



PDF Converter Professional 5.0

Automation interface

Table of Contents

PDF Converter Professional 5.0 Automation interface.....	1
<i>Table of Contents</i>	2
<i>Interfaces – Overview</i>	5
<i>IApp interface</i>	6
Method Exit.....	6
Method CloseAllDocs	6
Method GetActiveDoc	6
Method GetActiveTool	7
Method GetDVDoc	7
Method GetLanguage	7
Method GetNumDVDocs.....	7
Method Hide.....	8
Method Show	8
Method Minimize	8
Method Maximize	8
Method Restore	8
Method SetActiveTool	9
Method MenuItemExecute	9
Method MenuItemIsEnabled.....	9
Method MenuItemIsMarked	9
Method MenuItemRemove	10
Method GetFrame	10
Method SetFrame	10
Method ToolButtonIsEnabled.....	10
Method ToolButtonRemove.....	11
<i>IDDDoc interface</i>	12
Method Create	12
Method Open.....	12
Method Save.....	13
Method Close	13
Method DeletePages.....	13
Method GetFileName	14
Method MovePage	14
Method ReplacePages	14
Method InsertPages	15
Method GetFlags	15
Method SetFlags.....	16
Method ClearFlags	16
Method GetPageMode.....	16
Method SetPageMode	17
Method OpenDVDoc	17
Method DeleteThumbs	17
Method CreateThumbs	17
Method GetInfo	18
Method SetInfo.....	18
Method GetPermanentID	18
Method GetInstanceID	18
Method CropPages	19

Method AcquirePage	19
Method CreateTextSelect	19
<i>IDVDoc interface</i>	20
Method GetDDDdoc	20
Method Open	20
Method Close	20
Method PrintPages	21
Method PrintPagesSilent	21
Method GetTitle	21
Method Maximize	22
Method GetFrame	22
Method IsValid	22
Method GetViewMode	22
Method SetViewMode	23
Method GetDVPageView	23
Method SetTextSelection	23
Method ShowTextSelect	23
Method ClearSelection	24
Method BringToFront	24
Method SetTitle	24
Method PrintPagesEx	25
Method PrintPagesSilentEx	26
<i>IDDPAGE Interface</i>	27
Method GetRotate	27
Method SetRotate	27
Method GetNumAnnots	27
Method GetNumber	28
Method CropPage	28
Method GetAnnot	28
Method AddAnnot	29
Method AddNewAnnot	29
Method GetDoc	29
Method GetAnnotIndex	30
Method GetSize	30
Method CreateWordHilite	30
Method CreatePageHilite	30
<i>IDDAnnot interface</i>	31
Method GetColor	31
Method SetColor	31
Method GetDate	31
Method SetDate	32
Method GetRect	32
Method SetRect	32
Method GetTitle	32
Method SetTitle	33
Method GetSubType	33
Method IsValid	33
Method IsOpen	33
Method SetOpen	34
Method GetContents	34

Method SetContents	34
Method IsEqual	34
Method Perform	35
<i>IDVPageView</i>	36
Method DoGoBack	36
Method DoGoForward	36
Method GetDVDoc	36
Method GetDoc	37
Method GetPage	37
Method GetPageNum	37
Method GetZoom	37
Method GetZoomType	38
Method Goto	38
Method ScrollTo	38
Method ZoomTo	38
Method DevicePointToPage	39
Method PointToDevice	39
Method GetAperture	39
<i>IHiliteList</i>	40
Method Add.....	40
<i>IDDTextSelect</i>	40
Method Destroy.....	40
Method GetBoundingRect.....	40
Method GetNumText	41
Method GetPage	41
Method GetText	41
<i>Helper interfaces</i>	42
IRect interface	42
ITime interface	42
IPoint interface	42
<i>Examples</i>	43
Example 1 – Inserting pages into a document.....	43
Example 2 – Using the MenuItemExecute method of the IApp interface	44
Example 3 – Highlighting words in a PDF document	44
Example 4 – Adding annotations	46
<i>Appendix</i>	47
The menu names.....	47

Interfaces – Overview

PDF Converter Professional provides the automation interfaces below. These interfaces provide a wide range of functionality to control the application itself, manage and manipulate PDF documents, their properties, pages and annotations.

IApp	Creatable	Controls the PDF Professional application
IDDDoc	Creatable	Controls the PDF documents
IDVDoc	Creatable	Controls the view of PDF documents
IDDPPage	Non-creatable	Accessible through IDDDoc interface. Controls the pages within the given PDF document
IDDAnnot	Non-creatable	Accessible through IDDPPage interface
IDVPageView	Non-creatable	Accessible through IDVDoc interface. Controls viewing PDF pages
IHiliteList	Creatable	This collection is designed to specify the location of highlighted contents within a PDF document.
IDDTextSelect	Creatable	This interface is designed to set and show the highlights within a PDF document.
IRect	Creatable	These interfaces are designed to set or retrieve special properties of different objects
ITime	Creatable	
IPoint	Creatable	

This documentation provides full details of these interfaces and also gives examples for their usage. It also provides the implementation details and relationships between the different interfaces.

Each section provides the definition of a particular method with its input and return parameters, a short description of its usage and points to the related topics.

The documentation also provides several examples. These samples are collected in a separate chapter at the end of the document. The sections of methods contain references to the samples where the particular methods are being used.

IApp interface

The IApp interface is designed to drive the PDF Converter Professional application. It provides a control over the application main window and its UI elements, the menus, toolbars and the tools. The opened documents are also accessible through the interface.

The application main window can be kept hidden if the automation client performs document level operations only.

Method Exit

Syntax: VARIANT_BOOL Exit();

Parameters: None

Return value: VARIANT_TRUE, if pdfplus.exe exited properly.

Remarks: Closes all opened documents and quits from the application.

Example: [Example 1](#); [Example 2](#);

Method CloseAllDocs

Syntax: VARIANT_BOOL CloseAllDocs();

Parameters: None

Return value: Closes all documents opened in the application.

Remarks: Closes all documents referred by the IDVDoc interface pointer. The DDDoc objects have to be closed by the Automation client application.

Example: [Example 2](#)

Method GetActiveDoc

Syntax: IDispatch* GetActiveDoc();

Parameters: None

Return value: Retrieves an interface to the active DVDoc object.

Remarks: Use it to retrieve the topmost document selected by the user.

Example:

Method GetActiveTool

Syntax: `BSTR GetActiveTool();`
Parameters: None
Return value: Returns with the name of the selected tool. The return value can be a NULL pointer.
Remarks: Use it to retrieve the name of the selected tool.
Example:

Method GetDVDoc

Syntax: `IDispatch* GetDVDoc(long nIndex);`
Parameters: `long nIndex`: The identifier for the DVDoc object.
Return value: Retrieves an interface to the DVDoc object identified by its index.
Remarks: The application object maintains a list of open PDF documents. You can get any of them by referring to their index.
Example:

Method GetLanguage

Syntax: `BSTR GetLanguage();`
Parameters: None
Return value: Returns the abbreviated string of the language. The available language codes are the following:
ENU - English
DEU - German
FRA - French
DUT - Dutch
ITA - Italian
ESP - Spanish
SVE - Swedish
DAN - Danish
Remarks: Use this method to identify the product installation language.
Example:

Method GetNumDVDocs

Syntax: `long GetNumDVDocs();`
Parameters: None
Return value: Returns the number of opened documents.
Remarks: You can use this method for enumeration of open documents.
Example:

Method Hide

Syntax: `VARIANT_BOOL Hide();`
Parameters: None
Return value: `VARIANT_TRUE` if the operation succeeded.
Remarks: Hides the application main window.
Example:

Method Show

Syntax: `VARIANT_BOOL Show();`
Parameters: None
Return value: `VARIANT_TRUE` if the operation succeeded.
Remarks: Makes the application main window visible.
Example:

Method Minimize

Syntax: `VARIANT_BOOL Minimize(long bMinimize);`
Parameters: None
Return value: `VARIANT_TRUE` if the operation succeeded.
Remarks: Shows the application main window minimized.
Example:

Method Maximize

Syntax: `VARIANT_BOOL Maximize(long bMaximize);`
Parameters: None
Return value: `VARIANT_TRUE` if the operation succeeded.
Remarks: Shows the application main window minimized.
Example:

Method Restore

Syntax: `VARIANT_BOOL Restore(long bRestore);`
Parameters: None
Return value: `VARIANT_TRUE` if the operation succeeded.
Remarks: Shows the application main window maximized.
Example:

Method SetActiveTool

Syntax: `VARIANT_BOOL SetActiveTool(BSTR szButtonName, long bPersistent);`

Parameters: `BSTR szButtonName`: The name of the tool.
`long bPersistent`: if a positive number, the tool will remain active after it has been used, 0 otherwise.

Return value: `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method to activate a particular tool.

Example:

Method MenuItemExecute

Syntax: `VARIANT_BOOL MenuItemExecute(BSTR szMenuItemName);`

Parameters: `BSTR szMenuItemName`: Specifies the action to be performed.

Return value: `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method to perform a specific action associated with the given menu item. The possible menu item names are listed in the Appendix.

Example: [Example 2](#)

Method MenuItemIsEnabled

Syntax: `VARIANT_BOOL MenuItemIsEnabled(BSTR szMenuItemName);`

Parameters: `BSTR szMenuItemName`: Specifies the menu item.

Return value: Returns `VARIANT_TRUE` if the menu item is enabled.

Remarks: Before performing the `MenuItemExecute` method, it might be necessary to check the state of the menu item.

Example: [Example 2](#)

Method MenuItemIsMarked

Syntax: `VARIANT_BOOL MenuItemIsMarked(BSTR szMenuItemName);`

Parameters: `BSTR szMenuItemName`: Specifies the menu item.

Return value: Returns `VARIANT_TRUE` if the menu item is checked.

Remarks: Certain menu items have a 'checked' state. You can retrieve this state of a menu item. Not applicable for all menu items.

Example:

Method MenuItemRemove

Syntax: `VARIANT_BOOL MenuItemRemove(BSTR szMenuItemName);`

Parameters: `BSTR szMenuItemName`: Specifies the menu item.

Return value: `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method if you don't want to let a user have access to certain menu items. These items will be removed from the menu structure of the application.

Example:

Method GetFrame

Syntax: `IDispatch* GetFrame();`

Parameters: None

Return value: Return a reference to a `Rect` object.

Remarks: Use this method to retrieve the position and the size of the main application window.

Example:

Method setFrame

Syntax: `VARIANT_BOOL setFrame(IDispatch* iRect);`

Parameters: `IDispatch* iRect`: Dispatch interface of a `Rect` object that specifies the coordinates of the main application window.

Return value: `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method if you want to specify the coordinates of the main application window.

Example:

Method ToolButtonsEnabled

Syntax: `VARIANT_BOOL ToolButtonsEnabled(BSTR szButtonName);`

Parameters: `BSTR szButtonName`: Specifies the toolbar button.

Return value: Returns `VARIANT_TRUE` if the toolbar button is enabled.

Remarks: Verifies whether the specified button is enabled or disabled.

Example:

Method ToolButtonRemove

Syntax:	<code>VARIANT_BOOL ToolButtonRemove(BSTR szButtonName);</code>
Parameters:	<code>BSTR szButtonName</code> : Specifies the toolbar button.
Return value:	<code>VARIANT_TRUE</code> if the operation succeeded.
Remarks:	You can hide the specified toolbar button while the application runs through automation.
Example:	

IDDDoc interface

This interface controls PDF documents at file-level without involving user-interface related operations. The files can be opened, modified, saved and closed. The interface provides a number of page operations within and between the documents.

Method Create

Syntax:	VARIANT_BOOL Create();
Parameters:	None
Return value:	VARIANT_TRUE if the operation succeeded.
Remarks:	Creates a new PDF document. The file does not contain any pages.
Example:	

Method Open

Syntax:	VARIANT_BOOL Open(BSTR szFullPath);
Parameters:	BSTR szFullPath: Specifies the PDF file to be opened.
Return value:	VARIANT_TRUE if the operation succeeded.
Remarks:	Use this method to open the specified PDF file. A document opened in this way will not be displayed in the application window. Note that the maximum number of open documents is 50.
Example:	Example 1

Method Save

Syntax:	<code>VARIANT_BOOL Save(short nType, BSTR szFullPath);</code>
Parameters:	<p><code>short nType</code>: Specifies the method for how the document should be saved. The parameter can be a combination (with logical OR) of the following values.</p> <p><code>DDSaveIncremental (0)</code> -- Write changes only, not the complete file. This will always result in a larger file, even if objects have been deleted.</p> <p><code>DDSaveFull (1)</code> -- Write the entire file to the filename specified by <code>szFullPath</code>.</p> <p><code>DDSaveCopy (2)</code> -- Write a copy of the file into the file specified by <code>szFullPath</code>, but keep using the old file. This flag can only be specified if <code>DDSaveFull</code> is also used.</p> <p><code>DDSaveCollectGarbage (32)</code> -- Remove unreferenced objects; this often reduces the file size, and its usage is encouraged. This flag can only be specified if <code>DDSaveFull</code> is also used.</p> <p><code>DDSaveLinearized (4)</code> -- Save the file optimized for the web, providing hint tables. This allows the PDF file to be byte-served. This flag can only be specified if <code>DDSaveFull</code> is also used.</p> <p><code>BSTR szFullPath</code>: Specifies the file name. You are allowed to use the original file name that was used for the method <code>Open</code>.</p>
Return value:	<code>VARIANT_TRUE</code> : if the saving operation succeeded.
Remarks:	Saves the content of the document to the specified file.
Example:	Example 1

Method Close

Syntax:	<code>VARIANT_BOOL Close();</code>
Parameters:	None
Return value:	<code>VARIANT_TRUE</code> , if the document was closed properly.
Remarks:	Closes the <code>DDDoc</code> object.
Example:	Example 1

Method DeletePages

Syntax:	<code>VARIANT_BOOL DeletePages(long nStartPage, long nEndPage);</code>
Parameters:	<p><code>long nStartPage</code>: Specifies the first page to be deleted.</p> <p><code>long nEndPage</code>: Specifies the last page to be deleted.</p>
Return value:	<code>VARIANT_TRUE</code> : if the operation succeeded.
Remarks:	Deletes the specified pages from the document.
Example:	

Method GetFileName

Syntax: `BSTR GetFileName();`

Parameters: None

Return value: Returns with the name of the file referenced by the DDDoc object.

Remarks: Use it to retrieve the name of the opened document.

Example:

Method MovePage

Syntax: `VARIANT_BOOL MovePage(long nMoveAfterThisPage, long nPageToMove);`

Parameters: `long nMoveAfterThisPage:` Specifies the page number after which the moved page will be placed.

`long nPageToMove:` Specifies the page to be moved.

Return value: `VARIANT_TRUE` if the operation succeeded.

Remarks: Moves a page to a new position within the document.

Example:

Method ReplacePages

Syntax: `VARIANT_BOOL ReplacePages(long nStartPage, IDispatch* iDDDocSource, long nStartSourcePage, long nNumPages, long bMergeTextAnnotations);`

Parameters: `long nStartPage:` Specifies the first page in the target document to be replaced. The next `nNumPages` pages will be replaced.

`IDispatch* iDDDocSource:` Specifies the source document from where the document pages are copied.

`long nStartSourcePage:` Specifies the first page of the source document to be moved.

`long nNumPages:` Specifies the number of pages to be moved to the target document.

`long bMergeTextAnnotations:` Set it to value 1, if you want to merge the annotations of the original and the new pages. In case of value 0 the text annotations will not be merged.

Return value: `VARIANT_TRUE` if the operation succeeded.

Remarks: Replaces a given number of pages of the document with pages of another document. The annotations of the affected pages are optionally merged.

Example: [Example 1](#) is applicable for this method as well.

Method InsertPages

Syntax:	<code>VARIANT_BOOL InsertPages(long nInsertPageAfter, IDispatch* iDDDocSource, long nStartPage, long nNumPages, long bBookmarks);</code>
Parameters:	<p><code>long nInsertPagesAfter</code>: Specifies a page in the target document after which the new pages will be inserted. Use a value of -1 to insert pages to the beginning of the document.</p> <p><code>IDispatch* iDDDocSource</code>: Specifies the source document where the new pages are being inserted from.</p> <p><code>long nStartPage</code>: Specifies the very first page of the source document that has to be inserted into the target document.</p> <p><code>long nNumPages</code>: Specifies the number of pages following the start page to be inserted.</p> <p><code>long bBookmarks</code>: Set this parameter to 1 to copy the bookmarks along with the pages. Note that all bookmarks will be copied even if they do not point to existing pages in the new document!</p>
Return value:	VARIANT_TRUE if the operation succeeded.
Remarks:	Copies one or more pages from a PDF document and places them after the specified page in the current PDF file.
Example:	Example 1

Method GetFlags

Syntax:	<code>long GetFlags();</code>	
Parameters:	None	
Return value:	Returns the document flags. The return value is made by a logical OR operation of the following values.	
<code>DDDocNeedsSave</code>	= 1,	The document has been modified.
<code>DDDocRequiresFullSave</code>	= 2,	Incremental saving would not be enough. Use the <code>DDSaveFull</code> flag when document is saved.
<code>DDDocIsModified</code>	= 4,	The document is modified (e.g. bookmarks or annotations have been added).
<code>DDDocDeleteOnClose</code>	= 8,	The document is a temporary one. It has to be deleted when the file is closed.
<code>DDDocWasRepaired</code>	= 16,	The original document was repaired during the method <code>DDDoc.Open</code> . The reason for the repair might be a damaged or incorrectly saved PDF file.
<code>DDDocIsLinearized</code>	= 1024,	The document is linearized.
<code>DDDocIsOptimized</code>	= 2048	The document is optimized.
Remarks:	Use this method to retrieve the current state and other parameters of the document.	
Example:		

Method SetFlags

Syntax:	<code>VARIANT_BOOL SetFlags(long nFlags);</code>
Parameters:	<code>long nFlags</code> : The required value of the document flags.
Return value:	<code>VARIANT_TRUE</code> if the operation succeeded.
Remarks:	Sets the document flags. The possible values are described under the method <code>GetFlags</code> . Most of the flags are read-only and cannot be set or cleared. <code>DDDocNeedsSave</code> and <code>DDDocDeleteOnClose</code> flags can be set and cleared.
Example:	

Method ClearFlags

Syntax:	<code>VARIANT_BOOL ClearFlags(long nFlags);</code>
Parameters:	<code>long nFlags</code> : Specify the document flags to be cleared.
Return value:	<code>VARIANT_TRUE</code> if the operation succeeded.
Remarks:	Use this flag to clear document flags. Most of the flags are read-only and cannot be set or cleared. <code>DDDocNeedsSave</code> and <code>DDDocDeleteOnClose</code> flags can be set and cleared.
Example:	

Method GetPageMode

Syntax:	<code>long GetPageMode();</code>
Parameters:	None
Return value:	The return value can be one the following: <code>DDDontCare = 0,</code> <code>DDUseNone = 1,</code> <code>DDUseThumbs = 2,</code> <code>DDUseBookmarks = 3,</code> <code>DDFullScreen = 4</code>
Remarks:	Use this method to retrieve the current page mode of the document. This property controls how the document is displayed initially: displays only the document pages, the pages and the thumbnails or the pages and the bookmarks. The document can also be displayed in full screen mode.
Example:	

Method SetPageMode

Syntax:	<code>VARIANT_BOOL SetPageMode(long nPageMode);</code>
Parameters:	<code>long nPageMode</code> : Specifies the required Page Mode.
Return value:	<code>VARIANT_TRUE</code> if the Page Mode was set properly.
Remarks:	Sets the Page mode of the document. The possible values of the Page Mode property are described under the method <code>GetPageMode</code> .
Example:	

Method OpenDVDoc

Syntax:	<code>IDispatch* OpenDVDoc(BSTR szTitle);</code>
Parameters:	<code>BSTR szTitle</code> : The title of the document view.
Return value:	Returns a DVDoc object referring to the DDDoc object.
Remarks:	Creates a DVDoc object associated to the DDDoc object. A new document view will be displayed that shows the document referred by the DDDoc object. This call makes the application visible if it was hidden originally.
Example:	

Method DeleteThumbs

Syntax:	<code>VARIANT_BOOL DeleteThumbs(long nStartPage, long nEndPage);</code>
Parameters:	<code>long nStartPage, nEndPage</code> : These two values specify the interval of pages. The thumbnails of these pages will be removed from the file.
Return value:	<code>VARIANT_TRUE</code> if the operation succeeded.
Remarks:	Remove embedded thumbnail images of the pages.
Example:	

Method CreateThumbs

Syntax:	<code>VARIANT_BOOL CreateThumbs(long nStartPage, long nEndPage);</code>
Parameters:	<code>long nStartPage, nEndPage</code> : These two values specify the interval of pages. The thumbnails of these pages will be added to the file.
Return value:	<code>VARIANT_TRUE</code> if the operation succeeded.
Remarks:	Create embedded thumbnail images of the pages or a range of pages.
Example:	

Method GetInfo

Syntax: `BSTR GetInfo(BSTR szInfoKey);`

Parameters: `BSTR szInfoKey`: Specifies the key whose value is required.

Return value: Retrieves the value of the specified key.

Remarks: Use this method to retrieve the document information values like "Author", "Title", etc.

Example:

Method SetInfo

Syntax: `VARIANT_BOOL SetInfo(BSTR szInfoKey, BSTR szBuffer);`

Parameters: `BSTR szInfoKey`: Specifies the key whose value has to be set.
`BSTR szBuffer`: Specifies the value.

Return value: `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method to set the different document information parameters, like "Author", "Title", etc.

Example:

Method GetPermanentID

Syntax: `BSTR GetPermanentID();`

Parameters: None

Return value: Returns the permanent ID of the document.

Remarks:

Example:

Method GetInstanceID

Syntax: `BSTR GetInstanceID();`

Parameters: None

Return value: Returns the instance ID of the document.

Remarks:

Example:

Method CropPages

Syntax:	<code>VARIANT_BOOL CropPages(long nStartPage, long nEndPage, short nEvenOrOddPagesOnly, IDispatch* iRect);</code>
Parameters:	<code>long nStartPage</code> : Specifies the first page to be cropped. <code>long nEndPage</code> : Specifies the last page to be cropped. <code>short nEvenOrOddPagesOnly</code> : 0 – Crop all pages in the range 1 – Crop odd pages in the range 2 – Crop even pages in the range <code>IDispatch* iRect</code> : Specifies the cropping rectangle. The coordinates should be given in user space.
Return value:	VARIANT_TRUE if the operation succeeded.
Remarks:	Crops the pages in the given page range.
Example:	

Method AcquirePage

Syntax:	<code>IDispatch* AcquirePage(long nPage);</code>
Parameters:	<code>long nPage</code> : Specifies the index of the page to be acquired.
Return value:	Returns a reference to the specified PDF page.
Remarks:	Use this method to retrieve a reference to a document page. The reference can be used for different page operations. The reference is the IDispatch interface of a DDPAGE object.
Example:	Example 3 ; Example 4 ;

Method CreateTextSelect

Syntax:	<code>IDispatch* CreateTextSelect(long nPage, IDispatch* iRect);</code>
Parameters:	<code>long nPage</code> : Specifies the page where the selection should be created. <code>IDispatch* iRect</code> : Specifies the area of the text selection.
Return value:	Returns a reference to a text selection object.
Remarks:	Use this method to create a text selection based on a specific area of a page.
Example:	

IDVDoc interface

This interface is strongly related to the DDDoc interface. While DDDoc provides the file-level operations, this interface supplies methods to support viewing documents. Similarly to the other interface, you can open and close documents using DVDoc methods, but all of them control the document window and make that visible along with the application main window. The strong relationship between IDVDoc and DDDoc is represented with the method of GetDDoc that gives access to file level operations on the document opened in the document window.

Method GetDDoc

Syntax:	<code>IDispatch* GetDDoc()</code>
Parameters:	None
Return value:	Returns the IDispatch interface of the associated DDDoc object.
Remarks:	This method is used for getting access to the DDDoc object of the DVDoc object. You may need this method if you want to reach the properties of the DDDoc object or call its methods.
Example:	Example 3 ;

Method Open

Syntax:	<code>VARIANT_BOOL Open(BSTR szFullPath);</code>
Parameters:	<code>BSTR szFullPath</code> : Specifies the PDF file to be opened.
Return value:	Return <code>VARIANT_TRUE</code> if the file was opened.
Remarks:	Opens the specified file and displays it in a child window of the main application window.
Example:	Example 2 ; Example 3 ;

Method Close

Syntax:	<code>VARIANT_BOOL Close(long bNoSave);</code>
Parameters:	<code>long bNoSave</code> : Set it to 1 if you do not want to save the file; 0 value means that the application prompts for saving modifications.
Return value:	Return value is <code>VARIANT_TRUE</code> if the file was closed properly.
Remarks:	This method closes the document along with its window.
Example:	

Method PrintPages

Syntax:	<code>VARIANT_BOOL PrintPages(long nFirstPage, long nLastPage, long nPSLevel, long bBinaryOk, long bShrinkToFit);</code>
Parameters:	<p><code>nFirstPage</code>: The first page to be printed. It is zero-based. <code>nLastPage</code>: The last page to be printed. <code>nPSLevel</code>: If 2, PostScript Level 2 operators are used; if 3, PostScript Language Level 3 operators are also used. <code>bBinaryOk</code>: If 0, all data is encoded as 7-bit ASCII; if a positive number, binary data may be included in the PostScript program. <code>bShrinkToFit</code>: If a positive number, the page is shrunk to fit within the usable area of the printed page; if 0, it is not.</p>
Return value:	VARIANT_TRUE if the operation succeeded.
Remarks:	Displays the Print dialog and prints the specified range of pages.
Example:	

Method PrintPagesSilent

Syntax:	<code>VARIANT_BOOL PrintPagesSilent(long nFirstPage, long nLastPage, long nPSLevel, long bBinaryOk, long bShrinkToFit);</code>
Parameters:	<p><code>nFirstPage</code>: The first page to be printed. It is zero-based. <code>nLastPage</code>: The last page to be printed. <code>nPSLevel</code>: If 2, PostScript Level 2 operators are used; if 3, PostScript Language Level 3 operators are also used. <code>bBinaryOk</code>: If 0, all data is encoded as 7-bit ASCII; if a positive number, binary data may be included in the PostScript program. <code>bShrinkToFit</code>: If a positive number, the page is shrunk to fit within the usable area of the printed page; if 0, it is not.</p>
Return value:	VARIANT_TRUE if the operation succeeded.
Remarks:	Prints the specified ranges of pages without displaying the Print dialog box.
Example:	

Method GetTitle

Syntax:	<code>BSTR GetTitle();</code>
Parameters:	None
Return value:	Returns the title of the of the window associated with the document.
Remarks:	
Example:	

Method Maximize

Syntax: `VARIANT_BOOL Maximize(long bMaxSize);`

Parameters: `long bMaxSize`: If the value of the input parameter is greater than zero the document window is within the application main window.

Return value: `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method to maximize the size of the document window associated to the DVDoc object.

Example:

Method GetFrame

Syntax: `IDispatch* GetFrame();`

Parameters: None

Return value: Returns the IDispatch interface of a Rect object specifying the coordinates of the window associated with the document.

Remarks: The returned IDispatch interface refers to the `CPDFPlusRect` object that contains the coordinates of the document window.

Example:

Method IsValid

Syntax: `VARIANT_BOOL IsValid();`

Parameters: None

Return value: Returns `VARIANT_TRUE` if the document is a valid PDF document.

Remarks:

Example:

Method GetViewMode

Syntax: `long GetViewMode();`

Parameters: None

Return value: The return value indicates how the document is displayed.

Remarks: The possible return values can be the following:
`DDDontCare: 0` -- leave the view mode as it is.
`DDUseNone: 1` -- display without bookmark or thumbnails.
`DDUseThumbs: 2` -- display using thumbnails.
`DDUseBookmarks: 3` -- display using bookmarks.
`DDFullScreen: 4` -- display in full screen mode.

Example:

Method SetViewMode

Syntax: VARIANT_BOOL SetViewMode(long nType);

Parameters: long nType

Return value: VARIANT_TRUE if the operation succeeded.

Remarks: Use this method to modify the view of the document. The possible values are listed under the `GetViewMode` method.

Example:

Method GetDVPageView

Syntax: IDispatch* GetDVPageView();

Parameters: None

Return value: The method returns the IDispatch interface of the DVPageView object.

Remarks: Use this method if you need access to methods in the IDVPageView interface.

Example:

Method SetTextSelection

Syntax: VARIANT_BOOL SetTextSelection(IDispatch* iDDTextSelect);

Parameters: IDispatch* iDDTextSelect: Reference to an IDTextSelect object that specifies the text selection in the document.

Return value: VARIANT_TRUE if the operation succeeded.

Remarks: Use this method to set the required text selection.

Example: [Example 3](#);

Method ShowTextSelect

Syntax: VARIANT_BOOL ShowTextSelect();

Parameters: None

Return value: VARIANT_TRUE if the operation succeeded.

Remarks: Call this method to show the selected text.

Example: [Example 3](#);

Method ClearSelection

Syntax: `VARIANT_BOOL ClearSelection();`
Parameters: None
Return value: VARIANT_TRUE if the operation succeeded.
Remarks: Call this method to clear the text selection.
Example:

Method BringToFront

Syntax: `VARIANT_BOOL BringToFront();`
Parameters: None
Return value: VARIANT_TRUE if the operation succeeded.
Remarks: Makes to document window topmost among other document windows within the application window. This window gets the focus.
Example:

Method SetTitle

Syntax: `VARIANT_BOOL SetTitle(BSTR szTitle);`
Parameters: BSTR szTitle: Specifies the new title of the document window.
Return value: VARIANT_TRUE if the operation succeeded.
Remarks: Use this method to change the title of the document window.
Example:

Method PrintPagesEx

Syntax:	<code>VARIANT_BOOL PrintPagesEx(long nFirstPage, long nLastPage, long nPSLevel, long bBinaryOk, long bShrinkToFit, long bReverse, long bFarEastFontOpt, long bEmitHalftones, long iPageOption);</code>
Parameters:	<p><code>nFirstPage</code>: The first page to be printed. It is zero-based.</p> <p><code>nLastPage</code>: The last page to be printed.</p> <p><code>nPSLevel</code>: If 2, PostScript Level 2 operators are used; if 3, PostScript Language Level 3 operators are also used.</p> <p><code>bBinaryOk</code>: If 0, all data is encoded as 7-bit ASCII; if a positive number, binary data can be included in the PostScript program.</p> <p><code>bShrinkToFit</code>: If a positive number, the page is shrunk to fit within the usable area of the printed page; if 0, it is not.</p> <p><code>bReverse</code>: (PostScript printing only) If a positive number, print the pages in reverse order. If 0, print the pages in the regular order.</p> <p><code>bFarEastFontOpt</code>: (PostScript printing only) Set to a positive number if the destination printer has multi-byte fonts; set to 0 otherwise.</p> <p><code>bEmitHalftones</code>: (PostScript printing only) If a positive number, emit the halftones specified in the document. If 0, do not.</p> <p><code>iPageOption</code>: Pages in the range to print. Must be one of: DDAllPages (-3), DDEvenPagesOnly (-4), or DDOddPagesOnly (-5).</p>
Return value:	Returns <code>VARIANT_TRUE</code> if the operation succeeded.
Remarks:	Displays the Print dialog box and prints the specified pages.
Example:	

Method PrintPagesSilentEx

Syntax: VARIANT_BOOL PrintPagesSilentEx(long nFirstPage, long nLastPage, long nPSLevel, long bBinaryOk, long bShrinkToFit, long bReverse, long bFarEastFontOpt, long bEmitHalftones, long iPageOption);

Parameters:

- long nFirstPage: The first page to be printed. It is zero-based.
- long nLastPage: The last page to be printed.
- long nPSLevel : If 2, PostScript Level 2 operators are used; if 3, PostScript Language Level 3 operators are also used.
- long bBinaryOk: If 0, all data is encoded as 7-bit ASCII; if a positive number, binary data may be included in the PostScript program.
- long bShrinkToFit: if a positive number, the page is shrunk to fit within the usable area of the printed page; if 0, it is not.
- long bReverse: (PostScript printing only) If a positive number, print the pages in reverse order. If 0, print the pages in the regular order.
- long bFarEastFontOpt: (PostScript printing only) Set to a positive number if the destination printer has multi-byte fonts; set to 0 otherwise.
- long bEmitHalftones: (PostScript printing only) If a positive number, emit the halftones specified in the document. If 0, do not.
- long iPageOption: Pages in the range to print. Must be one of: DDAllPages (-3), DDEvenPagesOnly (-4), or DDOddPagesOnly(-5).

Return value: Returns VARIANT_TRUE if the operation succeeded.

Remarks: Prints the specified pages silently without displaying the Print dialog box.

Example:

IDDPPage Interface

This interface represents a page of a PDF document. Certain page level operations – cropping, rotating pages – and managing annotations and text highlights can be performed by its methods. The interface can be retrieved by the method `AcquirePage` of the `DDDoc` interface.

Method GetRotate

Syntax:	<code>short GetRotate();</code>
Parameters:	None
Return value:	The method returns the rotation value of the page in degrees. The possible values are listed under method <code>SetRotate</code>
Remarks:	Use this method to determine the rotation degree of the page.
Example:	

Method SetRotate

Syntax:	<code>VARIANT_BOOL SetRotate(short nRotate);</code>
Parameters:	<code>short nRotate</code> : Rotates the page with the given degree. The acceptable values are 0, 90, 180 and 270.
Return value:	The method returns <code>VARIANT_TRUE</code> if the operation succeeded.
Remarks:	This method influences the view of the document page. To make changes permanent, the PDF file has to be saved.
Example:	

Method GetNumAnnots

Syntax:	<code>long GetNumAnnots();</code>
Parameters:	None
Return value:	This method returns the number of annotations on the given page.
Remarks:	The method is used typically when the annotations have to be enumerated.
Example:	Example 4 ;

Method GetNumber

Syntax:	<code>long GetNumber();</code>
Parameters:	None
Return value:	This method retrieves the index of the page. Note that the page indexes start from zero.
Remarks:	Certain methods expect a page index as input parameter. Use this method to retrieve the appropriate value for the given page.
Example:	

Method CropPage

Syntax:	<code>VARIANT_BOOL CropPage(IDispatch* iRect);</code>
Parameters:	<code>IDispatch* iRect</code> : The IDispatch interface to a Rect object that determines the size of the crop area of the page.
Return value:	The method returns <code>VARIANT_TRUE</code> if the operation succeeded.
Remarks:	Use the method to adjust the visible area of the page. The coordinates have to be given in user space.
Example:	

Method GetAnnot

Syntax:	<code>IDispatch* GetAnnot(long nIndex);</code>
Parameters:	<code>long nIndex</code> : Specifies the index of the requested annotation on the page.
Return value:	The method retrieves the IDispatch interface of the selected annotation (<code>DDAnnot</code>) object.
Remarks:	Use this method to retrieve the given page annotation. The properties of the annotation can be accessed through the returned interface pointer.
Example:	

Method AddAnnot

Syntax:	<code>VARIANT_BOOL AddAnnot(long nIndexAddAfter, IDispatch* iDDAnnot);</code>
Parameters:	<p><code>long nIndexAddAfter</code>: Specifies the index of the new annotation. The annotation will be inserted after the specified index. Use a value of -1 in case of insertion at the very first position or if the given annotation is the first one on the page.</p> <p><code>IDispatch* iDDAnnot</code>: The reference to the annotation to be inserted. Note that the <code>DDAnnot</code> object is not a creatable one. You can get this interface pointer using other methods of the IDDPPage interface.</p>
Return value:	The method returns <code>VARIANT_TRUE</code> if the operation succeeded.
Remarks:	Use the method to add an annotation to the page. Typical usage of the method is to copy or move existing annotations to another page. The annotation can also be retrieved from another document. To create a new annotation, use the method <code>AddNewAnnot</code> .
Example:	

Method AddNewAnnot

Syntax:	<code>IDispatch* AddNewAnnot(long nIndexAddAfter, BSTR szSubType, IDispatch* iRect);</code>
Parameters:	<p><code>long nIndexAddAfter</code>: Specifies the index of the new annotation. The annotation will be inserted after the specified index. Use a value of -1 in case of insertion at the very first position or if the given annotation is the first one on the page.</p> <p><code>BSTR szSubType</code>: Specifies the type of annotation. It supports the "Text" subtype only.</p> <p><code>IDispatch* iRect</code>: The IDispatch interface of a <code>Rect</code> object specifies the location and the size of the annotation.</p>
Return value:	The method returns the IDispatch interface of the new <code>DDAnnot</code> object, if the insertion succeeded. You can use this value to set additional properties of the annotation (e.g. color). The return value is <code>NULL</code> if the operation failed.
Remarks:	Use this method to create a new annotation on the given page.
Example:	Example 4;

Method GetDoc

Syntax:	<code>IDispatch* GetDoc();</code>
Parameters:	None
Return value:	This method returns the IDispatch interface of the <code>DDDoc</code> object associated to the <code>DDPage</code> object.
Remarks:	
Example:	

Method GetAnnotIndex

Syntax: `long GetAnnotIndex(IDispatch* iDDAnnot);`

Parameters: `IDispatch* iDDAnnot`: The IDispatch interface of the DDAnnot object whose index has to be retrieved.

Return value: The method returns the zero-based index of the given annotation.

Remarks:

Example:

Method GetSize

Syntax: `IDispatch* GetSize();`

Parameters: None

Return value: The method returns the IDispatch interface of a Point object.

Remarks: Use this method to retrieve the size of the page measured in points.

Example:

Method CreateWordHilite

Syntax: `IDispatch* CreateWordHilite(IDispatch* iHiliteList);`

Parameters: `IDispatch* iHiliteList`: Specifies the series of characters to be highlighted.

Return value: The method returns the IDispatch interface of a DDTTextSelect object.

Remarks: The method is used for highlighting a series of words on the given page. The DDHiliteList object may specify several series of words.

Example: [Example 3](#);

Method CreatePageHilite

Syntax: `IDispatch* CreatePageHilite(IDispatch* iHiliteList);`

Parameters: `IDispatch* iHiliteList`: Specifies the series of characters to be highlighted.

Return value: The method returns the IDispatch interface of a DDTTextSelect object.

Remarks: The method is used for highlighting a series of characters on the given page. The DDHiliteList object may specify several series of characters.

Example: The example provided for the method CreateWordHilite is applicable with the appropriate changes. See [Example 3](#)

IDDAnnot interface

This interface provides access to properties of the annotations, like color, date, title, content, coordinates measured in user space, open state, subtype. Because the list of annotations is maintained by the DDPPage object, the interface is accessible by method DDPPage::GetAnnot. The annotations are identified by their index within the given page. You can also add annotations to a page using the DDPPage object. Please note that only annotations having subtype “Text” are creatable.

Method GetColor

Syntax: `long GetColor();`

Parameters: None

Return value: The method returns the color of the annotation in RGB values. The values are represented in this form: 0xBBGGRR. So, the value of 0xFF (255) represents a light red color, 0xFF00 (65280) is a light green value and 0xFF0000 (16711680) is a light blue.

Remarks:

Example:

Method SetColor

Syntax: `VARIANT_BOOL SetColor(long nRGBColor);`

Parameters: `long nRGBColor:` Specifies the color of the annotation

Return value: The method returns VARIANT_TRUE if the operation succeeded.

Remarks:

Example: [Example 4;](#)

Method GetDate

Syntax: `IDispatch* GetDate();`

Parameters: None

Return value: The method returns an IDispatch interface to a Time object.

Remarks: Use the properties of the Time object to retrieve the date of the annotation.

Example:

Method SetDate

Syntax: `VARIANT_BOOL SetDate(IDispatch* ITime);`

Parameters: `IDispatch* ITime`: Specifies the creation time of the annotation.

Return value: The method returns `VARIANT_TRUE` if the operation succeeded.

Remarks:

Example:

Method GetRect

Syntax: `IDispatch* GetRect();`

Parameters: None

Return value: The method returns the IDispatch interface for the Rect object

Remarks: The method returns the bounding rectangle of the annotation. Use the properties of the Rect object to determine the location of the annotation on the page.

Example:

Method SetRect

Syntax: `VARIANT_BOOL SetRect(IDispatch* iRect);`

Parameters:

Return value: The method returns `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method to set or change the location and size of the annotation.

Example:

Method GetTitle

Syntax: `BSTR GetTitle();`

Parameters: None

Return value: The method returns the title of the annotation.

Remarks:

Example:

Method SetTitle

Syntax: `VARIANT_BOOL SetTitle(BSTR szTitle);`

Parameters:

Return value: The method returns `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method to set or change the title of the annotation.

Example:

Method GetSubType

Syntax: `BSTR GetSubtype();`

Parameters: None

Return value: The method returns the subtype property of the annotation.

Remarks:

Example:

Method IsValid

Syntax: `VARIANT_BOOL IsValid();`

Parameters: None

Return value: The method returns `VARIANT_TRUE` if the annotation is valid.

Remarks: The method verifies the annotation object and checks whether or not it is deleted.

Example:

Method IsOpen

Syntax: `VARIANT_BOOL IsOpen();`

Parameters: None

Return value: The method returns `VARIANT_TRUE` if the text annotation is open.

Remarks: This method verifies whether or not a text annotation is open.

Example:

Method SetOpen

Syntax: `VARIANT_BOOL SetOpen(long bIsOpen);`

Parameters:

Return value: The method returns `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method to open a text annotation.

Example:

Method GetContents

Syntax: `BSTR GetContents();`

Parameters: None

Return value: The method retrieves the content of an annotation.

Remarks:

Example:

Method SetContents

Syntax: `VARIANT_BOOL SetContents(BSTR szContents);`

Parameters:

Return value: The method returns `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method to set or modify the content of the annotation.

Example: [Example 4;](#)

Method IsEqual

Syntax: `VARIANT_BOOL IsEqual(IDispatch* iDDAnnot);`

Parameters:

Return value: The method returns `VARIANT_TRUE` if the two compared annotations are equal.

Remarks: The method determines whether or not the current annotation is the same as a specified one.

Example:

Method Perform

Syntax: VARIANT_BOOL Perform(IDispatch* IDVDoc);

Parameters:

Return value: The method returns VARIANT_TRUE if the operation succeeded.

Remarks: The method refers to Link type annotations. It performs the action associated with the Link object.

Example:

IDVPageView

Similarly to the DVDoc interface, this interface provides a set of methods related to viewing PDF pages. The interface is not creatable, but accessible via the method GetDVPageView in the DVDoc interface.

Method DoGoBack

Syntax:	VARIANT_BOOL DoGoBack();
Parameters:	None
Return value:	The method returns VARIANT_TRUE if the operation succeeded.
Remarks:	The application tracks the views displayed in each document window. Use this method to step back to the previous view of the document window.
Example:	

Method DoGoForward

Syntax:	VARIANT_BOOL DoGoForward();
Parameters:	None
Return value:	The method returns VARIANT_TRUE if the operation succeeded.
Remarks:	The application tracks the views displayed in each document window. Use this method to step forward to the next view of the document window.
Example:	

Method GetDVDoc

Syntax:	IDispatch* GetDVDoc();
Parameters:	None
Return value:	The method returns the IDispatch interface of the DVDoc object displayed in the page view.
Remarks:	Use this method to retrieve the DVDoc object displayed in the view.
Example:	

Method GetDoc

Syntax:	<code>IDispatch* GetDoc();</code>
Parameters:	None
Return value:	The method returns the IDispatch interface of the DDDoc object displayed in the page view.
Remarks:	Use this method to retrieve the DDDoc object displayed in the view.
Example:	

Method GetPage

Syntax:	<code>IDispatch* GetPage();</code>
Parameters:	None
Return value:	The method returns the IDispatch interface of the DDPAGE object referring to the current PDF page displayed in the page view.
Remarks:	Use this method to retrieve the DDPAGE object displayed in the view.
Example:	

Method GetPageNum

Syntax:	<code>long GetPageNum();</code>
Parameters:	None
Return value:	The method returns the zero-based index of the current PDF page displayed in the page view.
Remarks:	Use this method to retrieve the index of the displayed page.
Example:	

Method GetZoom

Syntax:	<code>long GetZoom();</code>
Parameters:	None
Return value:	The method returns the current zoom factor as a percentage.
Remarks:	
Example:	

Method GetZoomType

Syntax: `short GetZoomType();`

Parameters: None

Return value: The method return a given zoom type.

Remarks: The possible return values are the following:

DVZoomNoVary (0)	A fixed zoom, such as 75%
DVZoomFitPage (1)	Fits the page in the window
DVZoomFitWidth (2)	Fits the page width to the window
DVZoomFitHeight (3)	Fits the page height to the window
DVZoomFitVisibleWidth (4)	Fits the visible content of the page into the window.

Example:

Method Goto

Syntax: `VARIANT_BOOL Goto(long nPage);`

Parameters: `long nPage`: Zero based index of the page

Return value: The method returns `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method to make a page specified with its index visible.

Example:

Method ScrollTo

Syntax: `VARIANT_BOOL ScrollTo(short nX, short nY);`

Parameters: `short nX, short nY`: Specifies the scrolling coordinates within the current page.

Return value: The method returns `VARIANT_TRUE` if the operation succeeded.

Remarks: Use this method to scroll the current page to the given position. The coordinates specify the left-top corner of the visible area in user space.

Example:

Method ZoomTo

Syntax: `VARIANT_BOOL ZoomTo(short nType, short nScale);`

Parameters: `short nType`: Specifies the type of the Zoom (see also the method `GetZoomType`)
`short nScale`: Specifies the scale of zoom as a percentage.

Return value: The method returns `VARIANT_TRUE` if the operation succeeded.

Remarks:

Example:

Method DevicePointToPage

Syntax: `IDispatch* DevicePointToPage(IDispatch* IPoint);`
Parameters: `IDispatch* IPoint`: Point to be converted.
Return value: The method returns the converted values.
Remarks: Conversion from device space to user space
Example:

Method PointToDevice

Syntax: `IDispatch* PointToDevice(IDispatch* IPoint);`
Parameters: `IDispatch* IPoint`: Point to be converted.
Return value: The method returns the converted values.
Remarks: Conversion from user space to device space.
Example:

Method GetAperture

Syntax: `IDispatch* GetAperture();`
Parameters: **None**
Return value: The method returns the IDispatch interface of a Rect object.
Remarks: The Rect object specifies the visible area of the page in user space.
Example:

IHiliteList

The HiliteList object is designed to specify parts of the text within a PDF document to be selected. The elements of the list may refer to individual characters or words depending on the usage of the object.

Method Add

Syntax:	<code>VARIANT_BOOL Add(short nOffset, short nLength);</code>
Parameters:	<code>short nOffset</code> : Specifies the index of the first object (character or word) to be highlighted. <code>short nLength</code> : Specifies the number of objects to be highlighted.
Return value:	The method returns <code>VARIANT_TRUE</code> if the operation succeeded.
Remarks:	The specified selections refer to characters if the HiliteList object is used by method <code>CreatePageHilite</code> of the <code>DDPage</code> object. They refer to words in case of the method <code>CreateWordHilite</code> .
Example:	Example 3 ;

IDDTTextSelect

The interface provides some read-only properties for the text selection on the given PDF page.

Method Destroy

Syntax:	<code>VARIANT_BOOL Destroy();</code>
Parameters:	None
Return value:	The method returns <code>VARIANT_TRUE</code> if the operation succeeded.
Remarks:	Use this method to delete the current text selection.
Example:	

Method GetBoundingRect

Syntax:	<code>IDispatch* GetBoundingRect();</code>
Parameters:	None
Return value:	The method returns a <code>Rect</code> object.
Remarks:	The return value of the method provides the bounding rectangle of the current text selection.
Example:	

Method GetNumText

Syntax: `long GetNumText();`

Parameters: None

Return value: The method returns the number of selected text objects.

Remarks:

Example:

Method GetPage

Syntax: `long GetPage();`

Parameters: None

Return value: The method returns the index of the page where the selection is being placed.

Remarks:

Example:

Method GetText

Syntax: `BSTR GetText(long nTextIndex);`

Parameters: `long nTextIndex:` Specifies the index of the selected text.

Return value: The method returns the selected text.

Remarks: Use this method to retrieve the text specified with the given index in the highlighted content.

Example:

Helper interfaces

The following interfaces are designed for retrieving special properties of different objects.

IRect interface

This interface is designed to get coordinates of a rectangular object. The interface provides four properties for the left, right, top and bottom coordinates. The type of these properties is long. You have full access (read/write) to all four properties. See also [Example 4](#)

ITime interface

This interface defines a Time object having seven properties: year, month, day, hour, minute, second and millisecond. You have full access (read/write) to all four properties.

IPoint interface

This interface defines a point having two properties: the X and Y coordinates. You have full access (read/write) to these properties. This interface is also used for referring to the size of an object. In this case, the property X refers to the width and Y to the height of the object.

Examples

These examples are designed in Visual C++ code. The examples are easily applicable to other programming languages supporting OLE automation.

Example 1 – Inserting pages into a document

The following example demonstrates how pages can be inserted from one PDF document into another one. The sample can also be applied to the methods `ReplacePages` and `Move Pages` of the `IDDDoc` interface

The referenced methods

Creating Application interface

Creating DDDoc interface

[IDDDoc::Open](#)

[IDDDoc::InsertPages](#)

[IDDDoc::Save](#)

[IDDDoc::Close](#)

[IApp::Exit](#)

```
CPDFPlusApp PDFApp;
if( !PDFApp.CreateDispatch( __uuidof( App ) ) )
{
    return E_FAIL;
}

CPDFPlusDDDoc ddDocTarget;
ddDocTarget.CreateDispatch( __uuidof( DDDoc ) );

if( 0 == ddDocTarget.Open( _T( "mydoc.pdf" ) ) )
{
    PDFApp.Exit();
    return E_FAIL;
}

CPDFPlusDDDoc ddDocSource;
ddDocSource.CreateDispatch( __uuidof( DDDoc ) );

if( 0 == ddDocSource.Open( _T( "pdfpages.pdf" ) ) )
{
    ddDocTarget.Close();
    PDFApp.Exit();
    return E_FAIL;
}

// Insert the first two pages of the source document into the 8th position
// of the target document
if( 0 != ddDocTarget.InsertPages( 8, ddDocSource, 0, 2, FALSE ) )
{
    ddDocTarget.Close();
    ddDocSource.Close();
    PDFApp.Exit();
    return E_FAIL;
}

// Save the modifications using DDDoc object
if( 0 == ddDocTarget.Save( DDSaveFull, "mydoc.pdf" ) )
{
```

```
//Saving the file failed  
}  
ddDocTarget.Close();  
ddDocSource.Close();  
PDFApp.Exit();
```

Example 2 – Using the MenuItemExecute method of the IApp interface

The following example opens a PDF document and creates dynamic form controls on it using the Auto-Form Typer feature.

The referred methods

Creating Application interface

Creating DVDoc interface

[IDVDoc::Open](#)

[IApp::MenuItemExecute](#)

[IApp::MenuItemIsEnabled](#)

[IApp::CloseAllDocs](#)

[IApp::Exit](#)

```
CPDFPlusApp PDFApp;  
CPDFPlusDVDoc dvOpenDoc;  
  
if( !PDFApp.CreateDispatch(__uuidof(App)) )  
    return E_FAIL;  
  
dvOpenDoc.CreateDispatch(__uuidof(DVDoc));  
  
if( dvOpenDoc.Open( CString("mydoc.pdf") ) == 0 )  
{  
    // File could not be opened  
    PDFApp.Exit();  
    return E_FAIL;  
}  
  
if( !PDFApp.MenuItemIsEnabled( _T("nuance:AutoFormTyper")) )  
{  
    if( !PDFApp.MenuItemExecute( _T("nuance:AutoFormTyper")) )  
    {  
        // Failed to perform the action  
        PDFApp.CloseAllDocs();  
        PDFApp.Exit();  
        return E_FAIL;  
    }  
}  
  
PDFApp.CloseAllDocs();  
PDFApp.Exit();
```

Example 3 – Highlighting words in a PDF document

This sample demonstrates how words can be selected within a page of a PDF document

The referred methods

Creating Application interface

Creating DVDoc interface

Creating HiliteList interface

Nuance PDF Converter Professional 5.0 – OLE Automation interface documentation
Examples

[IDVDoc::Open](#)

[IDDDoc::GetDDDoc](#)

[DDDoc::AcquirePage](#)

[IHiliteList::Add](#)

[DDPage::CreateWordHilite](#)

[DVDoc::SetTextSelection](#)

[DVDoc::ShowTextSelection](#)

```
CPDFPlusApp PDFApp;
if( !PDFApp.CreateDispatch(__uuidof(App)) )
    return E_FAIL;

//Open the PDF document
CPDFPlusDvDoc dvOpenDoc;
dvOpenDoc.CreateDispatch(__uuidof(DVDoc));
if( dvOpenDoc.Open( "mytest.pdf" ) == 0 )
{
    //"ERROR: can't open test file\n"
    PDFApp.Exit();
    return E_FAIL;
}

//Get the associated DDDoc object
LPDISPATCH lpdisp = dvOpenDoc.GetDDDoc();
CPDFPlusDDDoc ddDoc(lpdisp);

//Acquire the page to be highlighted
lpdisp = ddDoc.AcquirePage(0);
CPDFPlusDDPage ddPage(lpdisp);

//Create the list of highlights
CPDFPlusHiliteList hl;
hl.CreateDispatch(__uuidof(HiliteList));
hl.Add(12,1);    //12th word should be highlighted
hl.Add(52,2);   //and 2 additional words from the 52nd position

//Create the word highlights and display them
lpdisp = ddPage.CreateWordHilite(hl);
CPDFPlusDDTextSelect ts(lpdisp);
dvOpenDoc.SetTextSelection(ts);
dvOpenDoc.ShowTextSelect();
//The highlight list may refer to series of characters as well.
//Use method ddPage.CreatePageHilite to select series of
// characters
```

Example 4 – Adding annotations

This sample provides a code for adding a new annotation to a given page of the document and setting its properties.

The referred methods

Creating Application interface

Creating DVDoc interface

[DDPage::AcquirePage](#)

[Rect properties](#)

[DDPage::GetNumAnnots](#)

[DDPage::AddNewAnnot](#)

[DDAnnot::SetContents](#)

[DDAnnot::SetColor](#)

```
long AppendTextAnnotation(CPDFPlusDDDoc ddDocOpen, long pageindex, RECT rc,
                        LPCTSTR strComment)
{
    // Acquiring the page of the document
    LPDISPATCH lpdisp = ddDocOpen.AcquirePage( pageindex )
    if (NULL == lpdisp)
    {
        return ERROR_INVALID_PAGEINDEX;
    }
    CPDFPlusDDPage ddPage(lpdisp);

    CPDFPlusRect rcAnnot;
    if (FALSE ==rcAnnot.CreateDispatch( __uuidof( Rect ) ))
    {
        ddPage.ReleaseDispatch();
        return ERROR_INTERNAL;
    }
    rcAnnot.SetLeft( rc.left );
    rcAnnot.SetTop( rc.top );
    rcAnnot.SetRight( rc.right );
    rcAnnot.SetBottom( rc.bottom );
    long index = GetNumAnnots()-1;

    lpDisp = ddPage.AddNewAnnot(index, _T("Text"), rcAnnot );
    if (NULL == lpdisp)
    {
        ddPage.ReleaseDispatch();
        return ERROR_ADD_ANNOT_FAILED;
    }

    CPDFPlusDDAnnot ddAnnot(lpDisp);
    if( VARIANT_FALSE == ddAnnot.SetContents(strComment))
    {
        ddPage.ReleaseDispatch();
        ddAnnot.ReleaseDispatch();
        return ERROR_SETCONTENT;
    }
    ddAnnot.SetColor(128);

    return 0;
}
```

Appendix

The menu names

These menu names can be used as input parameter for their method MenuItemExcute of the IApp interface.

Menu item	Menu name
File menu	
Open	Open
New/Blank PDF	BlankPdf
New/Form file	FromFile
New/Create from multiple files	ConvertMultipleFiles
New/Combine multiple files	CombineMultipleFiles
New/Overlay multiple files	OverlayMultipleFiles
New/From clipboard	FromClipboard
New Package	PackageMultipleFiles
Close	Close
Save	Save
Save As	SaveAs
Revert to Last Save	Revert
Email	Email
Open Revision	open Revision
Save Revision	save Revision
Imposition	ZImposition
Print	Print
Print with Comment	PrintComment
Exit	Quit
Edit menu	
Undo	Undo
Redo	Redo
Cut	Cut
Copy	Copy
Paste	Paste
Delete	Delete
Select All	SelectAll
Deselect All	DeselectAll
Add Bookmark	AddBookmark
Find	Find
Search	Search
Preferences	Preference
View menu	
Go to/First Page	FirstPage
Go to/Previous Page	PrevPage
Go to/Next Page	NextPage
Go to/Last Page	LastPage
Go to/Go to Page	GoToPage
Go to/Previous View	PreviousView
Go to/Next View	NextView

Menu item	Menu name
Zoom/Zoom To	ZoomTo
Zoom/Zoom In	ZoomIn
Zoom/Zoom Out	ZoomOut
Zoom/Actual Size	ActualSize
Zoom/Fit Page	FitPage
Zoom/Fit Width	FitWidth
Zoom/Fit Visible	FitVisible
Zoom/Fit Height	FitHeight
Single Page	SinglePage
Continuous	Continuous
Continuous Facing	ContinuousFacing
Facing	Facing
Rotate Clockwise	RotateClockwise
Rotate Counterclockwise	RotateCounterclockwise
Toolbars/Casemap	nuance:CaseMap
Toolbars/Comment	Comment
Toolbars/Edit	UndoRedo
Toolbars/File	File
Toolbars/Form Tools	Form controls
Toolbars/Hummingbird	Hummingbird
Toolbars/Measure	Measure
Toolbars/Security	SecurityPolicy
Toolbars/Standard	BasicTools
Toolbars/View	View
Toolbars/Worksite	WorkSite
Toolbars/Zoom	Zoom
Toolbars/Reset Toolbars	ResetToolBar
Toolbars/Hide Toolbars	ShowToolBar
Toolbars/Show Button Labels/Default Labels	DefaultLabel
Toolbars/Show Button Labels/No Labels	NoLabel
Toolbars/Show Button Labels/All Labels	AllLabel
Toolbars/Customize Toolbars	CustomizeToolBar
Navigation Panels/Attachment	Attachment
Navigation Panels/Bookmark	Bookmark
Navigation Panels/Comments	Comments
Navigation Panels/Destinations	Destinations
Navigation Panels/Form Controls	nuance:FormTyperPanel
Navigation Panels/Model Tree	Z3DtreePanel
Navigation Panels/Pages	Thumbnail
Navigation Panels/Signatures	DigSigPanel
Clipart	ClipArt
Envelope	Delivery
Security	Security
Sign/Certify	Signature
Stamps	Stamps
Watermarks	Watermark
Menu Bar	HideMenuBar

Nuance PDF Converter Professional 5.0 – OLE Automation interface documentation
Appendix

Menu item	Menu name
Full Screen	FullScreen
Grid	Grid
Snap to Grid	Snap
Rulers	Rulers
Guides	Guides
Document menu	
Pages/New Page	NewPage
Pages/Insert Page	InsertPage
Pages/Extract Page	ExtractPage
Pages/Delete Page	DeletePage
Pages/Crop Page	CropPage
Pages/Rotate Page	RotatePage
Watermark/Add	Watermark:AddHeadersFootersWatermarks
Watermark/Delete	ZeonForm:DelWatermark
Insert Clip Art	ZEON:InsertClipArt
Sign & Certify/Sign Document	DigSigSignDoc
Sign & Certify/Certify Document	CertifyVisible
Sign & Certify/ Certify Document invisibly	CertifyInvisible
Sign & Certify/Sign Signature field	SignSignatureField
Sign & Certify/Sign/Certify Panel	SignCertifyPanel
Sign & Certify/Verify All Signatures	DigSigVerifyAll
Sign & Certify/Clear All Signature Fields	ClearAllField
Sign & Certify/View Signed Version	DigSigViewVersion
Sign & Certify/Compare Signed Version to Current Version	DigSigCompare
Security /Secure Document	ShowSecurityPanel
Security /Security Properties	SecurityProperty
Security /Remove Security Settings	RemoveSecuritySettings
Security /Secure Delivery	Delivery
Trusted Identities	Trusted Identity
Manage Digital IDs	Security Settings
Redaction/Mark Redaction	MarkRedaction
Redaction/Redaction Tool Properties	RedactionToolProperties
Redaction/Search and Redaction	RedactionSearch
Redaction/Apply Redaction	RedactionApply
Inspect Document	ExamineDoc
Bates Stamping/Add	ZEON:AddBatesStamp
Bates Stamping/Remove	ZEON:RemoveBatesStamp
Compare documents	ZEON:CompareDoc
Split document	Zeon:FileSplit
JavaScript /JavaScript Console	JavaScript Console
JavaScript /Document JavaScripts	Document JavaScripts
JavaScript /Set Document Actions	Set Document Actions
Create Full Text Indexes	Catalog
Embedded Index	Embed_Index
Document Properties	DocumentProperties

Comments menu

Menu item	Menu name
Note	Note
Text Box	FreeText
Callout	CallOut
Text Markups/Text Markup Tool	Markup
Text Markups/Insert Text at Cursor	Insert
Text Markups/Replace Selected Text	Replace
Text Markups/Add Note to Selected Text	NoteText
Text Markups/Highlight Selected Text	HighlightText
Text Markups/Underline Selected Text	UnderlineText
Text Markups/Cross Out Text for Deletion	CrossoutText
Stamp	Stamp
Highlight	Highlight
Cross-out	CrossOut
Underline	UnderLine
Draw Tools/Line	Line
Draw Tools/Rectangle	Rectangle
Draw Tools/Pencil	Pencil
Draw Tools/Oval	Oval
Draw Tools/Polygon	Polygon
Draw Tools/Polygon Line	PolyLine
Draw Tools/Cloud	Cloudy
Draw Tools/Eraser	Erase
Draw Tools/Hammer	Hammer
Attach a File	AttachFile
Attach a Sound	AttachSound
Create Comment Summary	CreateCommentSummary
Export Comments	ExportComment
Import Comments	ImportComment
Migrate Comments	MigrateComment
Search Comments	SearchComments
Show Comments Panel	ShowCommentsPanel
Forms menu	
Button tool	Form:Button
Check Box Tool	Form:Checkbox
Radio Button Tool	Form:RadioButton
Combo Box Tool	Form:Combobox
List Box Tool	Form:Listbox
Text Field Tool	Form:Textbox
Digital Signature Tool	Form:Signature
Set Tab Order	Form:SetOrder
Set Field Calculation Order	ZeonForm:CalcOrder
Highlight Form Fields	Form:Sep
Reset Form Fields	Form:Highlight
Import Data to Form	ZeonForm:ImportFDF
Export Data from Form	ZeonForm:ExportFDF
Form Typer	nuance:AutoFormTyper
Tools menu	

Nuance PDF Converter Professional 5.0 – OLE Automation interface documentation
Appendix

Menu item	Menu name
Standard/ Hand	Hand
Standard/ Select Text	SelectText
Standard/ Select Image	SelectImage
Standard/ Select Area	SelectArea
Standard/ Select Object	SelectObject
Standard/ Link	LinkTool
Standard/Crop	CropTool
Zoom/Zoom In	ZoomInTool
Zoom/Zoom Out	ZoomOutTool
Zoom/ Dynamic Zoom	ZoomDynamic
Zoom/ Loupe	Loupe
Advanced/Movie Tool	Zeon:Movie
Advanced/Sound Tool	Zeon:Sound
Advanced/3D Tool	3DTool
Measuring/Distance Tool	Zeon:Distance
Measuring/Perimeter Tool	Zeon:Perimeter
Measuring/Area Tool	Zeon:Area
Optimize PDF	nuance:Optimize
Tag PDF	nuance:Retag
Make Searchable PDF	&Make Searchable PDF
Read PDF/Read Current Page	nuance:TTS_PAGE
Read PDF/Read to End of Document	nuance:TTS_DOC
Read PDF/Read to File	nuance:TTS_FILE
Read PDF/Prev. para	nuance:TTS_PREVPARA
Read PDF/Next Para	nuance:TTS_NEXTPARA
Read PDF/Prev. Page	nuance:TTS_PREVPAGE
Read PDF/Next Page	nuance:TTS_NEXTPAGE
Read PDF/Pause	nuance:TTS_PAUSE
Read PDF/Stop	nuance:TTS_STOP
Read PDF/Properties	nuance:TTS_PROP
Touchup/Touchup Text	TouchUpTextTool
Touchup/Touchup Object	TouchUpObjectTool
Favorites menu	
Add To Favorites	AddToFavorites
Manage Favorites	ManageFavorites
Window menu	
New Window	NewWindow
Cascade	Cascade
Tile/Horizontally	TileHorizontal
Tile/Vertically	TileVertical
Close All	CloseAll
Horizontal Split	SplitHorz
Vertical Split	SplitVert
Quad Split	SpreadsheetSplit