (string *barcodeValue,* out IEnumerable< **ISQCommand** > *commands*) [protected], [virtual]

Attempt to decode a Document_IndexField command

Parameters

barcodeValue	Encoded value portion of a barcode text.
commands	If successful, list of commands decoded from the barcode

Returns

True on success, False on failure.

5.44.3.14 virtual bool Scanquire.Public.SQBarcodeEncoder.TryParse_Document_OwnerPassword (string *barcodeValue,* out IEnumerable< **ISQCommand** > *commands*) [protected], [virtual]

Attempt to decode a Document_OwnerPassword command.

Parameters

barcodeValue	Encoded value portion of a barcode text.
commands	If successful, list of commands decoded from the barcode

Returns

True on success, False on failure.

5.44.3.15 virtual bool Scanquire.Public.SQBarcodeEncoder.TryParse_Document_Password (string *barcodeValue,* out IEnumerable< **ISQCommand** > *commands*) [protected], [virtual]

Attempt to decode a Document_Password command.

Parameters

barcodeValue	Encoded value portion of a barcode text.
commands	If successful, list of commands decoded from the barcode

Returns

True on success, False on failure.

5.44.3.16 virtual bool Scanquire.Public.SQBarcodeEncoder.TryParse_Document_UserPassword (string *barcodeValue,* out IEnumerable< **ISQCommand** > *commands*) [protected], [virtual]

Attempt to decode a Document_UserPassword command.

Parameters

barcodeValue	Encoded value portion of a barcode text.
commands	If successful, list of commands decoded from the barcode

Returns

True on success, False on failure.

5.44.3.17 virtual bool Scanquire.Public.SQBarcodeEncoder.TryParse_Page_Bookmark (string *barcodeValue,* out IEnumerable< **ISQCommand** > *commands*) [protected], [virtual]

Attempt to decode a Page_Bookmark command.

Parameters

barcodeValue	Encoded value portion of a barcode text.
commands	If successful, list of commands decoded from the barcode

Returns

True on success, False on failure.

5.44.3.18 virtual bool Scanquire.Public.SQBarcodeEncoder.TryParseCustom (int *commandId,* string *barcodeValue,* out IEnumerable< **ISQCommand** > *commands*) [protected], [virtual]

Placeholder for subclasses to handle unacounted for decoding.

Parameters

commandId	Decoded command id.
barcodeValue	Encoded value string.
commands	If successful, list of commands decoded from the barcode.

Returns

True on success, False on failure.

5.44.3.19 virtual bool Scanquire.Public.SQBarcodeEncoder.TryParseCustom (string *barcodeText,* out IEnumerable< **ISQCommand** > *commands*) [protected], [virtual]

Placeholder for subclasses to handle unacounted for decoding.

Parameters

barcodeText	Encoded barcode string.
commands	If successful, list of commands decoded from the barcode.

Returns

True on success, False on failure.

5.44.4 Property Documentation

5.44.4.1 int Scanquire.Public.SQBarcodeEncoder.CommandIdPadding [get], [set]

Number of characters to pad the command id to.

5.44.4.2 string Scanquire.Public.SQBarcodeEncoder.CommandPrefix [get], [set]

Prefix to apply to barcode values to differentiate between barcode sources.

5.44.4.3 char Scanquire.Public.SQBarcodeEncoder.IndexFieldSeparator [get], [set]

For index field commands, the character to apply between the field namd and field value.

5.44.4.4 char Scanquire.Public.SQBarcodeEncoder.ValueSeparator [get], [set]

Character separator to apply between the commandid and the command value.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQBarcodeEncoder/SQBarcodeEncoder.cs

5.45 Scanquire.Public.SQBarcodeEncoders Class Reference

Singleton collection of available SQBarcodeEncoders

Inheritance diagram for Scanquire.Public.SQBarcodeEncoders:

Properties

- override string **DirectoryPath** [get]
- static SQBarcodeEncoders **Instance** [get]

Additional Inherited Members

5.45.1 Detailed Description

Singleton collection of available SQBarcodeEncoders

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQBarcodeEncoder/SQBarcodeEncoders.cs

5.46 Scanquire.Public.SQBarcodeReader Class Reference

Default implementation of an ISQBarcodeReader

Inheritance diagram for Scanquire.Public.SQBarcodeReader:

Public Member Functions

- override Task< Barcode[ ]> Read (SQImage image, CancellationToken cToken)

  Read all barcodes from an image.

Properties

- BarcodeReaderEx **BarcodeReader** `[get, set]`
- SQImageRegion ScanRegion `[get, set]`

  Region of the image to scan.

5.46.1 Detailed Description

Default implementation of an ISQBarcodeReader

5.46.2 Member Function Documentation

5.46.2.1 override Task<Barcode[ ]> Scanquire.Public.SQBarcodeReader.Read (**SQImage** *image,* CancellationToken *cToken*) `[virtual]`

Read all barcodes from an image.

Parameters

image	The image to read.

Returns

List of barcodes read from the specified image

Implements Scanquire.Public.SQBarcodeReaderBase.

5.46.3 Property Documentation

5.46.3.1 **SQImageRegion** Scanquire.Public.SQBarcodeReader.ScanRegion `[get], [set]`

Region of the image to scan.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQBarcodeReader/SQBarcodeReader.cs

5.47 Scanquire.Public.SQBarcodeReaderBase Class Reference

Inheritance diagram for Scanquire.Public.SQBarcodeReaderBase:

Public Member Functions

- abstract Task< Barcode[ ]> Read (SQImage image, CancellationToken cToken)

  Read all barcodes from an image.
- virtual Task< string[ ]> ReadText (SQImage image, CancellationToken cToken)

  Read all barcodes from an image.

5.47.1 Member Function Documentation

5.47.1.1 abstract Task<Barcode[ ]> Scanquire.Public.SQBarcodeReaderBase.Read (**SQImage** *image,* CancellationToken *cToken*) [pure virtual]

Read all barcodes from an image.

Parameters

image	The image to read.

Returns

List of barcodes read from the specified image

Implements Scanquire.Public.ISQBarcodeReader.

Implemented in Scanquire.Public.SQBarcodeReader.

5.47.1.2 virtual Task<string[ ]> Scanquire.Public.SQBarcodeReaderBase.ReadText (**SQImage** *image,* CancellationToken *cToken*) [virtual]

Read all barcodes from an image.

Parameters

image	The image to read.

Returns

List of barcodes strings from the specified image

Implements Scanquire.Public.ISQBarcodeReader.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQBarcodeReader/ISQBarcodeReader.cs

5.48 Scanquire.Public.SQBarcodeReaders Class Reference

Singleton collection of all avilable ISQBarcodeReaders

Inheritance diagram for Scanquire.Public.SQBarcodeReaders:

Properties

- override string **DirectoryPath** [get]
- static SQBarcodeReaders **Instance** [get]

Additional Inherited Members

5.48.1 Detailed Description

Singleton collection of all avilable ISQBarcodeReaders

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQBarcodeReader/SQBarcodeReaders.cs

5.49 Scanquire.Public.SQCommand_Document_IndexField Class Reference

A command to specify the value of a named metadata field to apply to a document.

Inheritance diagram for Scanquire.Public.SQCommand_Document_IndexField:

Public Member Functions

- **SQCommand_Document_IndexField** (string name, object value)

Properties

- string Name [get]

  The name of the metadata field associated with the document.
- object Value [get]

  The value associated with the named metadata field.

5.49.1 Detailed Description

A command to specify the value of a named metadata field to apply to a document.

5.49.2 Property Documentation

5.49.2.1 string Scanquire.Public.SQCommand_Document_IndexField.Name [get]

The name of the metadata field associated with the document.

5.49.2.2 object Scanquire.Public.SQCommand_Document_IndexField.Value [get]

The value associated with the named metadata field.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/SQCommand_Document_IndexField.cs

5.50 Scanquire.Public.SQCommand_Document_OwnerPassword Class Reference

A command to specify the owner password of a document.

Inheritance diagram for Scanquire.Public.SQCommand_Document_OwnerPassword:

Public Member Functions

- **SQCommand_Document_OwnerPassword** (string password)

Properties

- string **Password** `[get]`

5.50.1 Detailed Description

A command to specify the owner password of a document.

For document types only needing a single password, use the more generic command type.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/SQCommand_Document_OwnerPassword.cs

5.51 Scanquire.Public.SQCommand_Document_Password Class Reference

A command to specify the password of a document.

Inheritance diagram for Scanquire.Public.SQCommand_Document_Password:

Public Member Functions

- **SQCommand_Document_Password** (string password)

Properties

- string **Password** `[get]`

5.51.1 Detailed Description

A command to specify the password of a document.

For document types using owner/user password combinations use the more specific commands.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/SQCommand_Document_Password.cs

5.52 Scanquire.Public.SQCommand_Document_UserPassword Class Reference

A command to specify the user password of a document.

Inheritance diagram for Scanquire.Public.SQCommand_Document_UserPassword:

Public Member Functions

- **SQCommand_Document_UserPassword** (string password)

Properties

- string **Password** `[get]`

5.52.1 Detailed Description

A command to specify the user password of a document.

For document types only needing a single password, use the more generic command type.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/SQCommand_Document_UserPassword.cs

5.53 Scanquire.Public.SQCommand_Image_DrawBarcode Class Reference

A command requesting a barcode to be drawn on the specified image.

Inheritance diagram for Scanquire.Public.SQCommand_Image_DrawBarcode:

Public Member Functions

- **SQCommand_Image_DrawBarcode** (string value, string caption)

Properties

- string Value `[get]`

  The encoded barcode text to apply.
- string Caption `[get]`

  The caption to apply to the drawn barcode.

5.53.1 Detailed Description

A command requesting a barcode to be drawn on the specified image.

5.53.2 Property Documentation

5.53.2.1 string Scanquire.Public.SQCommand_Image_DrawBarcode.Caption [get]

The caption to apply to the drawn barcode.

5.53.2.2 string Scanquire.Public.SQCommand_Image_DrawBarcode.Value [get]

The encoded barcode text to apply.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/SQCommand_Image_DrawBarcode.cs

5.54 Scanquire.Public.SQCommand_Image_DrawImage Class Reference

A command to specify that an image be overlain on the specified image.

Inheritance diagram for Scanquire.Public.SQCommand_Image_DrawImage:

Public Member Functions

- **SQCommand_Image_DrawImage** (SQImage image)

Properties

- SQImage **Image** [get, set]

5.54.1 Detailed Description

A command to specify that an image be overlain on the specified image.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/SQCommand_Image_DrawImage.cs

5.55 Scanquire.Public.SQCommand_Image_DrawText Class Reference

A command specifying a text value to draw onto a specified image.

Inheritance diagram for Scanquire.Public.SQCommand_Image_DrawText:

Public Member Functions

- **SQCommand_Image_DrawText** (string text, Font font=null, Brush brush=null)

Properties

- string Text [get]

  The text value to draw.
- Brush Brush [get]

  The brush to use when drawing the text value.
- Font Font [get]

  The font to use when drawing the text value.

5.55.1 Detailed Description

A command specifying a text value to draw onto a specified image.

5.55.2 Property Documentation

5.55.2.1 Brush Scanquire.Public.SQCommand_Image_DrawText.Brush [get]

The brush to use when drawing the text value.

5.55.2.2 Font Scanquire.Public.SQCommand_Image_DrawText.Font [get]

The font to use when drawing the text value.

5.55.2.3 string Scanquire.Public.SQCommand_Image_DrawText.Text [get]

The text value to draw.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/SQCommand_Image_DrawText.cs

5.56 Scanquire.Public.SQCommand_Image_Size Class Reference

A command used to specify an image's drawn size.

Inheritance diagram for Scanquire.Public.SQCommand_Image_Size:

Public Member Functions

- **SQCommand_Image_Size** (SizeF size)
- **SQCommand_Image_Size** (float width, float height)

Properties

- SizeF SizeF [get]

  Size (in points)
- float Width [get]

  Width (in points)
- float Height [get]

  Height (in points)

5.56.1 Detailed Description

A command used to specify an image's drawn size.

5.56.2 Property Documentation

5.56.2.1 float Scanquire.Public.SQCommand_Image_Size.Height [get]

Height (in points)

5.56.2.2 SizeF Scanquire.Public.SQCommand_Image_Size.SizeF [get]

Size (in points)

5.56.2.3 float Scanquire.Public.SQCommand_Image_Size.Width [get]

Width (in points)

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/SQCommand_Image_Size.cs

5.57 Scanquire.Public.SQCommand_Page_Bookmark Class Reference

For document types that support bookmarks, specify a bookmark be applied to the specified page.

Inheritance diagram for Scanquire.Public.SQCommand_Page_Bookmark:

Public Member Functions

- **SQCommand_Page_Bookmark** (string title)

Properties

- string Title [get]

  Title of the bookmark to apply to the specified page.

5.57.1 Detailed Description

For document types that support bookmarks, specify a bookmark be applied to the specified page.

5.57.2 Property Documentation

5.57.2.1 string Scanquire.Public.SQCommand_Page_Bookmark.Title [get]

Title of the bookmark to apply to the specified page.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/SQCommand_Page_Bookmark.cs

5.58 Scanquire.Public.SQCommand_Page_Exclude Class Reference

Command specifying that the current page should be excluded from the final document.

Inheritance diagram for Scanquire.Public.SQCommand_Page_Exclude:

5.58.1 Detailed Description

Command specifying that the current page should be excluded from the final document.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/SQCommand_Page_Exclude.cs

5.59 Scanquire.Public.SQCommand_TerminateDocument Class Reference

Command specifying that the end of the current document has been reached.

Inheritance diagram for Scanquire.Public.SQCommand_TerminateDocument:

5.59.1 Detailed Description

Command specifying that the end of the current document has been reached.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommand/SQCommand_TerminateDocument.cs

5.60 Scanquire.Public.SQCommandImageBuilder Class Reference

SQCommandImageBuilder that prompts the user to enter a header, footer, comments and bookmarks and applys them to the page.

Inheritance diagram for Scanquire.Public.SQCommandImageBuilder:

Classes

- class ImageBuilderInputDialog

Public Member Functions

- override IEnumerable< Task< SQImage > > **Build** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)

Protected Member Functions

- virtual string InterpolateString (string value)

  Map any keywords to there defined values.

Protected Attributes

- ImageBuilderInputDialog **InputDialog** = new ImageBuilderInputDialog()

Properties

- string BarcodeEncoderName `[get, set]`

  Identifier of the barcode encoder to use for any commands requiring barcodes.
- ISQBarcodeEncoder BarcodeEncoder `[get, set]`

  ISQBarcodeEncoder defined by BarcodeEncoderName
- string[] Bookmarks `[get, set]`

  List of all bookmark titles to apply to the command image.
- Font DefaultTextFont `[get, set]`

  Default font to apply to the image for drawing text.
- Brush DefaultTextBrush `[get, set]`

  Default brush to apply when drawing text.
- string Header `[get, set]`

  Text to apply to the header section of the image
- string Footer `[get, set]`

  Text to apply to the footer section of the image.
- string LogoPath `[get, set]`

  Path to the log file (if any) to apply to the image.
- SizeF LogoSize `[get, set]`

  Size of the logo to apply to the image.
- bool PromptForComments `[get, set]`

  Specify wether to ask the user for input prior to generation.

5.60.1 Detailed Description

SQCommandImageBuilder that prompts the user to enter a header, footer, comments and bookmarks and applys them to the page.

5.60.2 Member Function Documentation

5.60.2.1 virtual string Scanquire.Public.SQCommandImageBuilder.InterpolateString (string *value*) `[protected]`, `[virtual]`

Map any keywords to there defined values.

5.60.3 Property Documentation

5.60.3.1 ISQBarcodeEncoder Scanquire.Public.SQCommandImageBuilder.BarcodeEncoder `[get]`, `[set]`

ISQBarcodeEncoder defined by BarcodeEncoderName

5.60.3.2 string Scanquire.Public.SQCommandImageBuilder.BarcodeEncoderName `[get]`, `[set]`

Identifier of the barcode encoder to use for any commands requiring barcodes.

5.60.3.3 string [ ] Scanquire.Public.SQCommandImageBuilder.Bookmarks `[get], [set]`

List of all bookmark titles to apply to the command image.

5.60.3.4 Brush Scanquire.Public.SQCommandImageBuilder.DefaultTextBrush `[get], [set]`

Default brush to apply when drawing text.

5.60.3.5 Font Scanquire.Public.SQCommandImageBuilder.DefaultTextFont `[get], [set]`

Default font to apply to the image for drawing text.

5.60.3.6 string Scanquire.Public.SQCommandImageBuilder.Footer `[get], [set]`

Text to apply to the footer section of the image.

5.60.3.7 string Scanquire.Public.SQCommandImageBuilder.Header `[get], [set]`

Text to apply to the header section of the image

5.60.3.8 string Scanquire.Public.SQCommandImageBuilder.LogoPath `[get], [set]`

Path to the log file (if any) to apply to the image.

5.60.3.9 SizeF Scanquire.Public.SQCommandImageBuilder.LogoSize `[get], [set]`

Size of the logo to apply to the image.

5.60.3.10 bool Scanquire.Public.SQCommandImageBuilder.PromptForComments `[get], [set]`

Specify wether to ask the user for input prior to generation.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommandImageBuilder/SQCommandImageBuilder.cs

5.61 Scanquire.Public.SQCommandImageBuilderBase Class Reference

Inheritance diagram for Scanquire.Public.SQCommandImageBuilderBase:

Public Member Functions

- virtual Task< SQImage > Build (IEnumerable< ISQCommand > commands, IProgress< ProgressEvent↵Args > progress, CancellationToken cToken)

  Compile the provided commands into a single image.
- virtual IEnumerable< Task< SQImage > > Build (IEnumerable< IEnumerable< ISQCommand_Image >> commandLists, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

  Generate a series of SQImages based on a series of command lists.

- abstract IEnumerable< Task< SQImage > > **Build** (IProgress< ProgressEventArgs > progress, CancellationToken cToken)

Protected Member Functions

- virtual Bitmap GenerateImageImage (SQCommand_Image_DrawImage command)

  Generate a sub-image containing the specified image.
- virtual Bitmap GenerateTextImage (SQCommand_Image_DrawText command)

  Generate a sub-image containing the specified text.
- virtual Bitmap GenerateBarcodeImage (SQCommand_Image_DrawBarcode command)

  Generate a sub-image containing a barcode.

Properties

- int **ImageDPI** `[get, set]`
- float ImageWidth `[get]`

  Width of the produced image (in points)
- float MinImageHeight `[get, set]`

  The minimum height for a produced image (in points)
- Padding ImageMargin `[get, set]`

  Margin (in points) for the whole image.
- int ElementSpacing `[get, set]`

  Vertical spacing (in points) between individual content elements.
- string BarcodeRendererName `[get, set]`

  ID of the barcode renderer to use for drawing any barcodes
- Code128BarcodeRenderer BarcodeRenderer `[get, set]`

  Code128BarcodeRenderer specified by BarcodeRendererName.

5.61.1 Member Function Documentation

5.61.1.1 virtual Task<**SQImage**> Scanquire.Public.SQCommandImageBuilderBase.Build (IEnumerable< **ISQCommand** > *commands*, IProgress< ProgressEventArgs > *progress*, CancellationToken *cToken*) `[virtual]`

Compile the provided commands into a single image.

5.61.1.2 virtual IEnumerable<Task<**SQImage**> > Scanquire.Public.SQCommandImageBuilderBase.Build (IEnumerable< IEnumerable< **ISQCommand_Image** >> *commandLists*, IProgress< ProgressEventArgs > *progress*, CancellationToken *cToken*) `[virtual]`

Generate a series of SQImages based on a series of command lists.

Returns

A compiled image for each command list provided.

5.61.1.3 virtual Bitmap Scanquire.Public.SQCommandImageBuilderBase.GenerateBarcodeImage (**SQCommand_Image_DrawBarcode** *command*) `[protected], [virtual]`

Generate a sub-image containing a barcode.

5.61.1.4 virtual Bitmap Scanquire.Public.SQCommandImageBuilderBase.GenerateImageImage (SQCommand_Image_DrawImage *command*) [protected], [virtual]

Generate a sub-image containing the specified image.

5.61.1.5 virtual Bitmap Scanquire.Public.SQCommandImageBuilderBase.GenerateTextImage (SQCommand_Image_DrawText *command*) [protected], [virtual]

Generate a sub-image containing the specified text.

5.61.2 Property Documentation

5.61.2.1 Code128BarcodeRenderer Scanquire.Public.SQCommandImageBuilderBase.BarcodeRenderer [get], [set]

Code128BarcodeRenderer specified by BarcodeRendererName.

5.61.2.2 string Scanquire.Public.SQCommandImageBuilderBase.BarcodeRendererName [get], [set]

ID of the barcode renderer to use for drawing any barcodes

5.61.2.3 int Scanquire.Public.SQCommandImageBuilderBase.ElementSpacing [get], [set]

Vertical spacing (in points) between individual content elements.

5.61.2.4 Padding Scanquire.Public.SQCommandImageBuilderBase.ImageMargin [get], [set]

Margin (in points) for the whole image.

5.61.2.5 float Scanquire.Public.SQCommandImageBuilderBase.ImageWidth [get]

Width of the produced image (in points)

5.61.2.6 float Scanquire.Public.SQCommandImageBuilderBase.MinImageHeight [get], [set]

The minimum height for a produced image (in points)

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommandImageBuilder/SQCommandImageBuilderBase.cs

5.62 Scanquire.Public.SQCommandImageBuilders Class Reference

Inheritance diagram for Scanquire.Public.SQCommandImageBuilders:

Properties

- override string **DirectoryPath** [get]
- static SQCommandImageBuilders **Instance** [get]

Additional Inherited Members

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommandImageBuilder/SQCommandImageBuilders.cs

5.63 Scanquire.Public.SQCommandReader Class Reference

An empty command reader

Inheritance diagram for Scanquire.Public.SQCommandReader:

Public Member Functions

- Task< IList< ISQCommand > > **Read** (SQImage image, int documentNumber, int pageNumber, CancellationToken cToken)

5.63.1 Detailed Description

An empty command reader

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommandReader/SQCommandReader.cs

5.64 Scanquire.Public.SQCommandReader_Barcode Class Reference

ISQCommandReader used for processing barcode base command images.

Inheritance diagram for Scanquire.Public.SQCommandReader_Barcode:

Public Member Functions

- Task< IList< ISQCommand > > **Read** (SQImage image, int documentNumber, int pageNumber, CancellationToken cToken)

Properties

- virtual string BarcodeEncoderName `[get, set]`

  Name of the ISQBarcodeEncoder to use for encoding/decoding.
- ISQBarcodeEncoder BarcodeEncoder `[get, set]`

  ISQBarcodeEncoder specified by BarcodeEncoderName
- virtual string BarcodeReaderName `[get, set]`

  Name of the ISQBarcodeReader to read the barcode values from the image.
- ISQBarcodeReader BarcodeReader `[get, set]`

  ISQBarcodeReader specified by BarcodeReaderName

5.64.1 Detailed Description

ISQCommandReader used for processing barcode base command images.

5.64.2 Property Documentation

5.64.2.1 ISQBarcodeEncoder Scanquire.Public.SQCommandReader_Barcode.BarcodeEncoder [get], [set]

ISQBarcodeEncoder specified by BarcodeEncoderName

5.64.2.2 virtual string Scanquire.Public.SQCommandReader_Barcode.BarcodeEncoderName [get], [set]

Name of the ISQBarcodeEncoder to use for encoding/decoding.

5.64.2.3 ISQBarcodeReader Scanquire.Public.SQCommandReader_Barcode.BarcodeReader [get], [set]

ISQBarcodeReader specified by BarcodeReaderName

5.64.2.4 virtual string Scanquire.Public.SQCommandReader_Barcode.BarcodeReaderName [get], [set]

Name of the ISQBarcodeReader to read the barcode values from the image.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommandReader/SQCommandReader_Barcode.cs

5.65 Scanquire.Public.SQCommandReaders Class Reference

Singleton collection of available ISQCommandReaders

Inheritance diagram for Scanquire.Public.SQCommandReaders:

Properties

- override string **DirectoryPath** [get]
- static SQCommandReaders **Instance** [get]

Additional Inherited Members

5.65.1 Detailed Description

Singleton collection of available ISQCommandReaders

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQCommandReader/SQCommandReaders.cs

5.66 Scanquire.Public.SQDocument Class Reference

A collection of SQPages and ISQCommands that apply to the document level.

Public Member Functions

- **SQDocument** (IEnumerable< ISQCommand_Document > commands, IEnumerable< SQPage > pages)
- **SQDocument** (IEnumerable< ISQCommand_Document > commands)
- **SQDocument** (IEnumerable< SQPage > pages)

Properties

- List< ISQCommand_Document > **Commands** `[get]`
- List< SQPage > **Pages** `[get]`

5.66.1 Detailed Description

A collection of SQPages and ISQCommands that apply to the document level.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQDocument/SQDocument.cs

5.67 Scanquire.Public.SQFile Class Reference

Description of SQFile.

Inheritance diagram for Scanquire.Public.SQFile:

Static Public Member Functions

- static SQFile FromFile (string path)

  Create a new instance from a local file.

Properties

- string **MimeType** `[get, set]`
- List< ISQCommand_Document > Commands `[get, set]`

  List of document based commands applied to the file.
- Dictionary< string, object > IndexFields `[get, set]`

  Metadata properties of the file.
- int PageCount `[get, set]`

  Number of pages in the file (optional).
- byte[ ] Data `[get, set]`

  Binary contents of the file.
- string Checksum `[get]`

  MD5 checksum of the Data.
- string FileExtension `[get, set]`

  Windows file extension.

5.67.1 Detailed Description

Description of SQFile.

5.67.2 Member Function Documentation

5.67.2.1 static SQFile Scanquire.Public.SQFile.FromFile (string *path*) `[static]`

Create a new instance from a local file.

5.67.3 Property Documentation

5.67.3.1 string Scanquire.Public.SQFile.Checksum `[get]`

MD5 checksum of the Data.

5.67.3.2 List<ISQCommand_Document> Scanquire.Public.SQFile.Commands `[get]`, `[set]`

List of document based commands applied to the file.

5.67.3.3 byte [ ] Scanquire.Public.SQFile.Data `[get]`, `[set]`

Binary contents of the file.

5.67.3.4 string Scanquire.Public.SQFile.FileExtension `[get]`, `[set]`

Windows file extension.

Accepts both dotted and undotted extensions ("pdf or .pdf")

5.67.3.5 Dictionary<string, object> Scanquire.Public.SQFile.IndexFields `[get]`, `[set]`

Metadata properties of the file.

5.67.3.6 int Scanquire.Public.SQFile.PageCount `[get]`, `[set]`

Number of pages in the file (optional).

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFile/SQFile.cs

5.68 Scanquire.Public.SQFile_Pdf Class Reference

Specialized type of SQFile for PDF files.

Inheritance diagram for Scanquire.Public.SQFile_Pdf:

Properties

- string **OwnerPassword** `[get, set]`
- string **UserPassword** `[get, set]`

Additional Inherited Members

5.68.1 Detailed Description

Specialized type of SQFile for PDF files.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFile/SQFile_Pdf.cs

5.69 Scanquire.Public.SQFileReader_Image Class Reference

SQFileReader for reading single or multi-page image formats (jpeg, tiff, etc).

Inheritance diagram for Scanquire.Public.SQFileReader_Image:

Public Member Functions

- override IEnumerable< Task< SQImage > > **Read** (SQFile file, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

5.69.1 Detailed Description

SQFileReader for reading single or multi-page image formats (jpeg, tiff, etc).

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFileReader/SQFileReader_Image.cs

5.70 Scanquire.Public.SQFileReader_Pdf Class Reference

Specialized ISQFileReader for reading PDF files.

Inheritance diagram for Scanquire.Public.SQFileReader_Pdf:

Public Member Functions

- void **AddPassword** (string password)
- virtual IEnumerable< Task< SQImage > > **Read** (SQFile file, IEnumerable< string > passwords, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- override IEnumerable< Task< SQImage > > **Read** (SQFile file, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

Properties

- List< string > OwnerPasswords `[get]`

  List of possible owner passwords to apply to the document upon opening.

5.70.1 Detailed Description

Specialized ISQFileReader for reading PDF files.

5.70.2 Property Documentation

5.70.2.1 List< string > Scanquire.Public.SQFileReader_Pdf.OwnerPasswords [get]

List of possible owner passwords to apply to the document upon opening.

Currently a placeholder, reader does not handle decryption at this time.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFileReader/SQFileReader_Pdf.cs

5.71 Scanquire.Public.SQFileReaderBase Class Reference

Inheritance diagram for Scanquire.Public.SQFileReaderBase:

Public Member Functions

- abstract IEnumerable< Task< SQImage > > **Read** (SQFile file, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQImage > > **Read** (string path, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual Task< SQImage[ ]> **ReadAll** (SQFile file, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual Task< SQImage[ ]> **ReadAll** (string path, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFileReader/SQFileReaderBase.cs

5.72 Scanquire.Public.SQFileReaders Class Reference

Singleton collection of available ISQFileReaders

Inheritance diagram for Scanquire.Public.SQFileReaders:

Static Public Member Functions

- static string FileExtensionToReaderName (string ext)

  Map a file extension to the name of an ISQFileReader.
- static IEnumerable< Task< SQImage > > **Read** (SQFile file, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- static IEnumerable< Task< SQImage > > **Read** (string path, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

- static Task< SQImage[ ]> **ReadAll** (SQFile file, IProgress< ProgressEventArgs > progress, Cancellation←↩ Token cToken)
- static Task< SQImage[ ]> **ReadAll** (string path, IProgress< ProgressEventArgs > progress, Cancellation←↩ Token cToken)

Properties

- override string **DirectoryPath** `[get]`
- static SQFileReaders **Instance** `[get]`

Additional Inherited Members

5.72.1 Detailed Description

Singleton collection of available ISQFileReaders

5.72.2 Member Function Documentation

5.72.2.1 static string Scanquire.Public.SQFileReaders.FileExtensionToReaderName (string *ext*) `[static]`

Map a file extension to the name of an ISQFileReader.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFileReader/SQFileReaders.cs

5.73 Scanquire.Public.Properties.SQFilesystemArchiver Class Reference

Inheritance diagram for Scanquire.Public.Properties.SQFilesystemArchiver:

Properties

- static SQFilesystemArchiver **Default** `[get]`
- string **DefaultRootPath** `[get]`

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Properties/SQFilesystemArchiver.Designer.cs

5.74 Scanquire.Public.SQFilesystemArchiver Class Reference

A basic archiver for accessing a local or NAS file system.

Inheritance diagram for Scanquire.Public.SQFilesystemArchiver:

Public Member Functions

- override IEnumerable< Task< SQImage > > AcquireFromFile (IProgress< ProgressEventArgs > progress, CancellationToken cToken)

  Acquiring from file depends on the type of archive, so defer to custom archiver implementation
- override async Task Send (SQFile file, int fileNumber, IProgress< ProgressEventArgs > progress, CancellationToken cancelToken)

  Sends a single file (encoded document) to the archiver.

Properties

- SQFilesystemConnector **FilesystemConnector** `[get, set]`
- FileDictionaryLoggerBase Log `[get, set]`

  Log to use when writing metadata about saved files.

Additional Inherited Members

5.74.1 Detailed Description

A basic archiver for accessing a local or NAS file system.

5.74.2 Member Function Documentation

5.74.2.1 override IEnumerable<Task<SQImage> > Scanquire.Public.SQFilesystemArchiver.AcquireFromFile (IProgress< ProgressEventArgs > *progress*, CancellationToken *cToken*) `[virtual]`

Acquiring from file depends on the type of archive, so defer to custom archiver implementation

Implements Scanquire.Public.SQArchiverBase.

5.74.2.2 override async Task Scanquire.Public.SQFilesystemArchiver.Send (SQFile *file*, int *fileNumber*, IProgress< ProgressEventArgs > *progress*, CancellationToken *cToken*) `[virtual]`

Sends a single file (encoded document) to the archiver.

Parameters

file	The file to archive.
fileNumber	For sending multiple files, the index of the file in the overall listing.
progress	
cToken	

Returns

Implements Scanquire.Public.SQArchiverBase.

5.74.3 Property Documentation

5.74.3.1 FileDictionaryLoggerBase Scanquire.Public.SQFilesystemArchiver.Log `[get], [set]`

Log to use when writing metadata about saved files.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQArchiver/SQFilesystemArchiver.cs

5.75 Scanquire.Public.SQFilesystemConnector Class Reference

Classes

- class SaveFileResult

Public Member Functions

- string[ ] **SelectFilesForOpen** (IWin32Window owner, SynchronizationContext context)
- string[ ] **SelectFilesForOpen** ()
- string[ ] **SelectFilesForOpen** (IWin32Window owner)
- string[ ] **SelectFilesForOpen** (SynchronizationContext context)
- string **SelectFileForOpen** (IWin32Window owner, SynchronizationContext context)
- string **SelectFileForOpen** ()
- string **SelectFileForOpen** (IWin32Window owner)
- string **SelectFileForOpen** (SynchronizationContext context)
- string **SelectFileForSave** (string ext, IWin32Window owner, SynchronizationContext context)
- string **SelectFileForSave** (string ext, IWin32Window owner)
- string **SelectFileForSave** (string ext, SynchronizationContext context)
- string **SelectFileForSave** (string ext)
- string **SelectFileForSave** (IWin32Window owner, SynchronizationContext context)
- string **SelectFileForSave** (IWin32Window owner)
- string **SelectFileForSave** (SynchronizationContext context)
- string **SelectFileForSave** ()
- async Task< SaveFileResult > **SaveFile** (string path, byte[ ] data, ExistingFileAction existingFileAction, I↩ Win32Window owner, SynchronizationContext context)
- async Task< SaveFileResult > **SaveFile** (string path, byte[ ] data, ExistingFileAction existingFileAction, I↩ Win32Window owner)
- async Task< SaveFileResult > **SaveFile** (string path, byte[ ] data, ExistingFileAction existingFileAction, SynchronizationContext context)
- async Task< SaveFileResult > **SaveFile** (string path, byte[ ] data, ExistingFileAction existingFileAction)
- async Task< SaveFileResult > **SaveFile** (string path, byte[ ] data, IWin32Window owner, Synchronization↩ Context context)
- async Task< SaveFileResult > **SaveFile** (string path, byte[ ] data, IWin32Window owner)
- async Task< SaveFileResult > **SaveFile** (string path, byte[ ] data, SynchronizationContext context)
- async Task< SaveFileResult > **SaveFile** (string path, byte[ ] data)
- bool TryParse (string path, out string versionText, out int versionNumber)

  Try to parse the version information out of the provided path.
- bool TryParse (string path, out string versionText)

  Try to parse the version information out of the provided path.
- bool TryParse (string path, out int versionNumber)

  Try to parse the version information out of the provided path.
- string RemoveVersion (string path)

  Parameters

path	

  Returns

  The provided path with all (if any) version info removed.
- string GenerateVersionText (int versionNumber)

  Generate a version string from a version number
- string AddOrChangeVersion (string path, int versionNumber)

Parameters

path	*Path (with or without version info) of the path to add version info to.*
versionNumber	

Returns

The provided path with the provided version number.

- bool ContainsVersion (string path)

  Returns

  True if the provided path contains version information, false otherwise.

- string GetPathForNewVersion (string path)

  Parameters

path	*Path (with or without version info) of the file to create the new version for.*

  Returns

  Absolute path to a new version.

- bool TryGetLatestVersion (string path, out string latestVersionPath, out int latestVersionNumber)

  Try to get the latest version of a file.

- bool TryGetLatestVersion (string path, out string latestVersionPath)

  Try to get the latest version of a file.

- bool TryGetLatestVersion (string path, out int latestVersionNumber)

  Try to get the latest version of a file.

Static Public Attributes

- static string **DefaultRootPath** = @"C:\Archives\"

Properties

- string RootPath `[get, set]`

  The directory to use as the initial directory for file dialogs, and for building relative paths.

- ExistingFileAction ExistingFileAction `[get, set]`

  Action to take when a specified destination already exists.

- string VersionPrefix `[get, set]`

  Prefix used to denote version information in a path.

- int VersionPadding `[get, set]`

  Minimum width of a version number string.

- bool AddVersionInfoToPrimaryVersion `[get, set]`

  If true, the primary version of a file will contain version information. If false, the primary version of a file will not contain any version information.

- OpenFileDialog **OpenFileDialog** `[get, set]`
- bool **OpenFileDialogRememberLastDirectory** `[get, set]`
- SaveFileDialog **SaveFileDialog** `[get, set]`
- bool **SaveFileDialogRememberLastDirectory** `[get, set]`
- FolderBrowserDialog **FolderBrowserDialog** `[get, set]`

5.75.1 Member Function Documentation

5.75.1.1 string Scanquire.Public.SQFilesystemConnector.GenerateVersionText (int *versionNumber*)

Generate a version string from a version number

Parameters

versionNumber	

Returns

If versionNumber is zero and AddVersionInfoToPrimaryVersion is false, an empty string. Otherwise: A version string formatted [VersionPrefix][#] (ex _v002).

5.75.1.2 string Scanquire.Public.SQFilesystemConnector.GetPathForNewVersion (string *path*)

Parameters

path	Path (with or without version info) of the file to create the new version for.

Returns

Absolute path to a new version.

The returned path is not created.

5.75.1.3 bool Scanquire.Public.SQFilesystemConnector.TryGetLatestVersion (string *path,* out string *latestVersionPath,* out int *latestVersionNumber*)

Try to get the latest version of a file.

Parameters

path	Path (with or without version info) of the file to check.
latestVersionPath	On success, absolute path to the latest version. On fail, null.
latestVersionNumber	On success, version number of the latest version. On fail, -1;

Returns

True if any versions exist, False if no versions found.

5.75.1.4 bool Scanquire.Public.SQFilesystemConnector.TryGetLatestVersion (string *path,* out string *latestVersionPath*)

Try to get the latest version of a file.

Parameters

path	Path (with or without version info) of the file to check.
latestVersionPath	On success, path to the latest version. On fail, null.

Returns

True if any versions exist, False if no versions found.

5.75.1.5 bool Scanquire.Public.SQFilesystemConnector.TryGetLatestVersion (string *path,* out int *latestVersionNumber*)

Try to get the latest version of a file.

Parameters

path	Path (with or without version info) of the file to check.
latestVersion↩ Number	On success, version number of the latest version. On fail, -1;

Returns

True if any versions exist, False if no versions found.

5.75.1.6 bool Scanquire.Public.SQFilesystemConnector.TryParse (string *path,* out string *versionText,* out int *versionNumber*)

Try to parse the version information out of the provided path.

Parameters

path	Path to parse.
versionText	On success, the text part of the version information (including the version prefix). On fail, null.
versionNumber	On success, the numeric version number. On fail, -1.

Returns

True if the path contained version info. False if the path does not contain version info.

5.75.1.7 bool Scanquire.Public.SQFilesystemConnector.TryParse (string *path,* out string *versionText*)

Try to parse the version information out of the provided path.

Parameters

path	Path to parse.
versionText	On success, the text part of the version information (including the version prefix). On fail, null.

Returns

True if the path contained version info. False if the path does not contain version info.

5.75.1.8 bool Scanquire.Public.SQFilesystemConnector.TryParse (string *path,* out int *versionNumber*)

Try to parse the version information out of the provided path.

Parameters

path	Path to parse.
versionNumber	On success, the numeric version number. On fail, -1.

Returns

True if the path contained version info. False if the path does not contain version info.

5.75.2 Property Documentation

5.75.2.1 bool Scanquire.Public.SQFilesystemConnector.AddVersionInfoToPrimaryVersion `[get]`, `[set]`

If true, the primary version of a file will contain version information. If false, the primary version of a file will not contain any version information.

5.75.2.2 ExistingFileAction Scanquire.Public.SQFilesystemConnector.ExistingFileAction [get], [set]

Action to take when a specified destination already exists.

5.75.2.3 string Scanquire.Public.SQFilesystemConnector.RootPath [get], [set]

The directory to use as the initial directory for file dialogs, and for building relative paths.

5.75.2.4 int Scanquire.Public.SQFilesystemConnector.VersionPadding [get], [set]

Minimum width of a version number string.

Ensures correct file sorting. (without padding, version 10 would come before version 2). Example, if VersionPadding is 3 and VersionPrefix is _v, a versioned file name would be foo_v001.ext.

5.75.2.5 string Scanquire.Public.SQFilesystemConnector.VersionPrefix [get], [set]

Prefix used to denote version information in a path.

Example, if VersionPadding is 3 and VersionPrefix is _v, a versioned file name would be foo_v001.ext.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFilesystemConnector/SQFilesystemConnector.cs

5.76 Scanquire.Public.SQFilesystemConnectors Class Reference

Inheritance diagram for Scanquire.Public.SQFilesystemConnectors:

Properties

- override string **DirectoryPath** [get]
- static SQFilesystemConnectors **Instance** [get]

Additional Inherited Members

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFilesystemConnector/SQFilesystemConnectors.cs

5.77 Scanquire.Public.SQFileWriter_Image Class Reference

Inheritance diagram for Scanquire.Public.SQFileWriter_Image:

Public Member Functions

- override Task< SQFile > **Write** (SQDocument document, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

Properties

- FREE_IMAGE_FORMAT **ImageFormat** `[get, set]`
- FREE_IMAGE_SAVE_FLAGS **SaveFlags** `[get, set]`
- string **MimeType** `[get, set]`

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFileWriter/SQFileWriter_Image.cs

5.78 Scanquire.Public.SQFileWriter_Pdf Class Reference

Inheritance diagram for Scanquire.Public.SQFileWriter_Pdf:

Public Member Functions

- override Task< SQFile > **Write** (SQDocument document, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

Protected Member Functions

- void **InitializeDocument** (PDFLibrary pdf, ref string ownerPassword, ref string userPassword)
- void **DrawText** (PDFLibrary pdf, string text)
- void **DrawImage** (PDFLibrary pdf, SQImage image)
- void **WriteBookmarks** (PDFLibrary pdf, IEnumerable< SQCommand_Page_Bookmark > bookmarks)
- void SetPageSize (PDFLibrary pdf, SizeF pageSize)

  Sets the size of the current page.
- void **WritePage** (PDFLibrary pdf, SQPage page)
- byte[ ] **EncodeImage** (SQImage image)

Properties

- override string **FileExtension** `[get, set]`
- PDFLibrary.Encryption **Encryption** `[get, set]`
- PDFLibrary.Permission **Permissions** `[get, set]`
- string **OwnerPassword** `[get, set]`
- string **UserPassword** `[get, set]`
- FREE_IMAGE_FORMAT **MonochromeImageFormat** `[get, set]`
- FREE_IMAGE_SAVE_FLAGS **MonochromeImageSaveFlags** `[get, set]`
- FREE_IMAGE_FORMAT **GreyscaleImageFormat** `[get, set]`
- FREE_IMAGE_SAVE_FLAGS **GreyscaleImageSaveFlags** `[get, set]`
- FREE_IMAGE_FORMAT **ColorImageFormat** `[get, set]`
- FREE_IMAGE_SAVE_FLAGS **ColorImageSaveFlags** `[get, set]`
- bool **ScalePagesToNearestHalfInch** `[get, set]`

5.78.1 Member Function Documentation

5.78.1.1 void Scanquire.Public.SQFileWriter_Pdf.SetPageSize (**PDFLibrary** *pdf,* SizeF *pageSize*) `[protected]`

Sets the size of the current page.

Parameters

pdf	
pageSize	Size of the page (in points)

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFileWriter/SQFileWriter_Pdf.cs

5.79 Scanquire.Public.SQFileWriter_Tiff Class Reference

<credit>http://stackoverflow.com/questions/398388/convert-bitmaps-to-one-multipage-tiff-in

Inheritance diagram for Scanquire.Public.SQFileWriter_Tiff:

Public Member Functions

- override Task< SQFile > **Write** (SQDocument document, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

Protected Member Functions

- Dictionary< string, object > **GetIndexFields** (IEnumerable< SQCommand_Document_IndexField > indexFieldCommands)

Protected Attributes

- const string **TIFF_MIME_TYPE** = "image/tiff"

Static Protected Attributes

- static ImageCodecInfo **TiffImageCodecInfo** = ImageCodecInfo.GetImageEncoders().First(ici => ici.Mime↩ Type == TIFF_MIME_TYPE)

Properties

- EncoderValue **Compression** `[get, set]`
- override string **FileExtension** `[get, set]`

5.79.1 Detailed Description

<credit>http://stackoverflow.com/questions/398388/convert-bitmaps-to-one-multipage-tiff-in

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFileWriter/SQFileWriter_Tiff.cs

5.80 Scanquire.Public.SQFileWriterBase Class Reference

Inheritance diagram for Scanquire.Public.SQFileWriterBase:

Public Member Functions

- abstract Task< SQFile > **Write** (SQDocument document, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual IEnumerable< Task< SQFile > > **Write** (IEnumerable< SQDocument > documents, IProgress< ProgressEventArgs > progress, CancellationToken cToken)
- virtual Task< SQFile[ ]> **WriteAll** (IEnumerable< SQDocument > documents, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

Properties

- virtual string FileExtension `[get, set]`

  Windows file extension for the resulting files.

5.80.1 Property Documentation

5.80.1.1 virtual string Scanquire.Public.SQFileWriterBase.FileExtension `[get]`, `[set]`

Windows file extension for the resulting files.

Automatically prepends the '.'

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFileWriter/SQFileWriterBase.cs

5.81 Scanquire.Public.SQFileWriters Class Reference

Manages all configured ISQFileWriters

Inheritance diagram for Scanquire.Public.SQFileWriters:

Properties

- override string **DirectoryPath** `[get]`
- static SQFileWriters **Instance** `[get]`

Additional Inherited Members

5.81.1 Detailed Description

Manages all configured ISQFileWriters

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQFileWriter/SQFileWriters.cs

5.82 Scanquire.Public.Properties.SQImage Class Reference

Inheritance diagram for Scanquire.Public.Properties.SQImage:

Properties

- static SQImage **Default** `[get]`
- bool **EncryptFiles** `[get]`
- global::System.Drawing.Size **ThumbnailSize** `[get]`

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Properties/SQImage.Designer.cs

5.83 Scanquire.Public.SQImage Class Reference

A single SQImage with revision control.

Inheritance diagram for Scanquire.Public.SQImage:

Public Member Functions

- SQImage ()

  Base constructor, sets up internal event handlers.
- SQImage (FreeImageBitmap image, out SQImageEditLock editLock)

  Creates a new instance and leaves the image open for editing
- SQImage (FreeImageBitmap image)
- SQImageEditLock BeginEdit ()

  Begins an edit operation. If WorkingCopy is not loaded, the latest revision will be loaded as the WorkingCopy.
- void Save (bool overwritePreviousRevision)

  Save the current WorkingCopy as a revision.
- void Save ()

  Save the current WorkingCopy as a new revision.
- void DiscardEdit (SQImageEditLock editLock)

  Releases a pending edit lock.
- void RollbackSingleRevision ()

  Discards the latest saved revision.
- void RollbackToOriginalRevision ()

  Discards all but the primary revision.
- void Dispose ()

  Releases resources and deletes all temporary files associated with the SQImage.
- void Dispose (bool disposing)

  Releases resources and deletes all temporary files associated with the SQImage.

Static Public Member Functions

- static SQImage FromBitmap (Bitmap bitmap, bool disposeBitmap)

  Create an SQImage from a .Net bitmap.
- static SQImage FromBitmap (Bitmap bitmap)

  Create a new SQImage from a Bitmap
- static SQImage FromImage (Image image, bool disposeImage)

  Create an SQImage from a .Net Image.
- static SQImage FromImage (Image image)

  Create a new SQImage from a image
- static SQImage FromBytes (byte[ ] data)

  Create a new SQImage from a byte array.
- static SQImage FromHDib (IntPtr hDib, bool freeHDib)

  Create a new SQImage from a dib handle (hDib).
- static SQImage FromHDib (IntPtr hDib)

  Create a new SQImage from a dib handle (hDib).

Public Attributes

- string ImageFilesDirectory = SQImageRevision.ImageFilesDirectory

  Subdirectory to store image files in

Protected Member Functions

- void **NotifyPropertyChanged** (string propertyName)
- void **NotifyRevisionsChanged** ()
- void **OnRevisionsChanged** (object sender, EventArgs e)
- void EnsureEditing ()

  Raises an InvalidOperationException if WorkingCopy has not been loaded for editing.
- void EnsureNotEditing ()

  Throws an InvalidOperationException if the image is open for editing.
- void EnsureRevisionsNotEmpty ()

  Throws an InvalidOperationException if there are no revisions.
- SQImageEditLock BeginEdit (FreeImageBitmap image)

  Sets the provided image as the WorkingCopy and initializes an Edit operation.
- void _RollbackSingleRevision ()

  Discards the latest saved revision.

Protected Attributes

- Stack< SQImageRevision > Revisions = new Stack<SQImageRevision>()

  Stack containing all revisions of the image with the latest on top.
- HashSet< SQImageEditLock > EditLocks = new HashSet<SQImageEditLock>()

  Set of all open edit locks for this image.

Properties

- SQImageRevision LatestRevision `[get]`

  The latest SAVED revision.
- int **RevisionCount** `[get]`
- FreeImageBitmap WorkingCopy `[get, set]`

  Copy of the latest revision kept in memory for editing.
- bool Editing `[get]`

  True if WorkingCopy is loaded (available for editing), false if not.
- Bitmap Thumbnail `[get, protected set]`

  Thumbnail image of the latest revision.

Events

- PropertyChangedEventHandler **PropertyChanged**
- EventHandler RevisionsChanged

  Fired when a new revision is saved, or a revision is rolled back

5.83.1 Detailed Description

A single SQImage with revision control.

5.83.2 Constructor & Destructor Documentation

5.83.2.1 Scanquire.Public.SQImage.SQImage ( )

Base constructor, sets up internal event handlers.

5.83.2.2 Scanquire.Public.SQImage.SQImage (FreeImageBitmap *image*, out **SQImageEditLock** *editLock*)

Creates a new instance and leaves the image open for editing

Caller is responsible for calling DiscardEdit with the returned editLick

Creates a clone of image, so caller can modify/dispose original image.

5.83.2.3 Scanquire.Public.SQImage.SQImage (FreeImageBitmap *image*)

Creates a clone of image, so caller can modify/dispose original image.

5.83.3 Member Function Documentation

5.83.3.1 void Scanquire.Public.SQImage._RollbackSingleRevision ( ) `[protected]`

Discards the latest saved revision.

This only affects the latest SAVED revision, it has no impact on WorkingCopy For performance, does not raise the ImageChanged event. Can result in an empty revision stack. Should only be used if an immediate edit is anticipated.

5.83.3.2 SQImageEditLock Scanquire.Public.SQImage.BeginEdit (FreeImageBitmap *image*) [protected]

Sets the provided image as the WorkingCopy and initializes an Edit operation.

Returns

Edit lock, caller is reposible for disposing the edit lock.

Image is cloned, so caller is responsible for disposing passed in value.

5.83.3.3 SQImageEditLock Scanquire.Public.SQImage.BeginEdit ( )

Begins an edit operation. If WorkingCopy is not loaded, the latest revision will be loaded as the WorkingCopy.

Returns

Edit lock, caller is reposible for disposing the edit lock.

5.83.3.4 void Scanquire.Public.SQImage.DiscardEdit (SQImageEditLock *editLock*)

Releases a pending edit lock.

5.83.3.5 void Scanquire.Public.SQImage.Dispose ( )

Releases resources and deletes all temporary files associated with the SQImage.

5.83.3.6 void Scanquire.Public.SQImage.Dispose (bool *disposing*)

Releases resources and deletes all temporary files associated with the SQImage.

Parameters

disposing	

Once an SQImage has been disposed, recovery is no longer available.

5.83.3.7 void Scanquire.Public.SQImage.EnsureEditing ( ) [protected]

Raises an InvalidOperationException if WorkingCopy has not been loaded for editing.

5.83.3.8 void Scanquire.Public.SQImage.EnsureNotEditing ( ) [protected]

Throws an InvalidOperationException if the image is open for editing.

5.83.3.9 void Scanquire.Public.SQImage.EnsureRevisionsNotEmpty ( ) [protected]

Throws an InvalidOperationException if there are no revisions.

5.83.3.10 static SQImage Scanquire.Public.SQImage.FromBitmap (Bitmap *bitmap,* bool *disposeBitmap*) [static]

Create an SQImage from a .Net bitmap.

Parameters

bitmap	The bitmap to create the SQImage from.
disposeBitmap	If true, disposes the bitmap after creating the SQImage. If false, caller is responsible for disposing bitmap.

Returns

A new SQImage from the specified bitmap.

No reference to the provided bitmap is retained, it can be freely disposed.

5.83.3.11 static SQImage Scanquire.Public.SQImage.FromBitmap (Bitmap *bitmap*) `[static]`

Create a new SQImage from a Bitmap

Caller is responsible for disposing bitmap.

5.83.3.12 static SQImage Scanquire.Public.SQImage.FromBytes (byte[ ] *data*) `[static]`

Create a new SQImage from a byte array.

Parameters

data	

Returns

5.83.3.13 static SQImage Scanquire.Public.SQImage.FromHDib (IntPtr *hDib,* bool *freeHDib*) `[static]`

Create a new SQImage from a dib handle (hDib).

5.83.3.14 static SQImage Scanquire.Public.SQImage.FromHDib (IntPtr *hDib*) `[static]`

Create a new SQImage from a dib handle (hDib).

Caller is responsible for releasing hDib.

5.83.3.15 static SQImage Scanquire.Public.SQImage.FromImage (Image *image,* bool *disposeImage*) `[static]`

Create an SQImage from a .Net Image.

Parameters

bitmap	The Image to create the SQImage from.
disposeBitmap	If true, disposes the Image after creating the SQImage. If false, caller is responsible for disposing Image.

Returns

A new SQImage from the specified Image.

No reference to the provided Image is retained, it can be freely disposed.

5.83.3.16 static SQImage Scanquire.Public.SQImage.FromImage (Image *image*) [static]

Create a new SQImage from a image

Caller is responsible for disposing image.

5.83.3.17 void Scanquire.Public.SQImage.RollbackSingleRevision ( )

Discards the latest saved revision.

This only affects the latest SAVED revision, it has no impact on WorkingCopy Cannot be used to rollback the first revision.

5.83.3.18 void Scanquire.Public.SQImage.RollbackToOriginalRevision ( )

Discards all but the primary revision.

5.83.3.19 void Scanquire.Public.SQImage.Save (bool *overwritePreviousRevision*)

Save the current WorkingCopy as a revision.

Parameters

overwrite← Previous← Revision	True to delete the previous revision, false to retain it.

5.83.3.20 void Scanquire.Public.SQImage.Save ( )

Save the current WorkingCopy as a new revision.

5.83.4 Member Data Documentation

5.83.4.1 HashSet<SQImageEditLock> Scanquire.Public.SQImage.EditLocks = new HashSet<SQImageEditLock>() [protected]

Set of all open edit locks for this image.

5.83.4.2 string Scanquire.Public.SQImage.ImageFilesDirectory = SQImageRevision.ImageFilesDirectory

Subdirectory to store image files in

5.83.4.3 Stack<SQImageRevision> Scanquire.Public.SQImage.Revisions = new Stack<SQImageRevision>() [protected]

Stack containing all revisions of the image with the latest on top.

5.83.5 Property Documentation

5.83.5.1 bool Scanquire.Public.SQImage.Editing [get]

True if WorkingCopy is loaded (available for editing), false if not.

5.83.5.2 SQImageRevision Scanquire.Public.SQImage.LatestRevision [get]

The latest SAVED revision.

Does NOT return WorkingCopy.

5.83.5.3 Bitmap Scanquire.Public.SQImage.Thumbnail [get], [protected set]

Thumbnail image of the latest revision.

Value is cached. Automatically disposes any previous value.

5.83.5.4 FreeImageBitmap Scanquire.Public.SQImage.WorkingCopy [get], [set]

Copy of the latest revision kept in memory for editing.

Use BeginEdit to load and Save or DiscardEdit to release. To conserve memory, limit the number of images with working copies loaded and always Save or Discard and edits as soon as possible. Automatically disposes any previous value. If you directly assign a value to WorkingCopy, you are passing complete control of the value (including disposal) to the SQImage.

5.83.6 Event Documentation

5.83.6.1 EventHandler Scanquire.Public.SQImage.RevisionsChanged

Fired when a new revision is saved, or a revision is rolled back

Does not get fired on changes to WorkingCopy.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQImage/SQImage.cs

5.84 Scanquire.Public.SQImageEditLock Class Reference

Used to lock an SQImage to allow multiple edit operations.

Inheritance diagram for Scanquire.Public.SQImageEditLock:

Public Member Functions

- SQImageEditLock (SQImage owner)

  Create a new Lock agains the specified SQImage.
- void **Dispose** ()
- void Dispose (bool disposing)

  Discards any pending edits and unlocks the SQImage.

Properties

- Guid Id [get]

  Unique identifier for this lock instance.
- SQImage Owner [get]

  The SQImage that the lock applies to.

5.84.1 Detailed Description

Used to lock an SQImage to allow multiple edit operations.

5.84.2 Constructor & Destructor Documentation

5.84.2.1 Scanquire.Public.SQImageEditLock.SQImageEditLock (**SQImage** *owner*)

Create a new Lock agains the specified SQImage.

Parameters

owner	

5.84.3 Member Function Documentation

5.84.3.1 void Scanquire.Public.SQImageEditLock.Dispose (bool *disposing*)

Discards any pending edits and unlocks the SQImage.

Parameters

disposing	

5.84.4 Property Documentation

5.84.4.1 Guid Scanquire.Public.SQImageEditLock.Id `[get]`

Unique identifier for this lock instance.

5.84.4.2 **SQImage** Scanquire.Public.SQImageEditLock.Owner `[get]`

The SQImage that the lock applies to.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQImage/SQImageEditLock.cs

5.85 Scanquire.Public.UserControls.SQImageListViewer Class Reference

Custom control for navigating SQImages.

Inheritance diagram for Scanquire.Public.UserControls.SQImageListViewer:

Public Types

- enum ThumbnailSizeMode { **Small**, **Medium**, **Large** }

  Size of the visible thumbnails
- enum NavigationDirection { NavigationDirection.Up, NavigationDirection.Down, NavigationDirection.Left, NavigationDirection.Right, NavigationDirection.PageDown, NavigationDirection.PageUp, NavigationDirection.Home, NavigationDirection.End }
- enum **ImageThumbnailViewMode** { **Thumbnails**, **Image**, **ThumbnailsAndImage** }

Public Member Functions

- bool Contains (SQImageListViewerItem item)

  Returns

  True if the specified item exists in the current collection
- bool Contains (SQImage image)

  Returns

  True if the specified image exists in the current collection
- void EnsureContains (SQImageListViewerItem item)

  Throws an InvalidOperationException if the specified item does not exist in the current collection
- void EnsureDoesNotContain (SQImageListViewerItem item)

  Throws an InvalidOperationException if the specified item exists in the current collection
- void EnsureSelected (SQImageListViewerItem item)

  Throws an InvalidOperationException if the specified item is not selected
- void EnsureNotSelected (SQImageListViewerItem item)

  Throws an InvalidOperationException if the specified item is selected
- void EnsureChecked (SQImageListViewerItem item)

  Throws an InvalidOperationException if the specified item is not checked
- void EnsureNotChecked (SQImageListViewerItem item)

  Throws an InvalidOperationException if the specified item is checked
- bool IsActive (SQImageListViewerItem item)

  Returns

  True if the specified item currently holds focus.
- void EnsureActive (SQImageListViewerItem item)

  Throws an InvalidOperationException if the specified item does not hold focus
- void EnsureNotActive (SQImageListViewerItem item)

  Throws an InvalidOperationException if the specified item holds focus
- bool HasActiveItem ()

  Returns

  True if the viewer has an active item.
- void EnsureHasActiveItem ()

  Throws an InvalidOperationException if the collection does not contain an active item
- void EnsureDoesNotHaveActiveItem ()

  Throws an InvalidOperationException if the collection contains an active item
- bool IsActiveItemChecked ()

  Returns

  True if the currently focused item is checked.
- void EnsureActiveItemIsChecked ()

  Throws an InvalidOperationException if the focused item is not checked.
- void EnsureActiveItemIsNotChecked ()

  Throws an InvalidOperationException if the focused item is checked.
- bool IsActiveItemSelected ()

  Returns

  True if the currently focused item is selected.
- void EnsureActiveItemIsSelected ()

  Throws an InvalidOperationException if the focused item is not selected.
- void EnsureActiveItemIsNotSelected ()

  Throws an InvalidOperationException if the focused item is selected.
- int IndexOf (SQImageListViewerItem item)

Returns

The index of the specified item within the collection.

- void _Insert (int index, SQImage image)

  Insert a new image before the specified index
- void Insert (int index, SQImage image)

  Insert a new image before the specified index
- async Task Insert (int index, IEnumerable< SQImage > source)

  Asynchronously Insert multiple images before the specified index.
- async Task Insert (IEnumerable< SQImage > source)

  Asynchronously Insert multiple images before the focused item.
- async Task Add (IEnumerable< SQImage > source)

  Asynchronously add multiple images to the end of the collection.
- void Add (SQImage image)

  Add an image to the end of the collection.
- void Remove (SQImageListViewerItem item)

  Remove the specified item from the collection
- void Remove (int index)

  Remove the item located at the specified index from the collection
- void Remove (IEnumerable< SQImageListViewerItem > items)

  Remove the specified items from the collection
- void Check (SQImageListViewerItem item)

  Change the specified item's checked status to true.
- void Check (IEnumerable< SQImageListViewerItem > items)

  Change the specified items' checked status to true.
- void CheckAll ()

  Change all items' checked status to true.
- void CheckSelected ()

  Change all selected items' checked status to true.
- void UnCheck (SQImageListViewerItem item)

  Change the specified items' checked status to false.
- void UnCheck (IEnumerable< SQImageListViewerItem > items)

  Change the specified items' checked status to false.
- void ClearChecked ()

  Change all items' checked status to false.
- void UnCheckSelected ()

  Change all selected items' checked status to false.
- void InvertCheck (SQImageListViewerItem item)

  Toggle the status of the checked state of the specified item.
- void InvertCheck (IEnumerable< SQImageListViewerItem > items)

  Toggle the status of the checked state of the specified items.
- void InvertCheckSelected ()

  Toggle the status of the checked state of all selected items.
- void Select (SQImageListViewerItem item)

  Change the selection state of the specified item to true.
- void Select (IEnumerable< SQImageListViewerItem > items)

  Change the selection state of the specified items to true.
- void SelectAll ()

  Change the selection state of all items to true.
- void UnSelect (SQImageListViewerItem item)

  Change the selection state of the specified item to false.
- void UnSelect (IEnumerable< SQImageListViewerItem > items)

Change the selection state of the specified items to false.

- void ClearSelected ()

  Change the selection state of all items to false.
- void InvertSelect (SQImageListViewerItem item)

  Toggle the selection state of the specified item.
- void InvertSelect (IEnumerable< SQImageListViewerItem > items)

  Toggle the selection state of the specified items.
- void ActivateItem (SQImageListViewerItem item)

  Set focus on the specified item.
- void ClearActive ()

  Remove focus from the currently focused item (if any)
- IEnumerable< SQImageListViewerItem > GetRange (int start, int end)

  Returns

  The items in the collection from index start to index end
- IEnumerable< SQImageListViewerItem > GetRange (SQImageListViewerItem start, SQImageListViewerItem end)

  Returns

  The items in the collection from item start to item end.
- void ClearAll (bool confirmWithUser, bool disposeImages)

  Removes all items from the collection
- void NavigateTo (int index)

  Set focus to the item at the specified index and ensure that it is visible.
- void NavigateTo (SQImageListViewerItem item)

  Set focus to the item at the specified item and ensure that it is visible.
- void Navigate (NavigationDirection direction)

  Set focus to the item in the specified direction from the current active item..
- void CropActive (Rectangle?region=null)

  Crop the active image to the specified region.
- void FillActiveImage (Rectangle?region=null, Color?color=null)

  Fill the specified region of the active image with the specified color.
- void Deskew (SQImage image, float angle)

  Deskew the specified image by the specified angle.
- void UndoLatestRevision (SQImage image)

  Undo a single revision of the specified image.
- void UndoActiveItemLatestRevision ()

  Undo a single revision of the active image.

Public Attributes

- SQSplitContainer **SplitContainer**
- EdocsUSA.Controls.ImageViewer **ActiveImageViewer**
- SQToolStrip **ThumbnailToolStrip**

Protected Member Functions

- virtual void **OnItemCountChanged** ()
- virtual void **OnSelectedItemCountChanged** ()
- virtual void **OnCheckedItemCountChanged** ()
- virtual void **OnSavedStatusChanged** ()
- virtual void **OnActiveItemChanged** (SQImageListViewerItem previousItem)
- virtual void **OnActiveItemCheckedChanged** ()

- virtual void **OnActiveItemSelectedChanged** ()
- virtual void **OnActiveItemValueUpdated** ()
- virtual void **OnViewModeChanged** (ImageThumbnailViewMode previousValue)
- virtual void **OnThumbnailSizeModeChanged** (ThumbnailSizeMode previousValue)
- virtual void **OnSmallThumbnailSizeChanged** (Size previousValue)
- virtual void **OnLargeThumbnailSizeChanged** (Size previousValue)
- virtual void **OnMediumThumbnailSizeChanged** (Size previousValue)
- Point GetThumbnailPosition (int index)
- Point GetThumbnailPosition (SQImageListViewerItem item)

  Returns

  The position of the top left corner of the thumbnail at the specified item.
- SQImageListViewerItem HitTest (Point p)

  Returns

  The item (if any) that is located at the specified point.
- void **UpdateThumbnailScrollBar** ()
- void **ScrollToThumbnailRow** (int row)
- bool IsThumbnailVisible (int index)

  Returns

  True if the item at the specified collection index is currently visible (on-screen)
- bool IsThumbnailVisible (SQImageListViewerItem item)

  Returns

  True if the item at the specified item is currently visible (on-screen)
- void EnsureThumbnailIsVisible (int index)

  Throws an InvalidOperationException if the item at the specified index collection is not visible (on-screen)
- void EnsureThumbnailIsVisible (SQImageListViewerItem item)

  Throws an InvalidOperationException if the item at the specified item is not visible (on-screen)
- void **InvalidateThumbnailPanel** ()
- override bool **ProcessCmdKey** (ref Message msg, Keys keyData)
- override void OnMouseWheel (MouseEventArgs e)

  Scroll the thumbnail panel when the mouse-wheel is moved.
- void _RotateImage (SQImage image, RotateFlipType rotateFlipType)

  Rotate the specified image by the specified rotation type.
- async Task RotateImage (SQImage image, RotateFlipType rotateFlipType)

  Rotate the specified image by the specified rotation type.
- async Task _Rotate (IEnumerable< SQImage > images, RotateFlipType rotateFlipType)

  Asynchronously rotate the specified images by the specified rotation type.
- async Task Rotate (IEnumerable< SQImage > images, RotateFlipType rotateFlipType)

  Asynchronously rotate the specified images by the specified rotation type.
- async Task RotateFlipSelected (RotateFlipType rotateFlipType)

  Asynchronously rotate all selected images by the specified rotation type.
- void _CropImage (SQImage image, Rectangle region)

  Crop the specified image to the specified region.
- void CropImage (SQImage image, Rectangle region)

  Crop the specified image to the specified region.
- void SelectFillColor ()

  Prompt the user to select the color to be used for future Fill operations.
- void _FillRectangle (SQImage image, Rectangle region, Color color)

  Fill the specified region of the specified image with the specified color.
- void FillRectangle (SQImage image, Rectangle region, Color color)

  Fill the specified region of the specified image with the specified color.
- void _Deskew (SQImage image, float angle)

Deskew the specified image by the specified angle.

- void DeskewRight (SQImage image)

  Deskew the specified image to the right by the default deskew angle.
- void DeskewActiveItemRight ()

  Deskew the active image to the right by the default deskew angle.
- void DeskewLeft (SQImage image)

  Deskew the specified image to the left by the default deskew angle.
- void DeskewActiveItemLeft ()

  Deskew the active image to the right by the default deskew angle.
- override void Dispose (bool disposing)

  Clean up any resources being used.

Protected Attributes

- readonly SynchronizedBindingList< SQImageListViewerItem > **_Items**
- SQImageListViewerItem **_ActiveItem** = null
- readonly Padding ThumbnailMargin = new Padding(DefaultSettings.ThumbnailMargin)

  Spacing between displayed thumbnails.
- readonly Padding ThumbnailBorder = new Padding(DefaultSettings.ThumbnailBorder)

  Size of the border around each thumbnail.
- readonly Color ThumbnailBorderColor = SystemColors.Control

  Color of a thumbnail's border when not checked or selected.
- readonly Color CheckedThumbnailBorderColor = SystemColors.ControlDark

  Color of an thumbnail's border when checked.
- readonly Color SelectedThumbnailBorderColor = SystemColors.Highlight

  Color of a thumbnail's border when selected.
- readonly Font SmallThumbnailCaptionFont = SystemFonts.SmallCaptionFont

  Font for thumbnail captions when ThumbnailSizeMode is small.
- readonly Font MediumThumbnailCaptionFont = SystemFonts.SmallCaptionFont

  Font for thumbnail captions when ThumbnailSizeMode is medium.
- readonly Font LargeThumbnailCaptionFont = SystemFonts.CaptionFont

  Font for thumbnail captions when ThumbnailSizeMode is large.
- readonly Brush SelectedCaptionBrush = new SolidBrush(SystemColors.HighlightText)

  Brush for painting captions for selected thumbnails.
- readonly Brush UnSelectedCaptionBrush = new SolidBrush(SystemColors.ControlText)

  Brush for painting captions for unselected thumbnails.
- readonly StringFormat CaptionStringFormat

  FormatString for painting thumbnail captions.
- int ThumbnailSelectionAnchor

  Index of the initial item selected for a multi-select operation.
- Brush CheckedThumbnailBorderBrush = new SolidBrush(SystemColors.Control)

  Brush to paint the border of items that are checked.
- Brush UnCheckedThumbnailBorderBrush = new SolidBrush(Color.DarkRed)

  Brush to paint the border of items that are not checked
- Brush SelectedThumbnailBrush = new SolidBrush(Color.FromArgb(125, SystemColors.Highlight))

  Brush to paint the thumbnails of items that are selected
- Brush DisabledThumbnailBrush = new SolidBrush(Color.FromArgb(128, SystemColors.Control))

  Brush to paint the thumbnails of items that are not selected
- Pen ActiveImageBorderPen

  Pen for painting the border of the focused item.

Properties

- BusyStatus **BusyStatus** `[get]`
- ImageThumbnailViewMode **ViewMode** `[get, set]`
- static int **DefaultSplitterDistance** `[get]`
- IEnumerable< SQImageListViewerItem > Items `[get]`

  Current contents of the viewer
- int ItemCount `[get]`

  Total number of items currently in the viewer.
- IEnumerable< SQImageListViewerItem > Selected `[get]`

  Enumeration of all selected items in the viewer.
- int SelectedItemCount `[get]`

  Number of selected items in the viewer.
- IEnumerable< SQImageListViewerItem > UnSelected `[get]`

  Enumeration of all un-selected items in the viewer.
- int UnSelectedItemCount `[get]`

  Total number of un-selected
- IEnumerable< SQImageListViewerItem > Checked `[get]`

  Enumerable of all checked items
- int CheckedItemCount `[get]`

  Total number of checked items
- IEnumerable< SQImageListViewerItem > UnChecked `[get]`

  Enumerable of all un-checked items
- int UnCheckedItemCount `[get]`

  Total number of un-checked items
- IEnumerable< SQImage > Images `[get]`

  Enumerable of all images in the viewer
- IEnumerable< SQImage > SelectedImages `[get]`

  Enumerable of all selected images in the viewer
- IEnumerable< SQImage > UnSelectedImages `[get]`

  Enumerable of all un-selected images in the viewer
- IEnumerable< SQImage > CheckedImages `[get]`

  Enumerable of all checked images in the viewer
- IEnumerable< SQImage > UnCheckedImages `[get]`

  Enumerable of all un-checked images in the viewer
- SQImageListViewerItem ActiveItem `[get, set]`

  The item that currently has focus
- int ActiveItemIndex `[get]`

  The index of the item that currently has focus
- SQImage ActiveImage `[get]`

  The image associated with the currently focused item.
- ThumbnailSizeMode **CurrentThumbnailSizeMode** `[get, set]`
- Size ThumbnailImageSize `[get]`

  Pixel dimensions of thumbnail images defined by ThumbnailSizeMode
- Size ThumbnailControlSize `[get]`

  Total size of displayed a object (including the image, caption, border and margin).
- Size LargeThumbnailSize `[get, set]`

  Size of thumbnail images to display when ThumbnailSize = Large
- Size SmallThumbnailSize `[get, set]`

  Size of thumbnail images to display when ThumbnailSize = Small
- Size MediumThumbnailSize `[get, set]`

Size of thumbnail images to display when ThumbnailSize = Medium

- int MaxVisibleThumbnailColumns `[get]`

  Maximum number of thumbnail columns that will fit in ThumbnailPanel.
- int MaxVisibleThumbnailRows `[get]`

  Maximum number of thumbnail rows that will fit in ThumbnailPanel.
- int MaxVisibleThumbnails `[get]`

  Maximum number of thumbnails that will fir in ThumbnailPanel
- int TotalThumbnailRows `[get]`

  Total number of rows for all thumbnails
- int FirstVisibleThumbnailIndex `[get]`

  Index of the first visible thumbnail (based on the current scroll position)
- SQImageListViewerItem FirstVisibleItem `[get]`

  Index of the first visible (on-screen) item
- int LastVisibleThumbnailIndex `[get]`

  Index of the last visible (on-screen) item
- SQImageListViewerItem LastVisibleItem `[get]`

  The last item visible (on-screen)
- int MaxThumbnailScrollBarValue `[get]`

  Maximum value for the thumbnail scrollbar (based on number of thumbnails)
- int ThumbnailScrollBarValue `[get, set]`

  Current value of the thumbnail scrollbar
- Font ThumbnailCaptionFont `[get]`

  Font for displaying the thumbnail caption.
- float **DeskewAngle** `[get, set]`
- Color FillColor `[get, set]`

  Color to fill in the selected region of the active item during a fill operation.
- bool Saved `[get, set]`

  True if the collection has not been modified since the last save operation.

Events

- EventHandler **ItemCountChanged**
- EventHandler **SelectedItemCountChanged**
- EventHandler **CheckedItemCountChanged**
- EventHandler **SavedStatusChanged**
- EventHandler **ActiveItemChanged**
- EventHandler **ActiveItemCheckedChanged**
- EventHandler **ActiveItemSelectedChanged**
- EventHandler **ActiveItemValueUpdated**
- EventHandler **ViewModeChanged**

5.85.1 Detailed Description

Custom control for navigating SQImages.

5.85.2 Member Enumeration Documentation

5.85.2.1 enum Scanquire.Public.UserControls.SQImageListViewer.NavigationDirection

Enumerator

Up Navigate to the item directly above the active item

Down Navigate to the item directly below the active item

Left Navigate to the previous item

Right Navigate to the next item

PageDown Navigate to the item one screen below the active item

PageUp Navigate to the item one screen above the active item

Home Navigate to the first item

End Navigate to the last item

5.85.2.2 enum Scanquire.Public.UserControls.SQImageListViewer.ThumbnailSizeMode

Size of the visible thumbnails

5.85.3 Member Function Documentation

5.85.3.1 void Scanquire.Public.UserControls.SQImageListViewer._CropImage (SQImage *image,* Rectangle *region*) [protected]

Crop the specified image to the specified region.

5.85.3.2 void Scanquire.Public.UserControls.SQImageListViewer._Deskew (SQImage *image,* float *angle*) [protected]

Deskew the specified image by the specified angle.

5.85.3.3 void Scanquire.Public.UserControls.SQImageListViewer._FillRectangle (SQImage *image,* Rectangle *region,* Color *color*) [protected]

Fill the specified region of the specified image with the specified color.

5.85.3.4 void Scanquire.Public.UserControls.SQImageListViewer._Insert (int *index,* SQImage *image*)

Insert a new image before the specified index

5.85.3.5 async Task Scanquire.Public.UserControls.SQImageListViewer._Rotate (IEnumerable< SQImage > *images,* RotateFlipType *rotateFlipType*) [protected]

Asynchronously rotate the specified images by the specified rotation type.

5.85.3.6 void Scanquire.Public.UserControls.SQImageListViewer._RotateImage (SQImage *image,* RotateFlipType *rotateFlipType*) [protected]

Rotate the specified image by the specified rotation type.

5.85.3.7 void Scanquire.Public.UserControls.SQImageListViewer.ActivateItem (**SQImageListViewerItem** *item*)

Set focus on the specified item.

5.85.3.8 async Task Scanquire.Public.UserControls.SQImageListViewer.Add (IEnumerable< **SQImage** > *source*)

Asynchronously add multiple images to the end of the collection.

5.85.3.9 void Scanquire.Public.UserControls.SQImageListViewer.Add (**SQImage** *image*)

Add an image to the end of the collection.

5.85.3.10 void Scanquire.Public.UserControls.SQImageListViewer.Check (**SQImageListViewerItem** *item*)

Change the specified item's checked status to true.

5.85.3.11 void Scanquire.Public.UserControls.SQImageListViewer.Check (IEnumerable< **SQImageListViewerItem** > *items*)

Change the specified items' checked status to true.

5.85.3.12 void Scanquire.Public.UserControls.SQImageListViewer.CheckAll ( )

Change all items' checked status to true.

5.85.3.13 void Scanquire.Public.UserControls.SQImageListViewer.CheckSelected ( )

Change all selected items' checked status to true.

5.85.3.14 void Scanquire.Public.UserControls.SQImageListViewer.ClearActive ( )

Remove focus from the currently focused item (if any)

5.85.3.15 void Scanquire.Public.UserControls.SQImageListViewer.ClearAll (bool *confirmWithUser*, bool *disposeImages*)

Removes all items from the collection

Parameters

confirmWithUser	True to confirm with the user if the collection contains images and the saved flag is not set.
disposeImages	True if the image objects are no longer needed and can be disposed.

5.85.3.16 void Scanquire.Public.UserControls.SQImageListViewer.ClearChecked ( )

Change all items' checked status to false.

5.85.3.17 void Scanquire.Public.UserControls.SQImageListViewer.ClearSelected ( )

Change the selection state of all items to false.

5.85.3.18 void Scanquire.Public.UserControls.SQImageListViewer.CropActive (Rectangle? *region* = null)

Crop the active image to the specified region.

5.85.3.19 void Scanquire.Public.UserControls.SQImageListViewer.CropImage (SQImage *image*, Rectangle *region*) [protected]

Crop the specified image to the specified region.

5.85.3.20 void Scanquire.Public.UserControls.SQImageListViewer.Deskew (SQImage *image*, float *angle*)

Deskew the specified image by the specified angle.

5.85.3.21 void Scanquire.Public.UserControls.SQImageListViewer.DeskewActiveItemLeft ( ) [protected]

Deskew the active image to the right by the default deskew angle.

5.85.3.22 void Scanquire.Public.UserControls.SQImageListViewer.DeskewActiveItemRight ( ) [protected]

Deskew the active image to the right by the default deskew angle.

5.85.3.23 void Scanquire.Public.UserControls.SQImageListViewer.DeskewLeft (SQImage *image*) [protected]

Deskew the specified image to the left by the default deskew angle.

5.85.3.24 void Scanquire.Public.UserControls.SQImageListViewer.DeskewRight (SQImage *image*) [protected]

Deskew the specified image to the right by the default deskew angle.

5.85.3.25 override void Scanquire.Public.UserControls.SQImageListViewer.Dispose (bool *disposing*) [protected]

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

5.85.3.26 void Scanquire.Public.UserControls.SQImageListViewer.EnsureActive (SQImageListViewerItem *item*)

Throws an InvalidOperationException if the specified item does not hold focus

5.85.3.27 void Scanquire.Public.UserControls.SQImageListViewer.EnsureActiveItemIsChecked ( )

Throws an InvalidOperationException if the focused item is not checked.

5.85.3.28 void Scanquire.Public.UserControls.SQImageListViewer.EnsureActiveItemIsNotChecked ( )

Throws an InvalidOperationException if the focused item is checked.

5.85.3.29 void Scanquire.Public.UserControls.SQImageListViewer.EnsureActiveItemIsNotSelected ( )

Throws an InvalidOperationException if the focused item is selected.

5.85.3.30 void Scanquire.Public.UserControls.SQImageListViewer.EnsureActiveItemIsSelected ( )

Throws an InvalidOperationException if the focused item is not selected.

5.85.3.31 void Scanquire.Public.UserControls.SQImageListViewer.EnsureChecked (SQImageListViewerItem *item*)

Throws an InvalidOperationException if the specified item is not checked

5.85.3.32 void Scanquire.Public.UserControls.SQImageListViewer.EnsureContains (SQImageListViewerItem *item*)

Throws an InvalidOperationException if the specified item does not exist in the current collection

5.85.3.33 void Scanquire.Public.UserControls.SQImageListViewer.EnsureDoesNotContain (SQImageListViewerItem *item*)

Throws an InvalidOperationException if the specified item exists in the current collection

5.85.3.34 void Scanquire.Public.UserControls.SQImageListViewer.EnsureDoesNotHaveActiveItem ( )

Throws an InvalidOperationException if the collection contains an active item

5.85.3.35 void Scanquire.Public.UserControls.SQImageListViewer.EnsureHasActiveItem ( )

Throws an InvalidOperationException if the collection does not contain an active item

5.85.3.36 void Scanquire.Public.UserControls.SQImageListViewer.EnsureNotActive (SQImageListViewerItem *item*)

Throws an InvalidOperationException if the specified item holds focus

5.85.3.37 void Scanquire.Public.UserControls.SQImageListViewer.EnsureNotChecked (SQImageListViewerItem *item*)

Throws an InvalidOperationException if the specified item is checked

5.85.3.38 void Scanquire.Public.UserControls.SQImageListViewer.EnsureNotSelected (SQImageListViewerItem *item*)

Throws an InvalidOperationException if the specified item is selected

5.85.3.39 void Scanquire.Public.UserControls.SQImageListViewer.EnsureSelected (SQImageListViewerItem *item*)

Throws an InvalidOperationException if the specified item is not selected

5.85.3.40 void Scanquire.Public.UserControls.SQImageListViewer.EnsureThumbnailIsVisible (int *index*) [protected]

Throws an InvalidOperationException if the item at the specified index collection is not visible (on-screen)

5.85.3.41 void Scanquire.Public.UserControls.SQImageListViewer.EnsureThumbnailIsVisible (SQImageListViewerItem *item*) [protected]

Throws an InvalidOperationException if the item at the specified item is not visible (on-screen)

5.85.3.42 void Scanquire.Public.UserControls.SQImageListViewer.FillActiveImage (Rectangle? *region* = `null`, Color? *color* = `null`)

Fill the specified region of the active image with the specified color.

5.85.3.43 void Scanquire.Public.UserControls.SQImageListViewer.FillRectangle (SQImage *image*, Rectangle *region*, Color *color*) [protected]

Fill the specified region of the specified image with the specified color.

5.85.3.44 Point Scanquire.Public.UserControls.SQImageListViewer.GetThumbnailPosition (int *index*) [protected]

Parameters

index	The index of the item relative to all visible items.

Returns

The position of the top left corner of the thumbnail at the specified index.

5.85.3.45 void Scanquire.Public.UserControls.SQImageListViewer.Insert (int *index*, SQImage *image*)

Insert a new image before the specified index

5.85.3.46 async Task Scanquire.Public.UserControls.SQImageListViewer.Insert (int *index*, IEnumerable< SQImage > *source*)

Asynchronously Insert multiple images before the specified index.

5.85.3.47 async Task Scanquire.Public.UserControls.SQImageListViewer.Insert (IEnumerable< SQImage > *source*)

Asynchronously Insert multiple images before the focused item.

5.85.3.48 void Scanquire.Public.UserControls.SQImageListViewer.InvertCheck (SQImageListViewerItem *item*)

Toggle the status of the checked state of the specified item.

5.85.3.49 void Scanquire.Public.UserControls.SQImageListViewer.InvertCheck (IEnumerable< SQImageListViewerItem > *items*)

Toggle the status of the checked state of the specified items.

5.85.3.50 void Scanquire.Public.UserControls.SQImageListViewer.InvertCheckSelected ( )

Toggle the status of the checked state of all selected items.

5.85.3.51 void Scanquire.Public.UserControls.SQImageListViewer.InvertSelect (SQImageListViewerItem *item*)

Toggle the selection state of the specified item.

5.85.3.52 void Scanquire.Public.UserControls.SQImageListViewer.InvertSelect (IEnumerable< SQImageListViewerItem > *items*)

Toggle the selection state of the specified items.

5.85.3.53 void Scanquire.Public.UserControls.SQImageListViewer.Navigate (NavigationDirection *direction*)

Set focus to the item in the specified direction from the current active item..

5.85.3.54 void Scanquire.Public.UserControls.SQImageListViewer.NavigateTo (int *index*)

Set focus to the item at the specified index and ensure that it is visible.

5.85.3.55 void Scanquire.Public.UserControls.SQImageListViewer.NavigateTo (SQImageListViewerItem *item*)

Set focus to the item at the specified item and ensure that it is visible.

5.85.3.56 override void Scanquire.Public.UserControls.SQImageListViewer.OnMouseWheel (MouseEventArgs *e*) `[protected]`

Scroll the thumbnail panel when the mouse-wheel is moved.

5.85.3.57 void Scanquire.Public.UserControls.SQImageListViewer.Remove (SQImageListViewerItem *item*)

Remove the specified item from the collection

5.85.3.58 void Scanquire.Public.UserControls.SQImageListViewer.Remove (int *index*)

Remove the item located at the specified index from the collection

5.85.3.59 void Scanquire.Public.UserControls.SQImageListViewer.Remove (IEnumerable< SQImageListViewerItem > *items*)

Remove the specified items from the collection

5.85.3.60 async Task Scanquire.Public.UserControls.SQImageListViewer.Rotate (IEnumerable< SQImage > *images,* RotateFlipType *rotateFlipType*) `[protected]`

Asynchronously rotate the specified images by the specified rotation type.

5.85.3.61 async Task Scanquire.Public.UserControls.SQImageListViewer.RotateFlipSelected (RotateFlipType *rotateFlipType*) `[protected]`

Asynchronously rotate all selected images by the specified rotation type.

5.85.3.62 async Task Scanquire.Public.UserControls.SQImageListViewer.RotateImage (SQImage *image*, RotateFlipType *rotateFlipType*) [protected]

Rotate the specified image by the specified rotation type.

5.85.3.63 void Scanquire.Public.UserControls.SQImageListViewer.Select (SQImageListViewerItem *item*)

Change the selection state of the specified item to true.

5.85.3.64 void Scanquire.Public.UserControls.SQImageListViewer.Select (IEnumerable< SQImageListViewerItem > *items*)

Change the selection state of the specified items to true.

5.85.3.65 void Scanquire.Public.UserControls.SQImageListViewer.SelectAll ( )

Change the selection state of all items to true.

5.85.3.66 void Scanquire.Public.UserControls.SQImageListViewer.SelectFillColor ( ) [protected]

Prompt the user to select the color to be used for future Fill operations.

5.85.3.67 void Scanquire.Public.UserControls.SQImageListViewer.UnCheck (SQImageListViewerItem *item*)

Change the specified items' checked status to false.

5.85.3.68 void Scanquire.Public.UserControls.SQImageListViewer.UnCheck (IEnumerable< SQImageListViewerItem > *items*)

Change the specified items' checked status to false.

5.85.3.69 void Scanquire.Public.UserControls.SQImageListViewer.UnCheckSelected ( )

Change all selected items' checked status to false.

5.85.3.70 void Scanquire.Public.UserControls.SQImageListViewer.UndoActiveItemLatestRevision ( )

Undo a single revision of the active image.

5.85.3.71 void Scanquire.Public.UserControls.SQImageListViewer.UndoLatestRevision (SQImage *image*)

Undo a single revision of the specified image.

5.85.3.72 void Scanquire.Public.UserControls.SQImageListViewer.UnSelect (SQImageListViewerItem *item*)

Change the selection state of the specified item to false.

5.85.3.73 void Scanquire.Public.UserControls.SQImageListViewer.UnSelect (IEnumerable< SQImageListViewerItem > *items*)

Change the selection state of the specified items to false.

5.85.4 Member Data Documentation

5.85.4.1 Pen Scanquire.Public.UserControls.SQImageListViewer.ActiveImageBorderPen [protected]

Initial value:

```
= new Pen(SystemColors.ActiveBorder)
        {
            DashStyle = DashStyle.Dash,
            Width = 3
        }
```

Pen for painting the border of the focused item.

5.85.4.2 readonly StringFormat Scanquire.Public.UserControls.SQImageListViewer.CaptionStringFormat [protected]

Initial value:

```
= new StringFormat()
        {
            LineAlignment = StringAlignment.Center,
            Alignment = StringAlignment.Center
        }
```

FormatString for painting thumbnail captions.

5.85.4.3 Brush Scanquire.Public.UserControls.SQImageListViewer.CheckedThumbnailBorderBrush = new SolidBrush(SystemColors.Control) [protected]

Brush to paint the border of items that are checked.

5.85.4.4 readonly Color Scanquire.Public.UserControls.SQImageListViewer.CheckedThumbnailBorderColor = SystemColors.ControlDark [protected]

Color of an thumbnail's border when checked.

5.85.4.5 Brush Scanquire.Public.UserControls.SQImageListViewer.DisabledThumbnailBrush = new SolidBrush(Color.FromArgb(128, SystemColors.Control)) [protected]

Brush to paint the thumbnails of items that are not selected

5.85.4.6 readonly Font Scanquire.Public.UserControls.SQImageListViewer.LargeThumbnailCaptionFont = SystemFonts.CaptionFont [protected]

Font for thumbnail captions when ThumbnailSizeMode is large.

5.85.4.7 readonly Font Scanquire.Public.UserControls.SQImageListViewer.MediumThumbnailCaptionFont = SystemFonts.SmallCaptionFont [protected]

Font for thumbnail captions when ThumbnailSizeMode is medium.

5.85.4.8 readonly Brush Scanquire.Public.UserControls.SQImageListViewer.SelectedCaptionBrush = new SolidBrush(SystemColors.HighlightText) [protected]

Brush for painting captions for selected thumbnails.

5.85.4.9 readonly Color Scanquire.Public.UserControls.SQImageListViewer.SelectedThumbnailBorderColor = SystemColors.Highlight [protected]

Color of a thumbnail's border when selected.

5.85.4.10 Brush Scanquire.Public.UserControls.SQImageListViewer.SelectedThumbnailBrush = new SolidBrush(Color.FromArgb(125, SystemColors.Highlight)) [protected]

Brush to paint the thumbnails of items that are selected

5.85.4.11 readonly Font Scanquire.Public.UserControls.SQImageListViewer.SmallThumbnailCaptionFont = SystemFonts.SmallCaptionFont [protected]

Font for thumbnail captions when ThumbnailSizeMode is small.

5.85.4.12 readonly Padding Scanquire.Public.UserControls.SQImageListViewer.ThumbnailBorder = new Padding(DefaultSettings.ThumbnailBorder) [protected]

Size of the border around each thumbnail.

5.85.4.13 readonly Color Scanquire.Public.UserControls.SQImageListViewer.ThumbnailBorderColor = SystemColors.Control [protected]

Color of a thumbnail's border when not checked or selected.

5.85.4.14 readonly Padding Scanquire.Public.UserControls.SQImageListViewer.ThumbnailMargin = new Padding(DefaultSettings.ThumbnailMargin) [protected]

Spacing between displayed thumbnails.

5.85.4.15 int Scanquire.Public.UserControls.SQImageListViewer.ThumbnailSelectionAnchor [protected]

Index of the initial item selected for a multi-select operation.

5.85.4.16 Brush Scanquire.Public.UserControls.SQImageListViewer.UnCheckedThumbnailBorderBrush = new SolidBrush(Color.DarkRed) [protected]

Brush to paint the border of items that are not checked

5.85.4.17 readonly Brush Scanquire.Public.UserControls.SQImageListViewer.UnSelectedCaptionBrush = new SolidBrush(SystemColors.ControlText) [protected]

Brush for painting captions for unselected thumbnails.

5.85.5 Property Documentation

5.85.5.1 SQImage Scanquire.Public.UserControls.SQImageListViewer.ActiveImage [get]

The image associated with the currently focused item.

5.85.5.2 SQImageListViewerItem Scanquire.Public.UserControls.SQImageListViewer.ActiveItem [get], [set]

The item that currently has focus

5.85.5.3 int Scanquire.Public.UserControls.SQImageListViewer.ActiveItemIndex [get]

The index of the item that currently has focus

5.85.5.4 IEnumerable<SQImageListViewerItem> Scanquire.Public.UserControls.SQImageListViewer.Checked [get]

Enumerable of all checked items

5.85.5.5 IEnumerable<SQImage> Scanquire.Public.UserControls.SQImageListViewer.CheckedImages [get]

Enumerable of all checked images in the viewer

5.85.5.6 int Scanquire.Public.UserControls.SQImageListViewer.CheckedItemCount [get]

Total number of checked items

5.85.5.7 Color Scanquire.Public.UserControls.SQImageListViewer.FillColor [get], [set]

Color to fill in the selected region of the active item during a fill operation.

5.85.5.8 SQImageListViewerItem Scanquire.Public.UserControls.SQImageListViewer.FirstVisibleItem [get], [protected]

Index of the first visible (on-screen) item

5.85.5.9 int Scanquire.Public.UserControls.SQImageListViewer.FirstVisibleThumbnailIndex [get], [protected]

Index of the first visible thumbnail (based on the current scroll position)

5.85.5.10 IEnumerable<SQImage> Scanquire.Public.UserControls.SQImageListViewer.Images [get]

Enumerable of all images in the viewer

5.85.5.11 int Scanquire.Public.UserControls.SQImageListViewer.ItemCount [get]

Total number of items currently in the viewer.

5.85.5.12 IEnumerable<SQImageListViewerItem> Scanquire.Public.UserControls.SQImageListViewer.Items [get]

Current contents of the viewer

5.85.5.13 Size Scanquire.Public.UserControls.SQImageListViewer.LargeThumbnailSize [get], [set]

Size of thumbnail images to display when ThumbnailSize = Large

5.85.5.14 SQImageListViewerItem Scanquire.Public.UserControls.SQImageListViewer.LastVisibleItem [get], [protected]

The last item visible (on-screen)

5.85.5.15 int Scanquire.Public.UserControls.SQImageListViewer.LastVisibleThumbnailIndex [get], [protected]

Index of the last visible (on-screen) item

5.85.5.16 int Scanquire.Public.UserControls.SQImageListViewer.MaxThumbnailScrollBarValue [get], [protected]

Maximum value for the thumbnail scrollbar (based on number of thumbnails)

5.85.5.17 int Scanquire.Public.UserControls.SQImageListViewer.MaxVisibleThumbnailColumns [get]

Maximum number of thumbnail columns that will fit in ThumbnailPanel.

5.85.5.18 int Scanquire.Public.UserControls.SQImageListViewer.MaxVisibleThumbnailRows [get]

Maximum number of thumbnail rows that will fit in ThumbnailPanel.

5.85.5.19 int Scanquire.Public.UserControls.SQImageListViewer.MaxVisibleThumbnails [get]

Maximum number of thumbnails that will fir in ThumbnailPanel

5.85.5.20 Size Scanquire.Public.UserControls.SQImageListViewer.MediumThumbnailSize [get], [set]

Size of thumbnail images to display when ThumbnailSize = Medium

5.85.5.21 bool Scanquire.Public.UserControls.SQImageListViewer.Saved [get], [set]

True if the collection has not been modified since the last save operation.

5.85.5.22 IEnumerable< SQImageListViewerItem > Scanquire.Public.UserControls.SQImageListViewer.Selected [get]

Enumeration of all selected items in the viewer.

5.85.5.23 IEnumerable< SQImage > Scanquire.Public.UserControls.SQImageListViewer.SelectedImages [get]

Enumerable of all selected images in the viewer

5.85.5.24 int Scanquire.Public.UserControls.SQImageListViewer.SelectedItemCount [get]

Number of selected items in the viewer.

5.85.5.25 Size Scanquire.Public.UserControls.SQImageListViewer.SmallThumbnailSize [get], [set]

Size of thumbnail images to display when ThumbnailSize = Small

5.85.5.26 Font Scanquire.Public.UserControls.SQImageListViewer.ThumbnailCaptionFont [get], [protected]

Font for displaying the thumbnail caption.

5.85.5.27 Size Scanquire.Public.UserControls.SQImageListViewer.ThumbnailControlSize [get]

Total size of displayed a object (including the image, caption, border and margin).

5.85.5.28 Size Scanquire.Public.UserControls.SQImageListViewer.ThumbnailImageSize [get]

Pixel dimensions of thumbnail images defined by ThumbnailSizeMode

5.85.5.29 int Scanquire.Public.UserControls.SQImageListViewer.ThumbnailScrollBarValue [get], [set], [protected]

Current value of the thumbnail scrollbar

Normalizes to >= 0

5.85.5.30 int Scanquire.Public.UserControls.SQImageListViewer.TotalThumbnailRows [get]

Total number of rows for all thumbnails

5.85.5.31 IEnumerable<SQImageListViewerItem> Scanquire.Public.UserControls.SQImageListViewer.UnChecked [get]

Enumerable of all un-checked items

5.85.5.32 IEnumerable<SQImage> Scanquire.Public.UserControls.SQImageListViewer.UnCheckedImages [get]

Enumerable of all un-checked images in the viewer

5.85.5.33 int Scanquire.Public.UserControls.SQImageListViewer.UnCheckedItemCount [get]

Total number of un-checked items

5.85.5.34 IEnumerable<SQImageListViewerItem> Scanquire.Public.UserControls.SQImageListViewer.UnSelected [get]

Enumeration of all un-selected items in the viewer.

5.85.5.35 IEnumerable<SQImage> Scanquire.Public.UserControls.SQImageListViewer.UnSelectedImages [get]

Enumerable of all un-selected images in the viewer

5.85.5.36 int Scanquire.Public.UserControls.SQImageListViewer.UnSelectedItemCount [get]

Total number of un-selected

The documentation for this class was generated from the following files:

- C:/Code/Scanquire 4/Scanquire.Public/UserControls/SQImageListViewer.cs
- C:/Code/Scanquire 4/Scanquire.Public/UserControls/SQImageListViewer.Designer.cs

5.86 Testing.ImageListViewer.SQImageListViewer Class Reference

Inheritance diagram for Testing.ImageListViewer.SQImageListViewer:

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

5.86.1 Member Function Documentation

5.86.1.1 override void Testing.ImageListViewer.SQImageListViewer.Dispose (bool *disposing*) [protected]

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

The documentation for this class was generated from the following files:

- C:/Code/Scanquire 4/Testing/ImageListViewer/SQImageListViewer.cs
- C:/Code/Scanquire 4/Testing/ImageListViewer/SQImageListViewer.Designer.cs

5.87 Scanquire.Public.Properties.SQImageListViewer Class Reference

Inheritance diagram for Scanquire.Public.Properties.SQImageListViewer:

Properties

- static SQImageListViewer **Default** [get]
- global::System.Drawing.Color **DefaultCheckedItemBorderColor** [get]
- bool **DefaultImageCheckedState** [get]
- int **DefaultItemBorderWidth** [get]
- int **DefaultItemMarginWidth** [get]
- global::System.Drawing.Size **DefaultLargeThumbnailSize** [get]
- global::System.Drawing.Size **DefaultSmallThumbnailSize** [get]
- global::System.Drawing.Color **DefaultUnCheckedItemBorderColor** [get]
- global::System.Drawing.Size **DefaultMediumThumbnailSize** [get]
- global::System.Drawing.Color **DefaultSelectedThumbnailBorderColor** [get]

- int **ThumbnailMargin** [get]
- int **ThumbnailBorder** [get]
- float **DefaultDeskewAngle** [get]
- int **DefaultSplitterDistance** [get]

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Properties/SQImageListViewer.Designer.cs

5.88 Scanquire.Public.UserControls.SQImageListViewerItem Class Reference

A warpper for an SQImage to be added to an SQImageViewer

Inheritance diagram for Scanquire.Public.UserControls.SQImageListViewerItem:

Public Member Functions

- **SQImageListViewerItem** (SQImageListViewer owner, SQImage value)

Protected Member Functions

- void **OnPropertyChanged** (string propertyName)
- void **OnCheckedChanged** ()
- void **OnSelectedChanged** ()
- virtual void **Value_RevisionsChanged** (object sender, EventArgs e)

Properties

- bool **Checked** [get, set]
- bool **Selected** [get, set]
- SQImage **Value** [get]
- SQImageListViewer Owner [get]

  The SQImageListViewer containing this item.
- int Index [get]

  The collection index of this item within it's owning SQImageListViewer.

Events

- PropertyChangedEventHandler **PropertyChanged**
- EventHandler **CheckedChanged**
- EventHandler **SelectedChanged**
- EventHandler **ValueUpdated**

5.88.1 Detailed Description

A warpper for an SQImage to be added to an SQImageViewer

5.88.2 Property Documentation

5.88.2.1 int Scanquire.Public.UserControls.SQImageListViewerItem.Index [get]

The collection index of this item within it's owning SQImageListViewer.

5.88.2.2 SQImageListViewer Scanquire.Public.UserControls.SQImageListViewerItem.Owner [get]

The SQImageListViewer containing this item.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/UserControls/SQImageListViewer.cs

5.89 Scanquire.Public.SQImageProcessor_ConvertColorDepth Class Reference

Specialized ISQImageProcessor to convert the color depth of an image.

Inheritance diagram for Scanquire.Public.SQImageProcessor_ConvertColorDepth:

Public Member Functions

- **SQImageProcessor_ConvertColorDepth** (FREE_IMAGE_COLOR_DEPTH colorDepth)
- void ProcessImage (ref SQImage image)

  Apply the color depth manipulation to the specified image.

Properties

- FREE_IMAGE_COLOR_DEPTH ColorDepth [get, set]

  The color depth to apply to the image.

5.89.1 Detailed Description

Specialized ISQImageProcessor to convert the color depth of an image.

5.89.2 Member Function Documentation

5.89.2.1 void Scanquire.Public.SQImageProcessor_ConvertColorDepth.ProcessImage (ref SQImage *image*)

Apply the color depth manipulation to the specified image.

Parameters

image	The image to apply the color depth to.

Implements Scanquire.Public.ISQImageProcessor.

5.89.3 Property Documentation

5.89.3.1 FREE_IMAGE_COLOR_DEPTH Scanquire.Public.SQImageProcessor_ConvertColorDepth.ColorDepth `[get]`, `[set]`

The color depth to apply to the image.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQImageProcessor/SQImageProcessor_ConvertColorDepth.cs

5.90 Scanquire.Public.SQImageProcessor_RotateFlip Class Reference

Specialized ISQImageProcessor to Rotate & Flip an SQImage

Inheritance diagram for Scanquire.Public.SQImageProcessor_RotateFlip:

Public Member Functions

- **SQImageProcessor_RotateFlip** (RotateFlipType rotateFlipType)
- void ProcessImage (ref SQImage image)

  Perform the RotateFlip operation specified by RotateFlipType on the specified image.

Properties

- RotateFlipType RotateFlipType `[get, set]`

  RotateFlip operation to perform.

5.90.1 Detailed Description

Specialized ISQImageProcessor to Rotate & Flip an SQImage

5.90.2 Member Function Documentation

5.90.2.1 void Scanquire.Public.SQImageProcessor_RotateFlip.ProcessImage (ref **SQImage** *image*)

Perform the RotateFlip operation specified by RotateFlipType on the specified image.

Parameters

image	

Implements Scanquire.Public.ISQImageProcessor.

5.90.3 Property Documentation

5.90.3.1 RotateFlipType Scanquire.Public.SQImageProcessor_RotateFlip.RotateFlipType `[get]`, `[set]`

RotateFlip operation to perform.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQImageProcessor/SQImageProcessor_RotateFlip.cs

5.91 Scanquire.Public.SQImageProcessors Class Reference

Manages all configured ISQImageProcessors.

Inheritance diagram for Scanquire.Public.SQImageProcessors:

Properties

- override string **DirectoryPath** `[get]`
- static SQImageProcessors **Instance** `[get]`

Additional Inherited Members

5.91.1 Detailed Description

Manages all configured ISQImageProcessors.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQImageProcessor/SQImageProcessors.cs

5.92 Scanquire.Public.SQImageRevision Class Reference

Represents a single revision of an image.

Inheritance diagram for Scanquire.Public.SQImageRevision:

Public Member Functions

- SQImageRevision ()

  Serialization constructor. Not intended for normal use.
- SQImageRevision (FreeImageBitmap fib)
- FreeImageBitmap GetOriginalImage ()

  Get a copy of the original image.
- Bitmap GetOriginalImageBitmap ()

  Get a copy of the original image.
- FreeImageBitmap GetPreviewImage ()

  Get a copy of the preview image.
- Bitmap GetPreviewImageBitmap ()

  Get a copy of the preview image.
- FreeImageBitmap GetThumbnailImage ()

  Get a copy of the thumbnail image.
- Bitmap GetThumbnailBitmap ()

  Get a copy of the thumbnail image.
- void Dispose ()

  Delete the temporary files from the file system.
- void Dispose (bool disposing)

  Delete the temporary files from the file system.

Static Public Attributes

- static string ImageFilesDirectory = Path.Combine(SettingsManager.TempDirectoryPath, "Image Files")

  Directory to store image files.

Protected Member Functions

- Bitmap GetBitmap (string filePath)

  Read an image from the specified file path and return a Bitmap.
- FreeImageBitmap GetFreeImageBitmap (string filePath)

  Read an image from the specified file path and return a FreeImageBitmap.
- byte[ ] ReadFromTemporaryFile (string path)

  Read the binary contents of a temporary file.
- string WriteImageToTemporaryFile (FreeImageBitmap fib)

  Compresses and writes the image data to a random temporary file.
- string WriteScaledImageToTemporaryFile (FreeImageBitmap fib, Size maxSize, bool scaleResolution)

  Rescales the image , compresses it and writes it to a random temporary file.

Properties

- static bool GeneratePreviewImage `[get, set]`

  True to generate a reduced size preview image.
- string ThumbnailImageFilePath `[get, set]`

  Path to the thumbnail image file.
- string **PreviewImageFilePath** `[get, set]`
- string OriginalImageFilePath `[get, set]`

  Path to the original (full size) image file.

5.92.1 Detailed Description

Represents a single revision of an image.

Saves images to the filesystem. Caller must manually call Dispose to delete these image files. Dispose is NOT called by the object finalizer (by design to allow recovery on application crash).

5.92.2 Constructor & Destructor Documentation

5.92.2.1 Scanquire.Public.SQImageRevision.SQImageRevision ( )

Serialization constructor. Not intended for normal use.

Refer to the class documentation for warnings on temporary files and disposing. Caller MUST immediately populate Thumbnail/Preview/Original ImageFilePath properties.

5.92.2.2 Scanquire.Public.SQImageRevision.SQImageRevision (FreeImageBitmap *fib*)

Refer to the class documentation for warnings on temporary files and disposing SQImageRevision does not retain any reference or dependency to fib after creation. fib can be immediately disposed by caller after creation.

5.92.3 Member Function Documentation

5.92.3.1 void Scanquire.Public.SQImageRevision.Dispose ( )

Delete the temporary files from the file system.

Once disposed, recovery will no longer be available.

5.92.3.2 void Scanquire.Public.SQImageRevision.Dispose (bool *disposing*)

Delete the temporary files from the file system.

Once disposed, recovery will no longer be available.

5.92.3.3 Bitmap Scanquire.Public.SQImageRevision.GetBitmap (string *filePath*) [protected]

Read an image from the specified file path and return a Bitmap.

Does not generate a file lock on filePath.

5.92.3.4 FreeImageBitmap Scanquire.Public.SQImageRevision.GetFreeImageBitmap (string *filePath*) [protected]

Read an image from the specified file path and return a FreeImageBitmap.

Does not generate a file lock on filePath.

5.92.3.5 FreeImageBitmap Scanquire.Public.SQImageRevision.GetOriginalImage ( )

Get a copy of the original image.

Caller is responsible for disposing image. Does not create a lock on associated image file.

5.92.3.6 Bitmap Scanquire.Public.SQImageRevision.GetOriginalImageBitmap ( )

Get a copy of the original image.

Caller is responsible for disposing image. Does not create a lock on associated image file.

5.92.3.7 FreeImageBitmap Scanquire.Public.SQImageRevision.GetPreviewImage ( )

Get a copy of the preview image.

Caller is responsible for disposing image. Does not create a lock on associated image file. If GeneratePreviewImage is false, will return a copy of the original image.

5.92.3.8 Bitmap Scanquire.Public.SQImageRevision.GetPreviewImageBitmap ( )

Get a copy of the preview image.

Caller is responsible for disposing image. Does not create a lock on associated image file. If GeneratePreviewImage is false, will return a copy of the original image.

5.92.3.9 Bitmap Scanquire.Public.SQImageRevision.GetThumbnailBitmap ( )

Get a copy of the thumbnail image.

Caller is responsible for disposing image. Does not create a lock on associated image file.

5.92.3.10 FreeImageBitmap Scanquire.Public.SQImageRevision.GetThumbnailImage ( )

Get a copy of the thumbnail image.

Caller is responsible for disposing image. Does not create a lock on associated image file.

5.92.3.11 byte [ ] Scanquire.Public.SQImageRevision.ReadFromTemporaryFile (string *path*) [protected]

Read the binary contents of a temporary file.

5.92.3.12 string Scanquire.Public.SQImageRevision.WriteImageToTemporaryFile (FreeImageBitmap *fib*) [protected]

Compresses and writes the image data to a random temporary file.

Returns

Path to the temporary file.

5.92.3.13 string Scanquire.Public.SQImageRevision.WriteScaledImageToTemporaryFile (FreeImageBitmap *fib,* Size *maxSize,* bool *scaleResolution*) [protected]

Rescales the image , compresses it and writes it to a random temporary file.

Parameters

maxSize	Maximum size for the rescaled image. Image proportion will be maintained.
scaleResolution	True to scale the image's resolution to maintain the original image's physical size.

Returns

Path to the temporary file.

5.92.4 Member Data Documentation

5.92.4.1 string Scanquire.Public.SQImageRevision.ImageFilesDirectory = Path.Combine(SettingsManager.TempDirectoryPath, "Image Files") [static]

Directory to store image files.

If provided a relative path, it will be saved to the applications temporary directory.

5.92.5 Property Documentation

5.92.5.1 bool Scanquire.Public.SQImageRevision.GeneratePreviewImage [static], [get], [set]

True to generate a reduced size preview image.

If false, all requests for preview image will redirect to the original image. Value retrieved from/stores to UserSetting Settings.SQImageRevision.GeneratePreviewImage. Forces save to Settings.SQImageRevision.

5.92.5.2 string Scanquire.Public.SQImageRevision.OriginalImageFilePath [get], [set]

Path to the original (full size) image file.

5.92.5.3 string Scanquire.Public.SQImageRevision.ThumbnailImageFilePath [get], [set]

Path to the thumbnail image file.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQImage/SQImageRevision.cs

5.93 Scanquire.Public.Properties.SQImageRevision Class Reference

Inheritance diagram for Scanquire.Public.Properties.SQImageRevision:

Properties

- static SQImageRevision **Default** [get]
- bool **GeneratePreviewImage** [get, set]
- global::System.Drawing.Size **MaxPreviewImageSize** [get]
- global::System.Drawing.Size **MaxThumbnailImageSize** [get]

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Properties/SQImageRevision.Designer.cs

5.94 Scanquire.Public.Properties.SQImageViewer Class Reference

Inheritance diagram for Scanquire.Public.Properties.SQImageViewer:

Properties

- static SQImageViewer **Default** [get]
- float **DefaultDeskewAngle** [get]
- double **DefaultZoomMultiplier** [get]

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Properties/SQImageViewer.Designer.cs

5.95 Scanquire.Public.SQPage Class Reference

A single logical page containing a single image in a document.

Public Member Functions

- **SQPage** (SQImage image)
- **SQPage** (SQImage image, IEnumerable< ISQCommand_Page > commands)

Properties

- SQImage Image [get]

  The image to be drawn on this page
- List< ISQCommand_Page > Commands [get]

  Any page commands associated with the page.

5.95.1 Detailed Description

A single logical page containing a single image in a document.

5.95.2 Property Documentation

5.95.2.1 List<**ISQCommand_Page**> Scanquire.Public.SQPage.Commands [get]

Any page commands associated with the page.

5.95.2.2 **SQImage** Scanquire.Public.SQPage.Image [get]

The image to be drawn on this page

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQPage/SQPage.cs

5.96 Scanquire.Public.SQPhotoAlbumFilesystemArchiver Class Reference

Simple archiver for storing photographs to the local filesystem.

Inheritance diagram for Scanquire.Public.SQPhotoAlbumFilesystemArchiver:

Classes

- class PhotoAlbumInputDialog

Public Member Functions

- override void Send (SQFile file)

  Intended for use with single page image files only, so all processing will occur at the send images level.
- override void **Send** (IEnumerable< SQImage > images)

Protected Attributes

- PhotoAlbumInputDialog **InputDialog** = new PhotoAlbumInputDialog()

Properties

- string RootPath `[get, set]`

  Root path to store archived photos to.
- FREE_IMAGE_FORMAT HighQualityImageFormat `[get, set]`

  Image format for storing images when high quality copies are requested.
- FREE_IMAGE_SAVE_FLAGS HighQualityImageSaveFlags `[get, set]`

  Compression flags for storing images when high quality copies are requested.
- string HighQualityImageExtension `[get, set]`

  File extension to apply to images stored using the high quality settings
- FREE_IMAGE_FORMAT LowQualityImageFormat `[get, set]`

  Image format for storing images when low quality copies are requested.
- FREE_IMAGE_SAVE_FLAGS LowQualityImageSaveFlags `[get, set]`

  Compression flags for storing images when low quality copies are requested.
- string LowQualityImageExtension `[get, set]`

  File extension to apply to images stored using the low quality settings.

5.96.1 Detailed Description

Simple archiver for storing photographs to the local filesystem.

5.96.2 Member Function Documentation

5.96.2.1 override void Scanquire.Public.SQPhotoAlbumFilesystemArchiver.Send (SQFile *file*)

Intended for use with single page image files only, so all processing will occur at the send images level.

5.96.3 Property Documentation

5.96.3.1 string Scanquire.Public.SQPhotoAlbumFilesystemArchiver.HighQualityImageExtension `[get], [set]`

File extension to apply to images stored using the high quality settings

5.96.3.2 FREE_IMAGE_FORMAT Scanquire.Public.SQPhotoAlbumFilesystemArchiver.HighQualityImageFormat `[get], [set]`

Image format for storing images when high quality copies are requested.

5.96.3.3 FREE_IMAGE_SAVE_FLAGS Scanquire.Public.SQPhotoAlbumFilesystemArchiver.HighQualityImageSaveFlags `[get], [set]`

Compression flags for storing images when high quality copies are requested.

5.96.3.4 string Scanquire.Public.SQPhotoAlbumFilesystemArchiver.LowQualityImageExtension `[get], [set]`

File extension to apply to images stored using the low quality settings.

5.96.3.5 FREE_IMAGE_FORMAT Scanquire.Public.SQPhotoAlbumFilesystemArchiver.LowQualityImageFormat `[get], [set]`

Image format for storing images when low quality copies are requested.

5.96.3.6 FREE_IMAGE_SAVE_FLAGS Scanquire.Public.SQPhotoAlbumFilesystemArchiver.LowQualityImageSaveFlags [get], [set]

Compression flags for storing images when low quality copies are requested.

5.96.3.7 string Scanquire.Public.SQPhotoAlbumFilesystemArchiver.RootPath [get], [set]

Root path to store archived photos to.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQArchiver/SQPhotoAlbumFilesystemArchiver.cs

5.97 Scanquire.Public.UserControls.SQSplitContainer Class Reference

A standard SplitContainer with a rectangular grabber

Inheritance diagram for Scanquire.Public.UserControls.SQSplitContainer:

Protected Member Functions

- override void **OnPaint** (PaintEventArgs e)

5.97.1 Detailed Description

A standard SplitContainer with a rectangular grabber

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/UserControls/SQSplitContainer.cs

5.98 Scanquire.Public.UserControls.SQThumbnailPanel Class Reference

Panel for displaying thumnbail images.

Inheritance diagram for Scanquire.Public.UserControls.SQThumbnailPanel:

5.98.1 Detailed Description

Panel for displaying thumnbail images.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/UserControls/SQThumbnailPanel.cs

5.99 Scanquire.Public.UserControls.SQThumbnailScrollBar Class Reference

Unselectable scrollbar for use with the SQThumbnailPanel

Inheritance diagram for Scanquire.Public.UserControls.SQThumbnailScrollBar:

5.99.1 Detailed Description

Unselectable scrollbar for use with the SQThumbnailPanel

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/UserControls/SQThumbnailScrollBar.cs

5.100 Scanquire.Public.UserControls.SQToolStrip Class Reference

A non-selectable ToolStrip that defaults to a borderless Professional render mode.

Inheritance diagram for Scanquire.Public.UserControls.SQToolStrip:

5.100.1 Detailed Description

A non-selectable ToolStrip that defaults to a borderless Professional render mode.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/UserControls/SQToolStrip.cs

5.101 Scanquire.Public.Properties.SQTwain Class Reference

Inheritance diagram for Scanquire.Public.Properties.SQTwain:

Properties

- static SQTwain **Default** `[get]`
- string **PreviousDataSourceName** `[get, set]`

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Properties/SQTwain.Designer.cs

5.102 Scanquire.Public.SQTwain Class Reference

Public Member Functions

- IEnumerable< Task< SQImage > > Acquire (IProgress< ProgressEventArgs > progress, CancellationToken cToken)

  Retrieve a series of images from the Twain device.
- IEnumerable< Task< SQImage > > Acquire (string scannerName, string settingName, bool showUi, bool promptForContinue, IProgress< ProgressEventArgs > progress, CancellationToken cToken)

  Retrieve a series of images from the specified Twain device.

Protected Member Functions

- void ApplyImageFilters (SQImage image, IEnumerable< ISQImageProcessor > imageProcessors)

  Apply a series of imageProcessors to the specified image.

Protected Attributes

- SQTwainSettingsDialog **_SQTwainSettingsDialog** = null

Properties

- static IntPtr HostWindowHandle `[get]`

  The handle of the window making Twain requests.
- static IWin32Window HostWindow `[get]`

  The window making Twain requests.
- string PreviousDataSourceName `[get, set]`

  Name of the last used data source, to be used when no new data source is provided.
- Twain Twain `[get]`

  Twain interface.
- SQTwainSettingsDialog SQTwainSettingsDialog `[get]`

  Dialog to display and allow user selection of twain settings.
- static SQTwain **Instance** `[get]`

5.102.1 Member Function Documentation

5.102.1.1 IEnumerable<Task<SQImage> > Scanquire.Public.SQTwain.Acquire (IProgress< ProgressEventArgs > *progress,* CancellationToken *cToken*)

Retrieve a series of images from the Twain device.

Parameters

progress	Used to report progress back to the caller.
cToken	Cancelation token that can be set by the caller to stop acquiring.

Returns

An SQImage task for each image acquired from the Twain device.

5.102.1.2 IEnumerable<Task<SQImage> > Scanquire.Public.SQTwain.Acquire (string *scannerName,* string *settingName,* bool *showUi,* bool *promptForContinue,* IProgress< ProgressEventArgs > *progress,* CancellationToken *cToken*)

Retrieve a series of images from the specified Twain device.

Parameters

scannerName	The name of the twain device to aquire from. Name must be as it appears in the source list.
settingName	The name of the stored configuration to apply to the acquire job.
showUi	If true, display the device's native UI. If false, do not display the device's native UI.

promptFor↵ Continue	If true, ask the user to continue when the device finishes returning images. If false, stop when the device finishes returning images.
progress	Used to report progress back to the caller.
cToken	Cancelation token that can be set by the caller to stop acquiring.

Returns

An SQImage task for each image acquired from the Twain device.

5.102.1.3 void Scanquire.Public.SQTwain.ApplyImageFilters (**SQImage** *image,* IEnumerable< **ISQImageProcessor** > *imageProcessors*) [protected]

Apply a series of imageProcessors to the specified image.

Parameters

image	The image to process.
image↵ Processors	The imageProcessors to apply to the image.

5.102.2 Property Documentation

5.102.2.1 IWin32Window Scanquire.Public.SQTwain.HostWindow [static], [get]

The window making Twain requests.

5.102.2.2 IntPtr Scanquire.Public.SQTwain.HostWindowHandle [static], [get]

The handle of the window making Twain requests.

5.102.2.3 string Scanquire.Public.SQTwain.PreviousDataSourceName [get], [set], [protected]

Name of the last used data source, to be used when no new data source is provided.

5.102.2.4 SQTwainSettingsDialog Scanquire.Public.SQTwain.SQTwainSettingsDialog [get], [protected]

Dialog to display and allow user selection of twain settings.

5.102.2.5 Twain Scanquire.Public.SQTwain.Twain [get]

Twain interface.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQTwain/SQTwain.cs

5.103 Scanquire.Public.SQTwainSetting Class Reference

Public Member Functions

- **SQTwainSetting** (byte[ ] customDSData)
- **SQTwainSetting** (byte[ ] customDSData, IEnumerable< string > imageProcessorNames)
- **SQTwainSetting** (byte[ ] customDSData, params string[ ] imageProcessorNames)

Properties

- string ScannerName [get, set]

  Device name as it appears in the device list.
- bool ShowDeviceUI [get, set]

  If true, display the device's native UI when acquiring. If false, do not display the device's native UI when acquiring.
- byte[ ] CustomDSData [get, set]

  Configuration values for the device. The value is retrieved from the device.
- List< string > ImageProcessorNames [get, set]

  List of names of image processors to apply to all acquired images.
- IEnumerable< ISQImageProcessor > ImageProcessors [get]

  The image processors to apply to all acquired images.

5.103.1 Property Documentation

5.103.1.1 byte [ ] Scanquire.Public.SQTwainSetting.CustomDSData [get], [set]

Configuration values for the device. The value is retrieved from the device.

5.103.1.2 List<string> Scanquire.Public.SQTwainSetting.ImageProcessorNames [get], [set]

List of names of image processors to apply to all acquired images.

5.103.1.3 IEnumerable<ISQImageProcessor> Scanquire.Public.SQTwainSetting.ImageProcessors [get]

The image processors to apply to all acquired images.

5.103.1.4 string Scanquire.Public.SQTwainSetting.ScannerName [get], [set]

Device name as it appears in the device list.

5.103.1.5 bool Scanquire.Public.SQTwainSetting.ShowDeviceUI [get], [set]

If true, display the device's native UI when acquiring. If false, do not display the device's native UI when acquiring.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQTwain/SQTwainSetting.cs

5.104 Scanquire.Public.SQTwainSettings Class Reference

All available SQTwainSettings.

Inheritance diagram for Scanquire.Public.SQTwainSettings:

Properties

- override string **DirectoryPath** [get]
- static SQTwainSettings **Instance** [get]

Additional Inherited Members

5.104.1 Detailed Description

All available SQTwainSettings.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQTwain/SQTwainSettings.cs

5.105 Scanquire.Public.SQTwainSettingsDialog Class Reference

Dialog window to allow users to select and configure a twain device.

Inheritance diagram for Scanquire.Public.SQTwainSettingsDialog:

Classes

- class NewSettingDialog

  Dialog to create a new setting configuration for a device.

Public Member Functions

- **SQTwainSettingsDialog** (Twain twain)

Protected Member Functions

- void **SettingsListBox_SelectedIndexChanged** (object sender, EventArgs e)
- override void **OnShown** (EventArgs e)
- override void **OnClosed** (EventArgs e)
- void LoadSettings ()

  Loads all settings from SQTwainSettings into SettingsListBox.
- void **AddNewSetting** ()
- void RemoveSelectedSetting ()

  Delete the currently selected setting.

Protected Attributes

- ComboBox ScannersComboBox

  Listing of all available devices.
- ListBox SettingsListBox

  Listing of all available settings for the device selected in ScannersComboBox.
- Button AddSettingsButton

  Adds a new setting to the device selected in ScannersComboBox.
- Button RemoveSettingsButton

  Removes the setting selected in SettingsListBox from the device selected in ScannersComboBox.
- ListBox FiltersListBox

  List of available image processors.
- CheckBox ShowUICheckBox

If checked, device's native UI should be displayed on acquire. If not checked, device's native UI should not be displayed on acquire.

- CheckBox UsePreviousSettingsCheckBox

  If checked, should apply the last settings used.

Properties

- Twain **Twain** `[get, set]`
- string PreviousDataSourceName `[get, set]`

  Name of the last selected device name.
- string SelectedScanner `[get]`

  Name of the device selected.
- string SelectedSetting `[get]`

  Name of the setting configuration selected for the specified device.
- bool ShowUI `[get, set]`

  If true, device's native UI should be displayed on acquire. If false, device's native UI should not be displayed on acquire.
- bool UsePreviousSettings `[get, set]`

  If true, use the previous settings.

5.105.1 Detailed Description

Dialog window to allow users to select and configure a twain device.

5.105.2 Member Function Documentation

5.105.2.1 void Scanquire.Public.SQTwainSettingsDialog.LoadSettings ( ) `[protected]`

Loads all settings from SQTwainSettings into SettingsListBox.

5.105.2.2 void Scanquire.Public.SQTwainSettingsDialog.RemoveSelectedSetting ( ) `[protected]`

Delete the currently selected setting.

5.105.3 Member Data Documentation

5.105.3.1 Button Scanquire.Public.SQTwainSettingsDialog.AddSettingsButton `[protected]`

Adds a new setting to the device selected in ScannersComboBox.

5.105.3.2 ListBox Scanquire.Public.SQTwainSettingsDialog.FiltersListBox `[protected]`

List of available image processors.

5.105.3.3 Button Scanquire.Public.SQTwainSettingsDialog.RemoveSettingsButton `[protected]`

Removes the setting selected in SettingsListBox from the device selected in ScannersComboBox.

5.105.3.4 ComboBox Scanquire.Public.SQTwainSettingsDialog.ScannersComboBox [protected]

Listing of all available devices.

5.105.3.5 ListBox Scanquire.Public.SQTwainSettingsDialog.SettingsListBox [protected]

Listing of all available settings for the device selected in ScannersComboBox.

5.105.3.6 CheckBox Scanquire.Public.SQTwainSettingsDialog.ShowUICheckBox [protected]

If checked, device's native UI should be displayed on acquire. If not checked, device's native UI should not be displayed on acquire.

5.105.3.7 CheckBox Scanquire.Public.SQTwainSettingsDialog.UsePreviousSettingsCheckBox [protected]

If checked, should apply the last settings used.

5.105.4 Property Documentation

5.105.4.1 string Scanquire.Public.SQTwainSettingsDialog.PreviousDataSourceName [get], [set], [protected]

Name of the last selected device name.

5.105.4.2 string Scanquire.Public.SQTwainSettingsDialog.SelectedScanner [get]

Name of the device selected.

5.105.4.3 string Scanquire.Public.SQTwainSettingsDialog.SelectedSetting [get]

Name of the setting configuration selected for the specified device.

5.105.4.4 bool Scanquire.Public.SQTwainSettingsDialog.ShowUI [get], [set]

If true, device's native UI should be displayed on acquire. If false, device's native UI should not be displayed on acquire.

5.105.4.5 bool Scanquire.Public.SQTwainSettingsDialog.UsePreviousSettings [get], [set]

If true, use the previous settings.

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/SQTwain/SQTwainSettingsDialog.cs

5.106 Scanquire.Public.Properties.TemporaryFile Class Reference

Inheritance diagram for Scanquire.Public.Properties.TemporaryFile:

Properties

- static TemporaryFile **Default** `[get]`
- bool **Encrypt** `[get]`

The documentation for this class was generated from the following file:

- C:/Code/Scanquire 4/Scanquire.Public/Properties/TemporaryFile.Designer.cs