

PDF-XChange Driver API SDK

© 2008 Tracker Software Products Ltd

Tracker Software Products Ltd.,
Units 1-3, Burleigh Oaks.
East Street. Turners Hill.
RH104PZ.
United Kingdom.

<http://www.docu-track.com>
[Mailto:Sales@docu-track.com](mailto:Sales@docu-track.com)

Table of Contents

Foreword	0
Part I Introduction	1
1 Welcome	1
2 Troubleshooting	2
3 Redistribution	2
Part II IPXCControlEx object	5
1 Property Printer	5
Part III Interface IPXCPrinter	7
1 Properties	7
Property Name	8
Property Option	8
Property Table Definition	9
Section Paper	10
Normal	11
n-Up	11
Booklet	12
List of Standard Paper Sizes	12
Section General	14
Section Compression	16
Subsection Compression.Color	16
Subsection Compression.Indexed	17
Subsection Compression.Mono	17
Section Graphics	18
Section Graphics.Color	18
Graphics.Color.Downsampling	18
Graphics.Color.Convert	19
Section Graphics.Indexed:	19
Graphics.Indexed.Downsampling	19
Graphics.Indexed.Convert	20
Section Graphics.MonoAndDrawings	20
Graphics.MonoAndDrawings.Downsampling	20
Graphics.MonoAndDrawings.Convert	21
Section Fonts	21
Section Security	22
Section DigiSig	24
Section Overlay	26
Section Links	28
Section Watermarks	29
Section HeadersFooters	29
Section DocInfo	31
Section Save	32
Section EMail	35
Section Optimization	36
Section Bookmarks	38
Section Saver	39
Macros	40
2 Methods	43
Method SetRegInfo	43

Method ResetDefaults	44
Method ApplyOptions	44
Method AddBookmarkItem	44
Method AddTextWatermark	47
Method AddImageWatermark	50
Method StorePrinterOptions	52
Method ReStorePrinterOptions	52
Method SetAsDefaultPrinter	53
Method RestoreDefaultPrinter	53
Method AddSMTPServer	53
3 Events	54
Asynchronous Events	55
Event OnStartDoc.....	55
Event OnStartPage.....	56
Event OnEndPage.....	56
Event OnEndDoc.....	57
Synchronous Events	57
Event OnDocSpooled.....	57
Event OnFileSaved.....	58
Event OnFileSent.....	58
Event OnError	59
4 Unsupported Property Declarations	59
Part IV Tracker Software Products	59
1 Contact Us	60
2 Products	61
3 Licensing	62
Index	67

1 Introduction

1.1 Welcome

The **PDF-XChange Drivers API SDK** is just one of a family of products we offer for a variety of Document creation, Manipulation and Viewing tasks - for various file formats including PDF and various Raster imaging formats.

Who should use the PDF-XChange Drivers API ?

Developers requiring the ability to create PDF documents from their own applications (*Clarion Developers - See below) or indeed managing the output of other file formats to PDF from 3rd party applications such as MS Office, AutoCAD, Internet Explorer etc, providing a set of 32/64 bit Virtual print drivers - the PDF-XChange Drivers API is an excellent choice. Creating arguable the smallest and most reliable facsimile in PDF format of the original document available today - at a very competitive price.

Other developer kits available for PDF Creation, Manipulation and Viewing from Tracker Software Products include, **PDF-Tools SDK**, **PDF-XChange PRO SDK** and the **PDF-XChange Viewer SDK**. For more information on all of our products please see :

Windows Support

From Version 4 we have dropped Drivers for all Windows versions prior to Windows 2000 - **should you still need to support Windows 9x/ME/NT**, we have continued to offer our highly successful PDF-XChange Version 3 product range, though it should be noted that NO further development will take place with our Version 3 product line.

The PDF-XChange Driver API ActiveX

Allows developers to control the properties of the PDF-XChange driver for the creation of PDF documents. Developers are able to harness the full power of our award-winning Driver by specifying parameters for PDF creation as used by many millions of contented End Users – directly from within the developer's own Software applications.

The PDF-XChange Driver API is supplied as an ActiveX component. Using its methods and properties all operations can be performed.

The sequence for creating a PDF file from a developer's application is:

1. Create a new unique (and temporary) virtual printer using the IPXCControlEx interface.
2. Using the IPXCPrinter interface returned by IPXCControlEx, set up all properties as required for PDF creation.
3. Print to the virtual printer created by step 1, either directly from developer's application or from any 3rd party application run from this application.
4. Close the created printer and destroy the temporary printer instance created.

*Clarion for Windows Developers generally do not need to use Virtual Print Drivers to output their

application reports using our products. Clarion uses a method to output reports to a Windows Metafile format for viewing that our PDF-Tools SDK is able to convert directly to PDF, whilst still retaining the reports formatting and text data - ensuring it is still a fully text searchable PDF file once created - we therefore recommend that unless Clarion developer's are converting 3rd party formats such as MS Office, HTML, AutoCAD etc to PDF - they use our PDF-Tools SDK and not the PDF-XChange Drivers API SDK.

1.2 Troubleshooting

In most environments the provided COM facilities are available transparently after the installation and registration are complete. However, for some 32-bit based IDE's (for example, Microsoft Visual Basic 6) running in 64-bit Windows, this may be problematic. In this event, you should manually reference the type library (pxccom.tlb) located in the PDF-XChange installation directory.

1.3 Redistribution

Redistribution.

We provide a comprehensive installation executable for developer's to distribute with their application and this is the ONLY method authorised for distribution, re-engineering or disassembly of the installation package provided is specifically and absolutely prohibited. Our installer contains all the required driver options required for all versions of Windows from Windows 2000 through to Vista 64 - and all points in between, both 32 and 64 bit!

Developers may either pack the provided installer within their own installation package - or provide separately and simply call from their own installer script using the extensive command line switches provided below.

The latest PDF-XChange Driver for redistribution purposes is always available from our web site - [here](#).

Or from this link : www.docu-track.com/PDFX4SA_sm.zip

Please note - it is required that you use the above installer and do not attempt to reverse engineer or de-compile and create your own - if you are not willing to adhere to this specific condition - you should not use the PDF-XChange Driver within your application.

Installer options and switches

Silently installing PDF-XChange.

This functionality is provided for users with multiple licenses and Developer SDK's - not for single (end) user licenses to be copied to multiple machines - which is illegal. A license is required for each machine to which PDF-XChange is installed - except for the Developer's SDK and evaluation use in demonstration mode.

Assuming you have a multiple user license - installing PDF-XChange to many workstations - it may be useful to be able to silently install without the need for any user interaction. This allows you to execute the install either from the command line with certain parameters and switches applied - or create a batch file (.bat file) and distribute this with the main installation executable for your users to run directly - alleviating the need for the System Administrator to visit each user and install.

Please note as with any system device - such as a printer, full Administrator privileges are required to install on later 'Win32' versions of Microsoft Windows.

Here is an example of the full command line string available:

Most items are optional unless otherwise stated

```
C:\PDFX4.exe /VERYSILENT /NORESTART /COMPONENTS="pdfSaver,PDF-XChange driver,
Help,Languagess" /DIR="C:\Program Files\PDF-XChange 4\" "/UserName:I
am" "/Organization:Tracker software" "/UserEmail:my@email.com"
```

Broken down into sections - these are the switches and what each does.

N.B. Please note that the indicated string content must begin and end with double quote marks (!)!

- **MAIN** install executable and path (**required**)

i.e.: `C:\PDFX4SA_sm.exe`

The is the full path to the installation executable, note it is the standard install executable - both silent and non silent installations are available from the single executable.

- **/VERYSILENT**

Designates that the installation once started should be silent and require no user interaction.

- **/SILENT**

Designates that the install should only request essential information from the user once started.

- **/NORESTART**

Once the install is complete - no reboot will occur - but beware - PDF-XChange does require a reboot before updates will take effect - not recommended.

- **/DIR= (required)**

the full path to install all required files to, created if not already available.

i.e. : `/DIR="C:\Program Files\PDF-XChange\"`

- **/Group= (Optional)**

The Windows 'Start Menu' folder in which to locate the Menu options provided when installing PDF-XChange - this will always be a 'Child' menu option of the default 'Programs Files' menu and this cannot be changed.

i.e. : `/GROUP="Tracker Software\PDF-XChange 4 Pro"`

- **/COMPONENTS=**

If this switch is specified only those components actually listed will be installed, otherwise all components are installed. The available components are:

- pdfSaver (Always Required)
- PDF-XChange driver (Always Required)
- Help
- Languagess (please note 'ss' is required)
- Autounload

i.e. : `/COMPONENTS="pdfSaver,PDF-XChange driver, Help,Languagess"`

- **/AutoUnload:nn**

If this option is specified, after the specified timeout (nn, in minutes) the PDFSaver4 module of the PDF-XChange driver will unload.

Useful when installed on a Windows Terminal Server or Citrix Metaframe server with multiple Users to minimise memory in use when PDF creation is not being used by a user - will automatically reload when required.

i.e. *"/Autounload:03"*

(Strictly speaking Autounload is not a component - but a default setting parameter mainly used in Citrix Metaframe or Terminal Services installations to minimise memory use on a server where PDFSaver4.exe could be loaded multiple times.)

- **/Key (Registration) - only to be used if you are also providing an END USER enabled license - not included as part of your Developers SDK License -EVER**

Allows the license key to be registered during silent installation - for live use. Without this information PDF-XChange will function in demo mode only.

i.e. : *"/Key:XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX"*

- **/UserName** Registration

Allows the registered username to be registered during silent installation.

i.e. : *"/Username"/UserName:MY Name"*

- **/Organization** Registration

Allows the Company details to be registered during silent installation.

i.e. : *"/Organization:Tracker software"*

- **/PDEFAULT= (Optional)**

Sets PDF-XChange to be the system default Printer - the default option without this entry is that PDF-XChange will **not** be the default printer.

- **/PName= (Optional)** printer renaming

Sets the PDF-XChange printer name as required - the default option without this entry is currently "PDF-XChange 4.0".

i.e. : *"/PName="PDF-XChange for my application"*

Please note that the string content must begin and end with double quote marks (").

- **/UserEmail** Registration

Allows the users email address details to be registered during silent installation.

i.e. : *"/UserEmail:my@email.com"*

- **/Lang**

Sets a language as the default language for the User Interface for all the installed programs and Print drivers (PDF-XChange, PDF-Tools, OFFice2PDF and the MS Office Addin's etc).

/LANG=language en (English) nl (Dutch) uk (Ukrainian) fr (French) gr (German) it (Italian) sp (Spanish) ko (Korean) ch (Chinese) jp (Japanese) fi (Finnish)

This may not be the complete list - please see the installation folder.

'C:\Program Files\Tracker Software\PDF-XChange 4 Pro\Languages' for the up to date list of supported language files.

Example (for German)

PDFX4SA_sm.exe [other parameters] /LANG=gr [Additional parameters]

Warning ! Failure to ensure that special switch characters such as quote marks (") and "/" are included in your command line or batch file, in the correct position - will result in failure to behave as required.

2 IPXCControlEx object

The **IPXCControlEx** ActiveX object is used to create a new **temporary** PDF-XChange printer instance, and returns the **IPXCPrinter** object, this in turn allows the developer's application to set and control the properties of the newly created printer instance. Once the job is completed the printer instance is destroyed.

2.1 Property Printer

Property IPXCControlEx.Printer

This property allows the creation of a new temporary printer based on the PDF-XChange 4.0 driver and returns the IPXCPrinter object associated with this printer.

(Please note at the time of writing a bug exists in MS Windows which may block this functionality from functioning in some versions of Windows)

Declaration:

C#

```
object get_Printer(  
    string pServerName,  
    string pPrinterName,  
    string pRegKey,  
    string pDevCode);
```

C++

```
HRESULT get_Printer(  
    [in] BSTR pServerName,  
    [in] BSTR pPrinterName,  
    [in] BSTR pRegKey,  
    [in] BSTR pDevCode,  
    [out,retval] VARIANT* ppPrinter);
```

VB

```
Property Printer(  
    pServerName As String,  
    pPrinterName As String,
```

```
pRegKey As String,  
pDevCode As String)
```

pServerName:

Reserved for future use. Should be a NULL or empty string.

pPrinterName:

Specifies the desired name for the printer to be created. If a printer of the same name already exists, then to ensure it is uniquely named, the new printer name will have a suffix automatically generated, based on the originally-passed value.

- **N.B.**

Differences between Licensed and Evaluation use.

The following 2 items are provided as part of your license purchase. If you are evaluating the **PDF-XChange Drivers API SDK**, all generated output will have evaluation watermarks printed on each page and these cannot be subsequently be removed. Once licensed you must re-compile your project, embedding your license strings provided within your code as outlined above and all new PDF files generated will be free of the evaluation watermarks placed on all evaluation output. All output created in evaluation mode will need to be recreated from the original source documents and materials.

pRegKey:

String containing the Developer registration key. If this registration key is not present or is invalid, the PDF-XChange demo label will be added to all PDF pages generated by the printer.

pDevCode:

String containing the developer code. If this developer code is not present or is invalid, the PDF-XChange demo label will be added to all PDF pages generated by the printer.

Return value

If successful, the function returns an **IPXCPrinter** object associated with newly created printer.

On failure the function returns a NULL.

Example

VB

```
Dim PDFPFactory As New PXCComLib.CPXCControlEx  
Dim WithEvents PDFPrinter As PXCComLib.CPXCPrinter  
  
Public Sub Form_Load()  
    Set PDFPrinter = PDFPFactory.Printer("", "PDF-XChange 4.0 Sample",  
    "<REG CODE>", "<DEV CODE>");  
End Sub  
  
Public Sub Form_Unload(Cancel As Integer)  
    Set PDFPrinter = Nothing  
End Sub
```

3 Interface IPXCPrinter

The **IPXCPrinter** ActiveX interface object is used to control all options of the PDF creation process. The object with this interface is returned by the **IPXCControlEx** ActiveX object. When printing is completed, the **IPXCPrinter** object should be released. Note that releasing this object will also remove the virtual printer associated with this object.

Events provided by **IPXCPrinter** object can be used to control printing process.

IPXCPrinter ActiveX interface provides the following Properties, Methods, and Events.

3.1 Properties

- Property Name returns the name of the printer associated with the object.
- Property Option is used to specify almost all the properties of the printer.

The following sections define the specific Property Options:

- Section Paper. Properties in this section allow you to define page layouts for the created PDF document.
 - Normal. Used to specify position and size of the page on the sheet.
 - n-Up. Defines how many and in which order pages should be placed on the sheet.
 - Booklet. Defines booklet (or book) creation parameters.
- Section General defines the PDF Specification to which the created PDF file should conform, and the options for the PDF viewing application.
- Section Compression specifies which compression options should be used for the PDF document's text and images.
 - Subsection Compression.color specifies compression for True color images.
 - Subsection Compression.Indexed specifies image compression for images with a color palette.
 - Subsection Compression.Mono specifies the compression method to be used to compress monochrome/black-&-white images.
- Section Graphics specifies how different types of images should be processed before inserting into a PDF document.
 - Section Graphics.color determines image downsampling and conversion options for True color images.
 - ❖ Graphics.color.Downsampling defines whether and how images should be "downsampled," i.e. have their resolution reduced.
 - ❖ Graphics.color.Convert defines whether color images should be converted to grayscale, monochrome, or black & white.
 - Section Graphics.Indexed determines image downsampling and conversion options for indexed images with palettes.
 - ❖ Graphics.Indexed.Downsampling defines whether and how images should be "downsampled," i.e. have their resolution reduced.
 - ❖ Graphics.Indexed.Convert defines whether color images should be converted to grayscale, monochrome, or black & white.

- Section Graphics.MonoAndDrawings determines settings for downsampling and converting monochrome images and black & white drawings.
 - ❖ Graphics.MonoAndDrawings.Downsampling defines whether and how images should be "downsampled," i.e. have their resolution reduced.
 - ❖ Graphics.MonoAndDrawings.Convert
- Section Fonts
- Section Security
- Section DigiSig
- Section Overlay
- Section Links
- Section Watermarks
- Section HeadersFooters
- Section DocInfo
- Section Save
- Section EMail
- Section Optimization
- Section Bookmarks
- Section Saver

3.1.1 Property Name

This property is a read only property and returns the name of the printer associated with the object.

Declaration:

C#

```
string Name;
```

C++

```
HRESULT get_Name([out, retval] BSTR* pName);
```

VB

```
Property Name As String
```

Please note that this name can be different from that passed in the Printer properties of the IPXCControlEx object, if a printer with the requested name already existed.

3.1.2 Property Option

This is a Write only parameterized property option and is used to specify almost all the properties of the PDF-XChange printer.

Declaration:**C#**

```
void set_Option(string pOptionName, object _p2);
```

C++

```
HRESULT put_Option ([in] BSTR pOptionName, [in] VARIANT _arg2);
```

VB

```
Property Option(pOptionName As String)
```

A detailed list of all available properties with detailed descriptions is listed in Properties. All properties are identified by a string for that pOptionName. The name of the property will also usually contain some parts separated by the dot (.) character. Values of these properties can be strings, integers, boolean or binary values (an array of bytes).

3.1.2.1 Property Table Definition

Each property section has a table detailing the possible property values.

The three columns of the tables provided are always:

- **Name.** The name of the property. This is what you assign the value to. For example: "**Paper.LayoutType**".
- **Possible Values.** These are the allowed values for the property specified in **Name**.
- **Description.** Describes the use or function of the property.

The possible values are described as numbers and corresponding strings, and either may optionally be used to set the value of the property.

For example: **0 (Normal)** means that this property can be set as the integer value **0** or as the name "**Normal**".

Therefore, both of these assignments are equivalent:

- **pPrinter.Option["Paper.LayoutType"] = 0**
- **pPrinter.Option["Paper.LayoutType"] = "Normal"**

Within the table column "Possible Values" if the entry = "Boolean Value" then an acceptable entry maybe any of the following:

For the value **0** any one of the following strings would also be acceptable: **No, False, Disabled**.

For the value **1** any one of the following strings would also be acceptable: **Yes, True, or Enable**.

For example, both of these two assignments are equivalent:

- **pPrinter.Option["Compress.Graphics"] = 1**
- **pPrinter.Option["Compress.Graphics"] = "Enable"**

Please note, names of properties and named values are **case sensitive**.

3.1.2.2 Section Paper

Properties in this section allow you to define page layouts for the created PDF document.

Name	Possible values	Description
LayoutType	0 (Normal) 1 (n-UP) 2 (Booklet)	Defines the layout type used for page creation of the PDF document. This property determines which other property sections are to be used: Normal, n-Up, or Booklet. Default value: 0 (Normal)
MirrorX	Boolean Value	Enables or disables horizontal mirroring of all drawings. Default value: False
MirrorY	Boolean Value	Enables or disables vertical mirroring of all drawings. Default value: False
SheetSizeIndex	From -2 to 41	Specifies the sheet used for PDF creation. Value -2 means custom sheet size which is defined by properties SheetWidth and SheetHeight. Value -1 means Auto sheet size. In this case sheet size is dependant on the layout type and page size. All other values are indices of the standard paper sizes defined in PDF-XChange. See List of Standard Paper Sizes. Default value: -1
SheetWidth	13-5080	Specifies the width and height of the sheet. Values are in tenths of millimetres and must be within the specified range. Default values: 2100 (width) and 2970 (height)
SheetHeight	13-5080	The maximum height/width of a PDF page are 200" x 200" or approx : 5080x5080 mm as defined in the PDF Specification.

3.1.2.2.1 Normal

When the **LayoutType** specified is **Normal**, the following properties are used to specify the position and page size on the sheet.

Name	Possible values	Description
LeftPos	Integer	These properties define the position of the top left corner of the page relative to the top left corner of the sheet. Values are in tenths of millimetres.
TopPos	Integer	These properties are used when both the Centre and Fit properties have False values. Default values: 0
PageScale	Integer	Specifies the scale level to use when positioning the page on the sheet. Value is in tenths of 1 percent. This property is only used if the Fit property is False. Default value: 1000
Center	Boolean Value	If this property is True, the page is centred on the sheet. Otherwise its position is determined by LeftPos and TopPos properties. This property is not used when the Fit property is True. Default value: True
Fit	Boolean Value	Specifies that the page should be 'fitted' to the sheet. Default value: True

3.1.2.2.2 n-Up

When the **LayoutType** specified is **nUP**, the following properties define how many and in which order, pages should be placed on the sheet.

Name	Possible values	Description
Nup	0 – one page per sheet 1 – two pages per sheet 2 – four pages per sheet	Defines the number of pages to be placed on one sheet. Default value: 0

	3 – six pages per sheet 4 – eight pages per sheet 5 – nine pages per sheet 6 – sixteen pages per sheet	
NupOrder	0 (AcrossFromLeft) 1 (DownFromLeft) 2 (AcrossFromRight) 3 (DownFromRight)	Defines the order in which pages should be placed on the sheet. Default value: 0 (AcrossFromLeft)
Borders	Boolean Value	Specifies if a border should be drawn around the pages on the sheet. Default value: False

3.1.2.2.3 Booklet

When the **LayoutType** chosen is **Booklet**, the following properties define the booklet creation parameters.

Name	Possible values	Description
BookletType	0 (Brochure) 1 (Book)	If the value of this property is Book, pages will be grouped in typographic gatherings by the number specified in the SheetsInBook property. Otherwise, the booklet will contain only one book with all the pages included. Default value: 0 (Brochure)
SheetsInBook	1-64	Defines the number of sheets in one book when the property BookletType has the value Book. Default value: 4
GutterSize	Positive integer	Defines the gutter size for the booklet. Value is specified in tenths of millimeters. Default value: 0

3.1.2.2.4 List of Standard Paper Sizes

List of predefined papers into PDF-XChange 4.0. Number, specified into column **Index**, can be used to specify the sheet size into property **Paper.SheetSizeIndex**.

Column Standard Index contains indexes for paper as it is defined into Windows API. Please note that

some paper definitions are specific for PDF-XChange and has no standard index.

Dimension column defines the size of paper for their standard orientation (portrait). Dimensions are specified in inches or in millimetres.

Index	Standard Index	Paper name	Dimension
0	1	Letter/ANSI A	8,5 x 11 "
1	3	Tabloid/ANSI B	11 x 17 "
2	-	ANSI Super B	13 x 19 "
3	24	ANSI C	17 x 22 "
4	25	ANSI D	22 x 34 "
5	26	ANSI E	34 x 44 "
6	-	ANSI F	28 x 40 "
7	5	US Legal	8,5 x 14 "
8	6	Statement	5,5 x 8,5 "
9	7	Executive	7,25 x 10,5 "
10	-	A0	841 x 1189 mm
11	-	A1	594 x 841 mm
12	62	A2	420 x 594 mm
13	8	A3	297 x 420 mm
14	9	A4	210 x 297 mm
15	11	A5	148 x 210 mm
16	-	B0	1000 x 1414 mm
17	-	B1	7070 x 1000 mm
18	-	B2	5000 x 7070 mm
19	42	B4 (ISO)	250 x 353,6 mm
20	12	B4 (JIS)	257 x 364 mm
21	34	B5	176 x 250 mm
22	13	B5 (JIS)	182 x 257 mm

23	14	Folio	8,5 x 13 "
24	15	Quarto	215 x 275 mm
25	18	Note	8,5 x 11 "
26	19	Envelope #9	3,875 x 8,875 "
27	20	Envelope #10	4,125 x 9,5 "
28	21	Envelope #11	4,5 x 10,375 "
29	23	Envelope #14	5 x 11,5 "
30	27	Envelope DL	110 x 220 mm
31	28	Envelope C5	162 x 229 mm
32	29	Envelope C3	324 x 458 mm
33	30	Envelope C4	229 x 324 mm
34	31	Envelope C6	114 x 162 mm
35	32	Envelope C65	114 x 229 mm
36	35	Envelope B6	125 x 176 mm
37	37	Envelope Monarch	3,875 x 7,5 "
38	38	Envelope Personal	3,625 x 6,5 "
39	-	18x24	18 x 24 "
40	-	24x36	24 x 36 "
41	-	30x42	30 x 42 "

3.1.2.3 Section General

Properties in this section define the PDF Specification to be applied to the created PDF file and the parameters for the PDF viewing application, which may be used to define the properties used when opening the document for Viewing.

Name	Possible values	Description
Specification	0 (Auto) 3 (1.3) 4 (1.4)	Defines the PDF Specification level to which the PDF file should conform. Value Auto (0), allows PDF-XChange to

	5 (1.5) 6 (1.6) 7 (1.7) -1 (PDF/A-1b)	choose the most efficient specification based on other specified options. When the value is PDF/A-1b(-1), PDF-XChange will create a PDF file which conforms to ISO standard PDF/A-1b. Please note that some PDF functions and content are not available/allowed when specifying PDF/A formatting. Default value: 0 (Auto)
PageLayout	0 (Default) 1 (ShowNone) 2 (Bookmarks) 3 (Thumbnails) 4 (FullScreen)	Default value: 0 (Default)
PageMode	0 (Default) 1 (ShowNone) 2 (Bookmarks) 3 (OptionalContent)	Default value: 0 (Default)
FullScreenMode	0 (Default) 1 (ShowNone) 2 (Bookmarks) 3 (OptionalContent)	Default value: 0 (Default)
MagnificationType	0 (Default) 1 (ActualSize) 2 (FitPage) 3 (FitWidth) 4 (FitHeight) 5 (Level)	Default value: 0 (Default)
MagnificationLevel	1-6400	Default value: 100
OpenToPage		Default value: 0
HideToolbars	Boolean Value	Default value: False
HideMenubar	Boolean Value	Default value: False
HideUI	Boolean Value	Default value: False

FitWindow	Boolean Value	Default value: False
CenterWindow	Boolean Value	Default value: False
ShowDocTitle	Boolean Value	Default value: False

3.1.2.4 Section Compression

The properties of this section specify which compression should be used for the PDF document's text and images.

Name	Possible values	Description
Graphics	Boolean Value	Specifies PDF document image compression. If this value is True, compression methods for different types of images are defined in the subsections for Color , Indexed and Mono images. Default value: True
Text	Boolean Value	If this property is True, textual content of the PDF pages will be compressed which reduces the size of the resulting PDF document. Default value: True

3.1.2.4.1 Subsection Compression.Color

Specifies compression for True color images.

Name	Possible values	Description
Enabled	Boolean Value	Default value: True
Method	0 (Auto) 1 (JPEG) 2 (ZIP) 3 (JPEGZIP) 4 (J2K) 5 (J2KZIP)	Specifies compression method to be used for compressing True color images. Please note, when the PDF/A specification is used for creating a PDF file, compression methods J2K and J2KZIP cannot be used. Default value: 0 (Auto)
JPEGQuality	1-100	Specifies the image quality when using JPEG compression (JPEG and JPEGZIP). Higher quality means better image definition, and a larger compressed stream,

		so a larger file. Default value: 75
--	--	--

3.1.2.4.2 Subsection Compression.Indexed

Specifies image compression for images with a color palette.

Name	Possible values	Description
Enabled	Boolean Value	If this property is True, Indexed images will be compressed by the method specified by property Method . Otherwise they will be stored into PDF without compression. Default value: True
Method	0 (Auto) 1 (PackBits) 2 (ZIP) 3 (LZW)	Defines the compression method to be used to compress images with palette. Please note, the compression method LZW cannot be used with the PDF/A output specification. Default value: 0 (Auto)

3.1.2.4.3 Subsection Compression.Mono

Defines the compression method to be used to compress Monochrome/black & white images.

Name	Possible values	Description
Enabled	Boolean Value	Default value: True
Method	0 (Auto) 1 (ZIP) 2 (CCITT3) 3 (CCITT4) 4 (RLE) 5 (JBIG2)	Defines the compression method to be used to compress monochrome/black & white images. Please note , the compression method JBIG2 cannot be used with the PDF/A output specification. Default value: 0 (Auto)
JBIG2Method	0 (Standard) 1 (CropBorders) 2 (Symbols)	Specifies preprocessing method of images which will be compressed using JBIG2 method. Default value: 0 (Standard)

3.1.2.5 Section Graphics

This section contains three subsections which define how different types of images should be processed before inserting into a PDF document.

- Section Graphics.color
 - Graphics.color.Downsample
 - Graphics.color.Convert
- Section Graphics.Indexed
- Section Graphics.MonoAndDrawings
 - Graphics.MonoAndDrawings.Convert:

3.1.2.5.1 Section Graphics.Color

Determines image downsampling and conversion options for True color images.

- Subsection Downsample defines when and how images should be downsampled.
- Subsection Convert specifies whether to use color reducing for True color images and into which format they should be converted.

3.1.2.5.1.1 Graphics.Color.Downsample

This subsection defines when and how images should be downsampled.

Name	Possible values	Description
Method	0 (None) 1 (Linear) 2 (Bilinear) 3 (Bicubic)	Specifies which method should be used for downsampling. If the value of this property is 0 (None), images of this type will not be downsampled. Default value: 0 (None)
To	50-2399	These two properties define when an image should be downsampled. If the original DPI of the image is equal to or above that of the value specified by the property IfAbove , the image will be downsampled to the DPI specified by the property To . Default value for To : 96 Default value for IfAbove : 128
IfAbove	51-2400	

3.1.2.5.1.2 Graphics.Color.Convert

Specifies whether color images should be converted to grayscale, monochrome or black & white.

Name	Possible values	Description
Method	0 (None) 1 (ToGrayscale) 2 (ToMonochrome) 3 (ToBlackAndWhite)	Specifies how images of this type should be converted. Value 0 (None) means that no images of this type will be converted. Default value: 0 (None)
Dither	Boolean Value	Specifies whether images should be dithered during conversion. This has meaning only for methods ToMonochrome and ToBlackAndWhite . Default value: True

3.1.2.5.2 Section Graphics.Indexed:

Determines image downsampling and conversion options for indexed images with palettes.

- Subsection Downsample defines when and how images should be downsampled.
- Subsection Convert specifies whether to use color reducing for indexed images and into which format they should be converted.

3.1.2.5.2.1 Graphics.Indexed.Downsampling

This subsection defines when and how images should be downsampled.

Name	Possible values	Description
Method	0 (None) 1 (Linear) 2 (Bilinear) 3 (Bicubic)	Specifies which method should be used for downsampling. If the value of this property is 0 (None), images of this type will not be downsampled. Default value: 0 (None)
To	50-2399	These two properties define when an image should be downsampled. If the original DPI of the image is equal to or above that of the value specified by the property IfAbove , the image will be downsampled to the DPI specified by the property To . Default value for To : 96 Default value for IfAbove : 128
IfAbove	51-2400	

3.1.2.5.2.2 Graphics.Indexed.Convert

Specifies whether indexed images should be converted to grayscale, monochrome, or black & white.

Name	Possible values	Description
Method	0 (None) 1 (ToGrayscale) 2 (ToMonochrome) 3 (ToBlackAndWhite)	Specifies how images of this type should be converted. Value 0 (None) means that no images of this type will be converted. Default value: 0 (None)
Dither	Boolean Value	Specifies whether images should be dithered during conversion. This has meaning only for methods ToMonochrome and ToBlackAndWhite . Default value: True

3.1.2.5.3 Section Graphics.MonoAndDrawings

This section determines settings for downsampling and converting monochrome images and black & white drawings.

- Subsection Downsample defines when and how images should be downsampled.
- Subsection Convert specifies whether mono or black & white images should be converted.

3.1.2.5.3.1 Graphics.MonoAndDrawings.Downsampling

This subsection defines when and how images should be downsampled.

Name	Possible values	Description
Method	0 (None) 1 (Linear) 2 (Bilinear) 3 (Bicubic)	Specifies which method should be used for downsampling. If the value of this property is 0 (None), images of this type will not be downsampled. Default value: 0 (None)
To	50-2399	These two properties define when an image should be downsampled. If the original DPI of the image is equal to or above that of the value specified by the property IfAbove , the image will be downsampled to the DPI specified by the property To . Default value for To : 96
IfAbove	51-2400	

		Default value for IfAbove : 128
--	--	--

3.1.2.5.3.2 Graphics.MonoAndDrawings.Convert

Specifies images should be converted as either mono or black & white images.

Name	Possible values	Description
Method	0 (None) 1 (ToGrayscale) 2 (ToMonochrome) 3 (ToBlackAndWhite)	Defines how mono images and line art should be converted. Value 0 (None) means that no conversions will be performed. Default value: 0 (None)
Thresold	1-255	Defines the level behind which color of line arts will be treated as black. For comparison the grayscale level of line art color is used. Default value: 128

3.1.2.6 Section Fonts

This sections contains properties which define embedding and other options for fonts to be used in the PDF document.

Name	Possible values	Description
EmbedAll	Boolean Value	Default value: False
EmbedProtected	Boolean Value	Default value: True
WriteToUnicode	Boolean Value	Default value: False
Subset	Boolean Value	If this option is enabled and if the percentage of characters used from the font specified is less than that specified in the SubsetLevel property, only the actually used characters from the font will be embedded within the PDF document. Otherwise the complete font file will be embedded within the PDF - resulting in a larger PDF file - but this will ensure consistent formatting/viewing is always possible. Default value: True
SubsetLevel	1-100	Default value: 100

Additionally there are two identical subsections **AlwaysEmbed** and **NeverEmbed**, which define lists

of fonts which should always be embedded or which never should be embedded.

Each of these subsections has the following properties:

Name	Possible values	Description
Enable	Boolean Value	Enable using the list defined by the Fonts property. Default value: False
Fonts	String	Contains a comma separated list of fonts. Default value: empty list.

3.1.2.7 Section Security

This section defines security options for the PDF document.

N.B. When the PDF/A format is specified the security options for the document will not be used and will be ignored, it is advisable therefore to actually disable security to enable faster file processing.

Name	Possible values	Description
Enabled	Boolean Value	Specifies if the PDF document should be secured or not. If this property has the value False, no other properties from this section will be used. Default value: False
HasUserPassword	Boolean Value	These properties define if the PDF document will contain a valid User and Owner password. If the PDF document has no user password, this document can be opened without the need to enter a password in any PDF viewing or manipulation tool - even if an Owner password is set. To ensure robust security - set both an Owner and User password ! Default values: False
HasOwnerPassword	Boolean Value	
UserPassword	String	Defines the User password. This password is used for opening the document with user privileges. Default value: empty string.
OwnerPassword	String	Defines the Owner password. This password is required to change permissions/security

		<p>settings in the PDF document. If this password is empty or the same as the User password, changing the permissions/security will be impossible.</p> <p>Default value: empty string.</p>
Type	<p>0 (40RSA) 1 (128RSA) 2 (128AES)</p>	<p>Defines the security method used for encrypting the document. The value of this property specifies which subsection (Perms40 or Perms128) will be used to define users permission for the document.</p> <p>Default value: 0 (40RSA)</p>
Perms40.AllowPrinting	Boolean Values	Default value: False
Perms40.AllowChanging		Default value: False
Perms40.AllowCopying		Default value: False
Perms40.AllowComment		Default value: False
Perms128.ContentAccess	Boolean Value	Default value: False
Perms128.AllowCopying	Boolean Value	Default value: False
Perms128.Changes	0 – 4	<p>Specifies the level of security to be applied regarding changes to the document.</p> <p>0 = No changes will be allowed.</p> <p>1 = Only document assembly is allowed for the user.</p> <p>2 = Only form field fill-in and signing is allowed.</p> <p>3 = Allows comment authoring, form field fill-in and signing of the document.</p> <p>4 = Allows general editing, comments and form field authoring for the user.</p> <p>Default value: 0</p>
Perms128.Printing	<p>0 – not allowed 1 – low resolution 2 – fully allowed</p>	<p>Specifies user's permissions for printing the document.</p> <p>Default value: 0</p>
Perms128.EncryptMeta	Boolean Value	<p>Specifies if metadata In the PDF document should be encrypted or not.</p> <p>N.B. This property is applicable for PDF</p>

		specifications 1.6 and higher. Default value: True
--	--	---

3.1.2.8 Section DigiSig

This section defines Digital Signature properties of the document and the properties for the signature used for signing.

Name	Possible values	Description
Enabled	Boolean value	If this property is False, digital signing is not used and no other properties need to be specified. Default value: False
Store	0 (System) 1 (File)	Defines which stored certificate should be used. If the value of this property is 0 (System), the default user's stored system certificate will be used. If the value of this property is 1 (File), an external PFX file must be specified. Default value: 0 (System)
Hash	Binary	This property specifies the hash value of the certificate to be used for signing the document. When Store value is 0 (System), PDF-XChange will find the corresponding certificate within the system certificate storage. When Store value is 1 (File), this property is not used.
PFXFile	String	When Store value is 1 (File), this specifies the PFX file which contains the digital signature for signing the document. When Store value is 0 (System), this property is not used.
PFXPassword	String	When Store value is 1 (File), this specifies the password for the PFX file. When Store value is 0 (System), this property is not used.
Reason	String	These properties define additional text

Location		properties to be included with the digital signature within the PDF document. Default values: empty string.
Contact		
Graphic	0 (None) 1 (Image) 2 (Name)	Defines how the graphic (left) portion of the signature fields should be displayed. If the value is None , this will not be used, and information about the digital signature will be placed in the entire area of the signature field. If the value is Image , the image file specified by GraphicFileName will be used. With the value Name , the signer's name as in the certificate will be used. Default value: 2 (Name)
GraphicFileName	String	When Graphic is Image , this is the fully qualified path name for the image to be used on the signature. This image can be any one of the supported image formats.
ShowFlags	Integer	A combination of flags which define how the signature field should look. See details below.
Page	Integer	Zero based page number - on which the digital signature should be placed. A value of -1 for this property places the signature on the last page of the document. Default value: -1
Left	Integer	These properties define the position (relative to the top left corner of the page) and the dimensions of the signature field. Values are specified in tenths of a millimetre. Default values: Left: 254 Top: 254 Width: 508 Height: 168
Top		
Width		
Height		

ShowFlags defines the information which should be placed on Signature field appearance. It can be combination of the following values:

Value	Meaning
1	If this value is used, the name of the signer (given from the certificate) will be shown on the signature field.
2	If this value is used, signing date will be placed on signature field.
4	If this value is used, the text specified by Location property will shown.
8	If this value is used, the text specified by Reason property will shown.
16	If this value is used, Distinguished Name field from certificate will be added to the signature field.
32	If this value used, each text item placed on the signature field will be prepended by its corresponding label (for example, Name: Tracker Software Ltd.)

3.1.2.9 Section Overlay

This section contains properties which define how a newly created document can be overlaid using an existing PDF file.

N.B. Using overlays automatically disables the PDF/A feature!

Name	Possible values	Description
Enabled	Boolean value	Enables or disables overlay feature of the PDF-XChange. If this flag is False, all other properties from this section will be ignored. Default value: False
OverlayFile	String	Specifies a file with which the newly created PDF will be overlaid. The value should be a fully-qualified path name for the PDF file. Default value: empty string.
OverlayPassword	String	If the file specified by the OverlayFile property is encrypted with a valid user password, OverlayPassword must contain a valid password for the file. Default value: empty string.
AsForeground	Boolean value	Defines where overlay pages should be located. Background is the default unless specified.

		Default value: False
Repeat	0 (NoRepeat) 1 (RepeatLastPage) 2 (RepeatContinuous)	<p>Specifies the repeat option for overlaying. Meaning of values:</p> <p>0 (NoRepeat): Do not Repeat - begin at the start of the overlay file and apply each corresponding page from this file to the corresponding page number in the newly generated PDF file. Should the newly created file have a greater number of pages than available in the overlay file - no overlay is applied to the remaining pages created in the newly generated PDF document.</p> <p>1 (RepeatLastPage): Apply the PDF overlay file pages to the corresponding pages on the newly generated PDF file - should the newly created file have a greater number of pages - apply the last page of the overlay file to all remaining pages. If the overlay file is a single page - this will be placed on all of the generated files pages by default.</p> <p>2 (RepeatContinuous): Begin at the start of the overlay file and apply each corresponding page from this file to the corresponding page number in the generated PDF file. Should the newly created file have a greater number of pages than available in the overlay file, start at the beginning of the Overlay file and begin the process again, applying the first page in the Overlay PDF file to the next page in the generated output PDF document in the sequence, then page 2 of the overlay file to the next generated PDF document page and so on - until the generated document is completed. If the overlay file is a single page - this will be placed on all of the generated files pages by default.</p> <p>Default value: 0 (NoRepeat)</p>
AlignH	0 (Left) 1 (Center) 2 (Right)	<p>Specifies the horizontal alignment for the overlay pages relative to the new document's pages.</p> <p>Default value: 1 (Center)</p>
AlignV	0 (Top)	Specifies the vertical alignment for the

	1 (Middle) 2 (Bottom)	overlay pages relative to the new document's pages. Default value: 1 (Middle)
Fit	Boolean value	'Fits' the OverlayFile pages to match that of the host page. Default value: False
KeepAspect	Boolean value	The OverlayFile pages' proportions should be retained when using other options. Default value: False

3.1.2.10 Section Links

This property determines whether to insert live links within the PDF document, what kinds of links, and how they are to be displayed (Border property).

Name	Possible values	Description
Enabled	Boolean value	This will detect URL and mail links - please note to be valid, links must be valid and occupy a single line of the page. Embedded links cannot be recognised (e.g. 'Click Here') Default value: False
DetectFiles	Boolean value	Links to files - must be valid and contain a fully qualified path. Default value: False
CheckExistance	Boolean value	Check a links validity as part of the recognition and PDF page inclusion process. Default value: False

The **Border** property of this section defines the links border parameters and the way the link will appear on the page.

Name	Possible values	Description
Type	0 (None) 1 (Underline) 2 (Rectangle)	Specifies the type of links border. Default value: 0 (None)
Style	0 (Solid)	Specifies style for links borders.

	1 (Dashed) 2 (Dotted)	Default value: 0 (Solid)
Color	Integer	Specifies RGB value of color for links border. To specify this value RGB macro defined into Windows API can be used. Default value: 255 (red color)
Width	5 - 50	Defines the width of links border. Value is specified in tenth of millimeters. Default value: 10

3.1.2.11 Section Watermarks

These properties allow the developer to enable or disable adding watermark(s) into the document, and to define a set of these watermarks.

Name	Possible values	Description
Enabled	Boolean value	If this property is True, watermarks defined by Watermarks property will be added to the PDF file. Otherwise watermarks will not be used. Default value: False
Watermarks	String	A comma to separate the list of watermark names should be added to the document. Each of these watermark should be added using AddTextWatermark or AddImageWatermark methods. Default value: empty string.

3.1.2.12 Section HeadersFooters

These properties define headers and footers to be placed on all pages of the PDF document.

Headers and footers contain three parts: left, center and right. Each part specifies the texts color and font that should be inserted into the document.

- Text placed in the left part is left aligned and placed at the left of the page (position for header's left part are defined by OffsetTop and OffsetLeft properties).
- Text of center part is center aligned on the page.
- Text of the right part is right aligned.

Any of the text fields may contain macros, which will be replaced with their values during printing of the PDF.

Name	Possible values	Description
Enabled	Boolean value	Default value: False
FirstPageNumber	Integer	Specifies the first page number for the headers and footers. The value of this property is used in the %[Page] macro. Default value: 1
OffsetLeft	Integer	Values of these properties define the positions on the page where the headers and footers should be placed. All values are in tenths of a millimeter. Default values: 127
OffsetTop		
OffsetRight		
OffsetBottom		

Header and footer are defined with two subsections, **Header** and **Footer** which are identical and contain three identical subsections **Left**, **Center**, and **Right**. Each part has properties:

Name	Possible values	Description
Font	String	The Font used for text which will be added into the document. Default value: Arial.
FontSize	10 – 32768	Specifies font size for the text. Font size is specified in tenth of point (1 point is 1/72 inch). Default value: 100
FontBold	Boolean value	Specifies if the font should be bold. Default value: False
FontItalic	Boolean value	Specifies if italic variant of the font should be used. Default value: False
Color	colorREF	Defines the color of the text. Default value: 0 (black color)
Value	String	Defines the text which should be placed into the part. Text can contains macros. Default value: empty string.

For example, to place a red label with the current page number at the right-top corner of each page,

using Verdana Bold font with 12 pt size, the following properties should be set:

C++

```
printer.Option[L"HeadersFooters.Enabled"] = TRUE;
printer.Option[L"HeadersFooters.Header.Right.Font"] = L"Verdana";
printer.Option[L"HeadersFooters.Header.Right.FontBold"] = TRUE;
printer.Option[L"HeadersFooters.Header.Right.FontSize"] = 120;
printer.Option[L"HeadersFooters.Header.Right.color"] = RGB(255, 0, 0);
printer.Option[L"HeadersFooters.Header.Right.Value"] = L"Page %[Page]";
```

VB

With PDFPrinter

```
.Option("HeadersFooters.Enabled") = TRUE;
.Option("HeadersFooters.Header.Right.Font") = "Verdana";
.Option("HeadersFooters.Header.Right.FontBold") = "Yes";
.Option("HeadersFooters.Header.Right.FontSize") = 120;
.Option("HeadersFooters.Header.Right.color") = 255;
.Option("HeadersFooters.Header.Right.Value") = "Page %[Page]";
```

End With

3.1.2.13 Section DocInfo

The properties of this section allow PDF document information fields to be defined and allow adding of XMP Metadata within the PDF document.

Name	Possible values	Description
Enabled	Boolean value	If this property is False, no information fields will be specified within the PDF document except the Producer field, which is always specified and contains the PDF-XChange version information for the creation of this PDF file. Default value: False
AskForDocInfo	Boolean value	If this option is True, a popup window where Document information fields can be entered will be shown after the document is spooled. Default value: False
AddXMP	Boolean value	If this option is True, <i>XMP metadata</i> will be added within the PDF document. This metadata will contain values of information fields, date of document creation etc. Please note , when PDF/A mode used, XMP metadata will be added into the document irregardless of the value of this property and Enabled property.

		Default value: False
Title	String	Specifies the Title information field of the document. Default value: empty string
Subject	String	Specifies the Subject information field of the document. Default value: empty string
Author	String	Specifies the Author information field of the document. Default value: empty string
Keywords	String	Specifies the Keywords information field of the document. Keywords should be comma separated. Default value: empty string
Creator	String	Specifies the Creator information field of the document. Default value: empty string

3.1.2.14 Section Save

Using properties of this section it is possible to define the folder where PDF document should be stored, the file name of the document and other document saving options.

Name	Possible values	Description
SaveType	0 (pdfSaver) 1 (AppendToLast) 2 (Save) 3 (JustEmail)	Specifies the action which should be performed when document is spooled. <ul style="list-style-type: none"> • pdfSaver specifies that the pdfSaver application (the document creation application of PDF-XChange) will display the Job Management window where the action which should be performed can be specified. • AppendToLast appends the newly printed document to the last job in the jobs list. pdfSaver may then be used to create a PDF file from this job. • Save (most used) generates and saves the PDF file from the printed document using other properties defined into this section.

		<ul style="list-style-type: none"> • JustEmail creates the PDF file and emails it using properties defined in the EMail section. <p>Default value: 2 (Save)</p>
ShowSaveDialog	Boolean value	<p>If this property is True, pdfSaver will display the Save As dialog where the path and file name of the PDF file must be specified.</p> <p>Canceling the Save As dialog will cancel the PDF file creation.</p> <p>Default value: True</p>
Path	String	<p>Defines the path where newly created PDF file should be saved. If this property is an empty string, the user's My Documents folder will be used for document storing.</p> <p>In some cases this property is not used. See the description of the File property.</p> <p>Default value: empty string</p>
File	String	<p>Defines file name of the newly created PDF file. This name can contain macros and a path. If this property contains a path, this path will be used instead of path specified in Path property.</p> <p>Default value: %[DocName]</p>
StripPath	Boolean value	<p>Defines how value of macro %[DocName] will be calculated.</p> <p>If this property is True and the document name passed from the printing application contains the path, this path will then be stripped during assigning %[DocName] property.</p> <p>Default value: False</p>
WhenExists	0 (ShowWarning) 1 (Overwrite) 2 (AutoNumber) 3 (Append) 4 (InsertBefore)	<p>Specifies the action which will be performed during saving if the file with the name specified by Path and File properties already exists.</p> <p>When value is ShowWarning, pdfSaver will show a warning dialog where the user can choose what to do.</p> <p>Value Overwrite allows pdfSaver to overwrite existing file without any warning</p>

		<p>dialogs.</p> <p>If the value is AutoNumber, pdfSaver will change the name of the destination file by appending with a numeric value. The format of this number is specified by AutoNumber.Start and AutoNumber.NumDigits properties.</p> <p>Values Append and InsertBefore allow pdfSaver to merge existing and newly created PDF documents by placing the newly created PDF at the end (for value Append) or at the beginning (InsertBefore) of the existing file.</p> <p>Please note, these options are not available when PDF/A feature is used.</p> <p>Default value: 0 (ShowWarning)</p>
AutoNumber.Start	0 – 999999	<p>Define the start number when auto-numbering used.</p> <p>Default value: 1</p>
AutoNumber.NumDigits	1 - 6	<p>Define the format of the number when auto-numbering used. The number of digits defines the minimum number of digits used for the number (if number is shorted it will be prepended by '0' characters).</p> <p>Default value: 3</p>
RunApp	Boolean value	<p>Specifies if the newly created PDF document should be opened after creation.</p> <p>Default value: True</p>
RunCustom	Boolean value	<p>Defines which application should be used to open created PDF document. If the value of this property is True, application specified by AppName will be used; otherwise the default PDF application will be used.</p> <p>This property used only when RunApp is True.</p> <p>Default value: False</p>
AppName	String	<p>This property is used only when both RunApp and RunCustom properties have the value True.</p> <p>It names the application which should be</p>

		used for opening the created PDF file. The value of this property must be the fully-qualified path name of the application. Default value: empty string
AppParams	String	This property specifies the parameters that will be passed to the AppName application when opening the PDF file. String %f represents the full path name of the created PDF file. Default value: %f

3.1.2.15 Section EMail

Email properties are used when the created PDF file should be emailed after creation and define the parameters for emailing. The created PDF file will be sent as an attachment of this email.

Name	Possible values	Description
Enabled	Boolean value	Specifies if the PDF document should be emailed after creation. If the value of this property is False, none of the other properties in this section are used. Default value: False
Method	0 (LaunchClient) 1 (SendViaClient) 2 (SendViaSMTP)	Specifies how the email should be sent. Value LaunchClient launches the default email client with parameters filled in from the other properties, and the user will send the email manually after making any corrections or emendations as required. Value SendViaClient launches the default email client and sending the email automatically without intervention from the user. Value SendViaSMTP will use the SMTP server specified by the Server property to send the email. Default value: 0 (LaunchClient)
From	String	Specifies the email account to use for the From email property.
To	String	Specifies various recipient(s) of the email. If more than one recipient is specified, they

Cc		should be separated by the semicolon (;) character.
Bcc		Each recipient can be specified as recipient name (for example, <i>John Smith</i>) and the email address will be resolved from the email client's address book; email address (for example, <i>john@server.com</i>); or as a combination of name and address: <i>John Smith <john@server.com></i> . Default values: empty string
Subject	String	Defines the subject string for the email. Default value: empty string
Content	String	Defines content of the email. Default value: empty string
ZIP	Boolean value	Specifies how the PDF file should be added as an attachment to the email. When True, the PDF file should be sent as a ZIP archive. When False, the PDF file should be sent as the original PDF file. Default value: True - send as ZIP archive
Server	String	Specifies the name of the SMTP server for sending email. This property is used only when the Method property has the value SendViaSMTP . The SMTP server must be defined using AddSMTPServer method. Default value: empty string

3.1.2.16 Section Optimization

This section controls Optimizations of the generated PDF file.

Name	Possible values	Description
Blending	One of following integer value: 0 – Normal 1 – Multiply	Specifies blending mode used for blending line art during printing from some application (for example from AutoCAD when lines overlaing is enabled in it). Default value: 4

	<p>2 – Screen</p> <p>3 – Overlay</p> <p>4 – Darken</p> <p>5 – Lighten</p> <p>6 – colorDodge</p> <p>7 – colorBurn</p> <p>8 – HardLight</p> <p>9 – SoftLight</p> <p>10 – Difference</p> <p>11 – Exclusion</p>	
DeTessellation	<p>0 (None)</p> <p>1 (Simple)</p>	<p>Specifies if detessellation should be used for a sequence of triangles. Detessellation is useful when printing from AutoCAD and allow to produce smaller and optimal PDF files.</p> <p>Default value: 1 (Simple)</p>
UseTransparency	Boolean value	<p>Enables transparency drawing into the PDF file.</p> <p>PDF-XChange can detect some drawing sequences and can then replace them with transparency drawing.</p> <p>Please note: when PDF/A mode is used, this property is forced to False value.</p> <p>Default value: True</p>
UseMaskedImages	Boolean value	<p>Using masked images produces smaller PDF files for some drawing sequences.</p> <p>Default value: True</p>
GrayPaletteAsRGB	Boolean value	<p>If this option is enabled, the palettes of grayscaled images are stored as an RGB palette where each element of this palette has the same value for each of its RGB components.</p> <p>Default value: False</p>
ImagesOptimization	Boolean value	<p>Enables the optimization of image sequences.</p> <p>Such optimization requires many manipulations with images but causes the resulting PDF to be much smaller and better</p>

		for viewing. Default value: True
EnableSwapping	Boolean value	Enables swapping images to the disk during image optimization. When swapping is turned off optimization time is smaller as a rule, but requires more memory. Default value: True
SwapToDiskLimit	1-8	Defines the image size (in megabytes) in memory starting from which this image will be swapped to disk (if option EnableSwapping has True value). Default value: 1

3.1.2.17 Section Bookmarks

This section activates automatic bookmark tree creation for the PDF document and defines the parameters for this process.

Name	Possible values	Description
Enabled	Boolean value	Enables automatic creation of a bookmark tree within the document. If the value of this property is False, no bookmarks tree will be created, and the rest of the properties of this section are ignored. Default value: False
AllowMultiLine	Boolean value	Activates detection of multiline bookmark items. Default value: True
DocTitleAsRoot	Boolean value	When True, the Root element of the bookmarks tree will be the document title (see DocInfo.Title property) and all other top level bookmarks will be children of this element. Default value: False
MatchLevel	Boolean value	When True, a detected non-top level item will only be added to the tree if a previously detected item of higher level already exists. When False, a detected item will always be

		<p>added to the tree, and if there was no parent detected at a higher level, dummy parents will be added.</p> <p>Default value: False</p>
DisplayMode	<p>0 (AllOpened)</p> <p>1 (AllClosed)</p> <p>2 (ByItem)</p> <p>3 (OpenToLevel)</p>	<p>Defines which items of the bookmarks tree should be initially opened into the PDF viewer application.</p> <p>Value AllOpened means that all items will be opened.</p> <p>Value AllClosed means that all items will be closed.</p> <p>Value ByItem means the state of the item defined by its template definition will be used.</p> <p>Value OpenToLevel, means that all items up to the level specified by UpToLevel property will be opened. All other items will be closed by default.</p> <p>Default value: 2 (ByItem)</p>
UpToLevel	Integer	<p>Defines the level into bookmarks tree up to which items will be initially opened into the PDF viewer application.</p> <p>Only active if DisplayMode has the value 3 (OpenToLevel).</p> <p>Default value: 1</p>
Template	String	<p>Specifies the name of Template used for detection and creation bookmarks tree within the PDF document.</p> <p>This property is not used with the ActiveX control and should be an empty string which means current template. The current template can be created by using method <code>AddBookmarkItem</code>.</p> <p>Default value: empty string</p>

3.1.2.18 Section Saver

This section controls the **ShowProgress** property of the **pdfSaver** application.

Name	Possible values	Description
ShowProgress	Boolean value	If this property is True, the progress dialog

		<p>will be shown during the printing and PDF creation process.</p> <p>Value False disables this progress window.</p> <p>Default value: True</p>
--	--	---

N.B. Other properties may be added later.

3.1.2.19 Macros

PDF-XChange supports macros for file naming and for headers/footers text. All macros has syntax:

```
%[Macro{ : extra}]
```

Where **Macro** is a name of the macro value, and **extra** defines extra specifier for the macro.

Items between **{}** are optional. Not all macros use an extra specifier, and will ignore all text after **:** up to closing bracket **]**.

The following macros are support into PDF-XChange:

Name	Extra	Description
AppName	not defined	Specifies the executable module of printing application. It contains only a file name of executable, and doesn't contain the path of the this module. For example, when printing is from Microsoft Word, this macro will have the value Winword.exe .
DocName	not defined	Specifies the document name as it was passed to the PDF-XChange driver by printing application (via DOCINFO structure at printing start). If property Save.StripPath is True and document name, passed by application, contains a path, this path will be stripped before assigning to the macro.
DocPath	not defined	Specifies the path part (if it is present) of the document name passed by the printing application.
Date	Format string. See below for syntax of format string.	Has a value of date of document printing. Value can be formatted using format specified in extra part of the macro. When no extra is specified, default regional settings are used for formatting.
Year	not defined	Year number. Format of this number (two digits or four) depends from regional settings.
Month	not defined	Month number. Format depends from

		regional settings.
Day	not defined	Day of month (as digits). Format depends from regional settings.
Time	Format string. See below for syntax of format string.	Printing time as string. Value can be formatted using format specified in extra part of the macro. When no extra is specified, default regional settings are used for formatting.
Hour	not defined	Hour of printing. Format depends from regional settings.
Minute	not defined	Minute of printing. Format depends from regional settings.
Second	not defined	Seconds of printing. Format depends from regional settings.
Computer	not defined	The name of user's computer.
User	not defined	The name (login name) of user who makes printing.
Page	not defined	Current page number. Can be used only for headers and footers. Starting page number is defined by HeadersFooters.FirstPageNumber property.
Pages	not defined	Total number of pages into the document.
Env	Environment variable name	Has a value of environment variable specified by extra string. If there are no such environment variable, or extra string isn't specified, value of this macro will be an empty string. For example, %[Env:UserName] will have the login name of the user who prints.

Format of picture string used to form the date string. The following elements can be used to construct a format picture string. If there are spaces to separate the elements in the format string, these spaces will appear in the same location in the result string. The letters must be in uppercase or lowercase as shown in the table, for example, "MM" not "mm". Characters in the format string that are enclosed in single quotation marks appear in the same location and unchanged in the output string.

Picture	Meaning
---------	---------

d	Day of month as digits with no leading zero for single-digit days.
dd	Day of month as digits with leading zero for single-digit days.
ddd	Day of week as a three-letter abbreviation. Depends of regional settings.
dddd	Day of week as its full name. Depends of regional settings.
M	Month as digits with no leading zero for single-digit months.
MM	Month as digits with leading zero for single-digit months.
MMM	Month as three-letter abbreviation. Depends of regional settings.
MMMM	Month as its full name. Depends of regional settings.
y	Year as last two digits, but with no leading zero for years less than 10.
yy	Year as last two digits, with leading zero for years less than 10.
yyyy	Year represented by full four or five digits, depending of the calendar used. Thai Buddhist and Korean calendars both have five digit year. The "yyyy" pattern will show five digits for these two calendars, and four digits for all other supported calendars.
yyyyy	Same as "yyyy".
gg	Period/era string. This element is ignored if the date does not have an associated era or period string. Depends of regional settings.

Format of picture string used to form the time string. The following elements can be used to construct a format picture string.

Picture	Meaning
h	Hours with no leading zero for single-digit hours; 12-hour clock.
hh	Hours with leading zero for single-digit hours; 12-hour clock.
H	Hours with no leading zero for single-digit hours; 24-hour clock.
HH	Hours with leading zero for single-digit hours; 24-hour clock.
m	Minutes with no leading zero for single-digit minutes.
mm	Minutes with leading zero for single-digit minutes.
s	Seconds with no leading zero for single-digit seconds.
ss	Seconds with leading zero for single-digit seconds.

t	One character time marker string, such as A or P. Depends of regional settings.
tt	Multicharacter time marker string, such as AM or PM. Depends of regional settings.

3.2 Methods

Methods are used to perform ActiveX control actions that simple property assignments cannot accomplish.

- Method **SetRegInfo** is used to set registration information specific for developer for the printer.
- Method **ResetDefaults** resets all properties to their default values.
- Method **ApplyOptions** is used to apply all properties changed with property **Option**.
- Method **AddBookmarkItem** adds a new item into bookmarks template tree.
- Method **AddTextWatermark** is used to define a new text watermark which may be added to the document.
- Method **AddImageWatermark** is used to define a new image watermark which may be added to the document.
- Method **StorePrinterOptions** is used to store all printer options into any external storage which supports an **IStream** interface.
- Method **ReStorePrinterOptions** restores all printer options previously stored by method **StorePrinterOptions**.
- Method **SetAsDefaultPrinter** sets the printer associated with the **IPXCPrinter** object as the system default printer.
- Method **RestoreDefaultPrinter** restores the system default printer which was stored during a call to the **SetAsDefaultPrinter** method.
- Method **AddSMTPServer** adds a definition of an SMTP server, which can be used for sending PDF files via email.

3.2.1 Method SetRegInfo

This method is used to set registration information specific for developer for the printer. This method may be used when registration information was not passed into the **Printer** property of **IPXCControlEx** object during printer creation.

Declaration:

C#

```
void SetRegInfo(string pRegKey, string pDevCode);
```

C++

```
HRESULT SetRegInfo([in] BSTR pRegKey, [in] pDevCode);
```

VB

```
Sub SetRegInfo(pRegKey As String, pDevCode As String)
```

3.2.2 Method ResetDefaults

This method resets all properties to their default values, as defined in the Properties sections above.

Declaration:

C#

```
void ResetDefaults();
```

C++

```
HRESULT ResetDefaults();
```

VB

```
Sub ResetDefaults()
```

3.2.3 Method ApplyOptions

This method is used to apply all properties changed with property **Option**.

This method should be called before printing onto the printer associated with the **IPXCPrinter** object.

Declaration:

C#

```
void ApplyOptions(int dwFlags);
```

C++

```
HRESULT ApplyOptions([in] DWORD dwFlags);
```

VB

```
Sub ApplyOptions(dwFlags As Long)
```

dwFlags

Reserved for further usage and should be set to 0.

3.2.4 Method AddBookmarkItem

Method **AddBookmarkItem** adds a new item into the bookmarks template tree. Returned value from this method can be used for subsequent calls of this method.

Declaration:

C#

```
int AddBookmakrkItem(int nParent,  
    int dwDetectFlags,  
    string sDet_FontName,  
    int nDet_FontStyle,  
    int nDet_Size,  
    int nDet_SizeDelta,  
    int nDet_Textcolor,  
    int dwDisplayFlags,  
    int dwDisplaycolor);
```

C++

```
HRESULT AddBookmakrkItem([in] long nParent,
    [in] long dwDetectFlags,
    [in] BSTR sDet_FontName,
    [in] long nDet_FontStyle,
    [in] long nDet_Size,
    [in] long nDet_SizeDelta,
    [in] long nDet_Textcolor,
    [in] long dwDisplayFlags,
    [in] long dwDisplaycolor,
    [out,retval] long* pID);
```

VB

```
Function AddBookmakrkItem(nParent As Long,
    dwDetectFlags As Long,
    sDet_FontName As String,
    nDet_FontStyle As Long,
    nDet_Size As Long,
    nDet_SizeDelta As Long,
    nDet_Textcolor As Long,
    dwDisplayFlags As Long,
    dwDisplaycolor As Long) As Long
```

nParent

Specifies the parent ID of the item to be created. This could be a value returned from a previous call to this function, or -1, if the item should be a top level item.

dwDetectFlags

A bit combination integer value which specifies which properties of the body text are to be used to match it as a bookmark item. A combination of the following values can be used.

Bit	Value	Meaning
0	1	If this bit is set, the font name specified is used by the <i>sDet_FontName</i> parameter to determine all matching text/strings. <i>sDet_FontName</i> may not be NULL or an empty string when this flag is set.
1	2	If this bit is set, the font style specified is used by the <i>nDet_FontStyle</i> parameter to determine all matching text/strings.
2	4	If this bit is set, the font size specified is used by the <i>nDet_Size</i> and <i>nDet_SizeDelta</i> parameters to determine all matching text/strings. Text with the font size $nDet_Size \pm nDet_SizeDelta$ matches and is therefore added as a bookmark item.
3	8	If this bit is set, the text color specified by <i>nDet_Textcolor</i> is also used as part of the process in detecting potential matches.

The value of this parameter cannot be **zero (0)**; at least one matching parameter **must** be used.

sDet_FontName

Specifies the font name for matching.

nDet_FontStyle

Specifies the font style for matching as an integer value which may have any one of the following values:

Value	Meaning
0	Normal Font
1	Bold Font
2	Italic Font
3	Bold Italic Font

nDet_Size

Specifies the font size to be used for matching. The font size is specified in tenths of a point (one point = 1/72 inches). This unit is 1/720 of an inch.

nDet_SizeDelta

Specifies the font size delta to be used for matching. The delta value is specified in tenths of a point (one point = 1/72 inches). This unit is 1/720 of an inch.

nDet_Textcolor

Specifies text color to be used for matching.

dwDisplayFlags

A bit combination integer value which defines how the bookmark item will be shown in the bookmark tree of the PDF viewing application. May be a combination of the following flags:

Bit	Value	Meaning
0	1	When this bit is set, the bookmark item will be drawn using a bold font.
1	2	When this bit is set, the bookmark item will be drawn using an italic font.
2	4	When this bit is set, the bookmark item will be initially opened in the bookmark tree window of the viewing application. This flag is used only when the property <code>Bookmarks.DisplayMode</code> has a value <code>ByItem</code> .

dwDisplaycolor

Specifies the color bookmark items will be displayed in, if matching this template, in the viewer's bookmark tree window. The RGB Macro defined in the Windows API may be used to specify the color value.

Return Value

This function returns an internal ID for the newly created bookmark item. This value may be used as a `nParent` parameter for later calls to the `AddBookmarkItem` method.

Example.

The following example creates simple bookmarks detection tree, which makes text drawn using Times New Roman, 12 pt, bold font as top-level items. And items drawn by the same font, but not bold - as second level items. Also items drawn using Arial, bold, 14 pt, will be added as top level item and will be displayed using red color.

VB

```
Dim parent as Long
parent = PXCPrinter.AddBookmakrkItem(-1, 7, "Times New Roman", 1, 120, 5, 0, 0, 0)
PXCPrinter.AddBookmakrkItem(parent, 7, "Times New Roman", 0, 120, 5, 0, 0, 0)
PXCPrinter.AddBookmakrkItem(-1, 7, "Arial", 1, 140, 5, 0, 0, 255)
```

3.2.5 Method AddTextWatermark

This method is used to define a new text watermark to be added to the document. This methods parameters correspond to the parameters available in the **PDF-XChange Driver V 4.x GUI** during the adding/editing of a Text Watermark (topic Watermarks; Add... button).

Declaration:**C#**

```
void AddTextWatermark(
    string sName,
    string sText,
    string sFontName,
    int dwFontWeight,
    int bItalic,
    int bOutline,
    int nFontSize,
    int nLineWidth,
    int nTextcolor,
    int dwAlign,
    int xOffset,
    int yOffset,
    int nAngle,
    int dwOpacity,
    int dwFlags,
    int dwPlaceType,
    string sRange);
```

C++

```
HRESULT AddTextWatermark(
    [in] BSTR sName,
    [in] BSTR sText,
    [in] BSTR sFontName,
    [in] long dwFontWeight,
    [in] long bItalic,
    [in] long bOutline,
    [in] long nFontSize,
    [in] long nLineWidth,
    [in] long nTextcolor,
```

```
[in] long dwAlign,  
[in] long xOffset,  
[in] long yOffset,  
[in] long nAngle,  
[in] long dwOpacity,  
[in] long dwFlags,  
[in] long dwPlaceType,  
[in] BSTR sRange);
```

VB

```
Sub AddTextWatermark(  
    sName As String,  
    sText As String,  
    sFontName As String,  
    dwFontWeight As Long,  
    bItalic As Long,  
    bOutline As Long,  
    nFontSize As Long,  
    nLineWidth As Long,  
    nTextColor As Long,  
    dwAlign As Long,  
    xOffset As Long,  
    yOffset As Long,  
    nAngle As Long,  
    dwOpacity As Long,  
    dwFlags As Long,  
    dwPlaceType As Long,  
    sRange As String)
```

sName

Specifies the name of the newly created watermark item. This name is used to identify this watermark in the Watermarks.Watermarks property string. If there is already a watermark with the same name - it will be overwritten!

sText

Specifies the watermark text.

sFontName

Specifies the font (by name) to use when drawing the watermark. **N.B. Required !**

dwFontWeight

Specifies the weight of the font. Possible values are: 400 for Normal and 700 for Bold.

bItalic

When True, specifies the italic variant of the font identified by the *sFontName* parameter.

bOutline

When True, the text will be drawn as an outline; when False, as solid filled text.

nFontSize

Specifies the font size of the text. Font size is specified in tenths of a point (1 pt = 1/72 inch). When this parameter is 0 (zero), the font size will be adjusted to fit the entire page.

nLineWidth

Specifies the line width for outlining the contour of the text. Line width is specified in tenths of points. Used only when *bOutline* is TRUE.

nTextcolor

Specifies the text color of the watermark.

dwAlign

Specifies the text alignment on the page. When *nFontSize* is 0 (zero) this parameter is ignored and the watermark will be centered vertically and horizontally (value 9).

dwAlign may be a combination any of the following values:

Bit	Value	Meaning
0	1	Horizontal Center Align.
1	2	Right Align: Cannot be set when bit 0 (Horizontal center align is set) When both bits 0 and 1 is not set, Left align used.
4	8	Vertical Center align.
5	16	Bottom Align. Cannot bit set when bit 4 is set. When both bits 4 and 5 is not set, Top align used.

xOffset, yOffset

Specify Horizontal and Vertical offsets for the watermark relative to the top-left corner of the page. Values are specified in tenths of millimeters.

nAngle

Specifies the Rotation angle for the watermark. Possible values are from -90 to 90 degrees. Positive values mean counterclockwise rotation.

dwOpacity

Specifies the Opacity level of the watermark. Value must be in the range of 0 (fully transparent) to 100 (fully opaque).

dwFlags

Bit combination integer value which defines how the watermark should be placed and when it should be visible, may be a combination of the following flags:

Bit	Value	Meaning
1	2	If this bit is set the watermark will be placed as a background on the page. Otherwise it will be placed in the foreground - over the content of the existing page.
2	4	Watermark should be visible during viewing the PDF document in the PDF viewing application.

3	8	Watermark should be visible when the PDF document is exported (for example as an image format) from the viewing application.
4	16	Watermark should be printed when the PDF document is printed from the PDF viewing application.

dwPlaceType

Defines on which pages the watermark should be placed. May be any of the following values:

Value	Meaning
0	Watermark will be placed on all pages of the document
1	Watermark will be placed only on the first page of the document.
2	Watermark will be placed only on the last page of the document.
3	Watermark will be placed only on even pages of the document.
4	Watermark will be placed only on odd pages of the document.
5	Watermark will be placed on pages specified by the sRange parameter.

sRange

A String defining the page range(s) onto which the watermark should be placed. Each range can be: page number, or range of pages in the form [p1]-[p2], which means from p1 to p2 inclusive. One of p1 or p2 may be omitted. In this case all pages prior to p2 or all pages after p1 will be included into the range. Ranges should be separated by a comma(,) or semicolon (;).

Example of range string: "-3, 5, 7-11, 15-". This decodes as "pages 1, 2, 3, 5, 7, 8, 9, 10, 11, 15 to the end of the document."

3.2.6 Method AddImageWatermark

This method is used to define a new image watermark to be added to the document. The parameters of this method correspond to the parameters available in the **PDF-XChange Driver V 4.x** UI during the adding/editing of a text watermark (page Watermarks; Add... button).

Declaration:**C#**

```
void AddImageWatermark(
    string sName,
    string sImageFileName,
    int dwTranscolor,
    int dwWidth,
    int dwHeight,
    int dwAlign,
    int xOffset,
    int yOffset,
    int nAngle,
```

```
int dwOpacity,
int dwFlags,
int dwPlaceType,
string sRange);
```

C++

```
HRESULT AddImageWatermark(
    [in] BSTR sName,
    [in] BSTR sImageFileName,
    [in] long dwTranscolor,
    [in] long dwWidth,
    [in] long dwHeight,
    [in] long dwAlign,
    [in] long xOffset,
    [in] long yOffset,
    [in] long nAngle,
    [in] long dwOpacity,
    [in] long dwFlags,
    [in] long dwPlaceType,
    [in] BSTR sRange);
```

VB

```
Sub AddImageWatermark(
    sName As String,
    sImageFileName As String,
    dwTranscolor As Long,
    dwWidth As Long,
    dwHeight As Long,
    dwAlign As Long,
    xOffset As Long,
    yOffset As Long,
    nAngle As Long,
    dwOpacity As Long,
    dwFlags As Long,
    dwPlaceType As Long,
    sRange As String)
```

sName

Specifies the name of the newly created watermark item. This name may be used in the Watermarks property. If there is already an existing watermark with the same name, it will be *over written*.

sImageFileName

Specifies the fully qualified path name to the image file to be used as a watermark. The image may be in any of the following supported image formats: BMP, PNG, JNG, JPEG, JPEG 2000, TIFF, JBIG, JBIG2, GIF, PCX, DCX, EMF, WMF. If the image is a multipage image, only the first page will be used as a watermark.

dwTranscolor

Specifies the transparent color for the image. If this parameter is anything other than **-1**, each pixel in the *sImageFileName* image which has a color specified by the *dwTranscolor* parameter will be transparent in the PDF file generated.

dwWidth, dwHeight

Specifies the dimensions of the watermark. The image specified by the *sImageFileName*

parameter will be resized to 'fit' within the specified area and will retain its aspect ratio.

All other parameters of this method have the same meaning as the corresponding parameters of `AddTextWatermark` method.

Return value

The method returns `S_OK` if successful, or an error code if it fails.

3.2.7 Method `StorePrinterOptions`

This method is used to store all printer options into any external storage method supporting an `IStream` interface.

Declaration:**C#**

```
void StorePrinterOptions(IStream pStream);
```

C++

```
HRESULT StorePrinterOptions(IStream* pStream);
```

VB

```
Sub StorePrinterOptions(pStream As IStream)
```

pStream

Pointer to the **IStream** object into which all printer properties will be serialized.

Return value

The method returns `S_OK` if successful; otherwise an error code is returned.

3.2.8 Method `ReStorePrinterOptions`

This method restores all original printer options previously saved using the method `StorePrinterOptions` to save to an external storage method supporting an `IStream` interface.

Declaration:**C#**

```
void ReStorePrinterOptions(IStream pStream);
```

C++

```
HRESULT ReStorePrinterOptions(IStream* pStream);
```

VB

```
Sub ReStorePrinterOptions(pStream As IStream)
```

pStream

Pointer to the **IStream** object where the printer properties are to be stored.

Return value

The method returns `S_OK` if successful; otherwise an error code is returned.

3.2.9 Method SetAsDefaultPrinter

This method sets the printer associated with the **IPXCPrinter** object as the system default printer. The original system default printer can be restored later using the method `RestoreDefaultPrinter`.

The original system default printer will be restored automatically when releasing the **IPXCPrinter** object.

Declaration:

C#

```
void SetAsDefaultPrinter();
```

C++

```
HRESULT SetAsDefaultPrinter();
```

VB

```
Sub SetAsDefaultPrinter()
```

Return value

The method returns `S_OK` if successful; otherwise an error code is returned.

3.2.10 Method RestoreDefaultPrinter

This method restores the original system default printer modified when calling `SetAsDefaultPrinter` method. If the method **SetAsDefaultPrinter** was not previously called this method is ignored.

Declaration:

C#

```
void RestoreDefaultPrinter();
```

C++

```
HRESULT RestoreDefaultPrinter();
```

VB

```
Sub RestoreDefaultPrinter()
```

Return value

The method returns `S_OK` if successful; otherwise an error code is returned.

3.2.11 Method AddSMTPServer

This method adds the parameters of a defined SMTP server, which may then be used for sending PDF files via email.

For more details see description of **EMail** section.

Declaration:

C#

```
void AddSMTPServer(  
    string sName,  
    int nPort,  
    int bNeedAuth,
```

```
string sUser,  
string sPassword,  
long bSecureConnection);
```

C++

```
HRESULT AddSMTPServer(  
    [in] BSTR sName,  
    [in] long nPort,  
    [in] long bNeedAuth,  
    [in] BSTR sUser,  
    [in] BSTR sPassword,  
    [in] long bSecureConnection);
```

VB

```
Sub AddSMTPServer(  
    sName As String,  
    nPort As Long,  
    bNeedAuth As Long,  
    sUser As String,  
    sPassword As String,  
    bSecureConnection As Long)
```

sName

Specifies the name or IP address of the SMTP server.

nPort

Specifies the port number to be used to communicate with the SMTP server. The usual SMTP default is port 25.

bNeedAuth

Set to True if the specified server requires authorization for sending email.

sUser

User name used for authorization on the server. Not used when *bNeedAuth* is False.

sPassword

Password used for access authorization to the server. Not used when *bNeedAuth* is False.

bSecureConnection

Set to True if the server requires a secure connection or False otherwise.

Return value

Method returns S_OK if success, and other values on fail.

3.3 Events

The **IPXCPrinter** interface provides events which can be used by a Developer's applications to determine the stage of the PDF creation process. Some events are fired during document spooling and these events are asynchronous to prevent dead-locking. Events which are fired during the generation of PDF documents are synchronous.

An **asynchronous event** means that PDF-XChange fires the event and will not wait while application receives this event and returns from it. It also means that the application will receive these events immediately when they are fired.

When a **synchronous event** is fired, PDF-XChange will wait until the application has finished this event.

All events have a **JobID** parameter which specifies an internal Job identifier of the printed document. This ID can be used when multiple documents are printing to determine from which document the event was fired.

The following events are supported in **PDF-XChange Driver v4**.

- Asynchronous Events:
 - Event OnStartDoc is fired when a new document starts to print.
 - Event OnStartPage is fired for each page when it starts to print.
 - Event OnEndPage is fired for each page after it prints.
 - Event OnEndDoc is fired when the complete document is spooled by the GDI.
- Synchronous Events:
 - Event OnDocSpooled is fired immediately when PDF-XChange starts processing the document that was printed by the GDI.
 - Event OnFileSaved is fired when the PDF file is saved.
 - Event OnFileSent is fired when emailing of the PDF file was enabled and after the email was sent. If emailing is not enabled this event will not fire.
 - Event OnError is fired when an error occurs during the processing of the PDF file.

3.3.1 Asynchronous Events

The following Asynchronous events are supported in **PDF-XChange Driver v4**.

- Event OnStartDoc is fired when a new document starts to print.
- Event OnStartPage is fired for each page when it starts to print.
- Event OnEndPage is fired for each page after it prints.
- Event OnEndDoc is fired when the complete document is spooled by the GDI.

3.3.1.1 Event OnStartDoc

This event is fired when a new document starts to print.

Declaration:

C#

```
event OnStartDoc(int JobID, string lpszDocName, string lpszAppName);
```

C++

```
HRESULT OnStartDoc(LONG JobID, BSTR lpszDocName, BSTR lpszAppName);
```

VB

```
Event OnStartDoc(JobID As Long, lpszDocName As String, lpszAppName As String)
```

JobID

Internal Job identifier.

lpszDocName

Specifies the document name as it was passed from the printing application.

lpszAppName

Specifies the module name of the printing application.

Return value

PDF-XChange ignores returned values from this event.

3.3.1.2 Event OnStartPage

This event is fired for each page when it starts to print.

Declaration:

C#

```
event OnStartPage(int JobID, int nPageNumber);
```

C++

```
HRESULT OnStartPage(LONG JobID, LONG nPageNumber);
```

VB

```
Event OnStartPage(JobID As Long, nPageNumber As Long)
```

JobID

Internal Job identifier.

nPageNumber

Zero based page number when spooling.

Return value

PDF-XChange ignores returned values from this event.

3.3.1.3 Event OnEndPage

This event is fired for each page after printing completes.

Declaration:

C#

```
event OnEndPage(int JobID, int nPageNumber);
```

C++

```
HRESULT OnEndPage(LONG JobID, LONG nPageNumber);
```

VB

```
Event OnEndPage(JobID As Long, nPageNumber As Long)
```

JobID

Internal Job identifier.

nPageNumber

Zero based page number spooled.

Return value

PDF-XChange ignores returned values from this event.

3.3.1.4 Event OnEndDoc

This event is fired when the complete document has been spooled by the GDI.

Declaration:**C#**

```
event OnEndDoc(int JobID, int bOK);
```

C++

```
HRESULT OnEndDoc(LONG JobID, LONG bOK);
```

VB

```
Event OnEndDoc(JobID As Long, bOK As Long)
```

JobID

Internal Job identifier.

bOK

When this parameter is True, the document has fully spooled and PDF-XChange will continue to process this document to produce the PDF file. If there was an error during printing (e.g. printing cancelled) there will be no further document processing.

Return value

PDF-XChange ignores returned values from this event.

3.3.2 Synchronous Events

The following Synchronous events are supported by the PDF-XChange V4 Driver.

- Event OnDocSpooled is fired immediately PDF-XChange starts processing the document printed by the GDI.
- Event OnFileSaved is fired when the PDF file is saved.
- Event OnFileSent is fired when PDF emailing is enabled and after the email has been sent. If emailing is not enabled this event will not fire.
- Event OnError is fired if an error occurs during the processing of the PDF file.

3.3.2.1 Event OnDocSpooled

This event is fired immediately PDF-XChange begins processing the document printed by the GDI.

Declaration:**C#**

```
event OnDocSpooled(int JobID, string lpszDocName, string lpszAppName);
```

C++

```
HRESULT OnDocSpooled(LONG JobID, BSTR lpszDocName, BSTR lpszAppName);
```

VB

```
Event OnDocSpooled(JobID As Long, lpszDocName As String, lpszAppName As String)
```

These event parameters are identical to the parameters for the OnStartDoc event.

During handling for this event the application can set document specific properties (for example, the path where the PDF document should be stored as defined by the printing application).

The application should return S_OK from this event handler.

3.3.2.2 Event OnFileSaved

This event is fired when the PDF file is saved. This allows the application to perform any application-based processing to this file.

Declaration:**C#**

```
event OnFileSaved(int JobID, string lpszFileName);
```

C++

```
HRESULT OnFileSaved(LONG JobID, BSTR lpszFileName);
```

VB

```
Event OnFileSaved(JobID As Long, lpszFileName As String)
```

JobID

Internal Job identifier.

lpszFileName

Specifies the fully qualified path name for where the PDF document has been saved.

Return value

PDF-XChange ignores returned values from this event.

3.3.2.3 Event OnFileSent

This event is fired when emailing of a PDF file is enabled and after the email has been sent. If emailing is not enabled, this event will not be fired.

Declaration:**C#**

```
event OnFileSent(int JobID, string lpszFileName);
```

C++

```
HRESULT OnFileSent(LONG JobID, BSTR lpszFileName);
```

VB

```
Event OnFileSent(JobID As Long, lpszFileName As String)
```

Parameters for this event are the same as the parameters for the OnFileSaved event.

PDF-XChange ignores returned values from this event.

3.3.2.4 Event OnError

This event is fired when an error occurs during the processing of the PDF file. It further indicates that the events **OnFileSaved** and/or **OnFileSent** will not be fired for this document and processing of the document is terminated.

Declaration:

C#

```
event OnError(int JobID, int dwErrorCode);
```

C++

```
HRESULT OnError(LONG JobID, LONG dwErrorCode);
```

VB

```
Event OnError(JobID As Long, dwErrorCode As Long)
```

JobID

Internal Job identifier.

dwErrorCode

Specifies an error (code) occurred during document processing.

Return value

PDF-XChange ignores the returned values from this event.

3.4 Unsupported Property Declarations

All these methods will return an E_NOTIMPL error code:

- Event **LastPrinterEvent**
- Method **PrintDocument**
- Method **WaitForPrintEvent**

4 Tracker Software Products

Who are we and what do we do ?

We at Tracker Software Products take great pride in the software products we create and distribute. We sell our products directly, via Distributors, Resellers and OEM partners - in some cases with our products and larger partners these products are sold under different labels and names than those we sell directly, this is to allow our partners to build a following for their own brand and protect their future.

However - no matter how our products reach you, we want you to experience the best results possible - please do contact us if for any reason you are dissatisfied or have a suggestion how we can improve our product offerings.

You may also be interested in related products available from us - in the following brief topics you will find details of how to contact us, request support and summary details of the products available from us for both 'End Users' and Software Development tools for other Software developers to utilise

in their product offerings.

Please do contact us if you cannot find the information you require within this manual/help file.

4.1 Contact Us

How to Contact Us ...

Head Office

Tracker Software Products Ltd.
Units 1-3 Burleigh Oaks
East Street Turners Hill,
RH10 4PZ Sussex. England.

Tel: **+44 020 8123 4934** Sales/Administration
(pls do not use for support issue's)

Fax: **+44(0)1342-718060**

North America

Tracker Software Products
5100 Thimsen Ave
Suite 229
Minnetonka, MN 55345. USA.

Tel: **+001 (952) 232-0414**
(pls do not use for support issue's)

Our Web Site: <http://www.docu-track.com>

We also have offices and representatives in several other locations including : United States, France, Germany and Ukraine - in some instances after an initial contact with our UK office you may be referred to one of these locations if appropriate.

To contact us for support related issues:

Please see this FAQ page before contacting our support department - it may save you the task!

We recommend you use our Web Based User Support Forums and scan the existing library of questions and answers, if you don't find a suitable response then feel free to post your own - **all questions receive a response within 1 business day at worst!**

If for any reason you have difficulty linking to the forum or feel it is inappropriate for your needs then please email sales@docu-track.com , we regret we cannot answer support requests via telephone without a valid support contract. The number above is answered by administration staff who are not trained to assist with technical problems.

To Contact us for Sales/Administration related issues:

sales@docu-track.com End User, Developer and OEM.

admin@docu-track.com

All this information and a good deal more is available via our web site and the links provided.

Magazine reviews and press requests.

We are keen to assist in any way possible - please contact our sales department for any information or help you may require.

4.2 Products

Products Offered By Tracker Software Products Updates Can be downloaded from our Downloads page at our Web site

End User/Retail Products

You can Purchase Direct from our web site and be using any of our products the same day!

- PDF-XChange - Create fully native Adobe compatible PDF Files from virtually any Windows 32 Bit software application
- PDF-Tools - Create and manipulate Adobe PDF Files and batch Convert Images to PDF Files and more ...
- Raster-XChange - Create and manipulate Image files from any Windows document or application.
- TIFF-XChange Create and manipulate TIFF files from any Windows document or application.

Software Developers SDK's and other Products

You can Purchase Developer SDK's from our web site and be using any of our products the same day!

- PDF-XChange PRO SDK - Create fully native Adobe compatible PDF Files from your application output.
- PDF-Tools SDK - Create and manipulate Adobe PDF Files and batch Convert Images to PDF Files and more ...
- PDF-XChange Viewer SDK - Embed PDF viewing directly within your software applications (ActiveX and Simple DLL options available)
- Image-XChange SDK - Print, Convert, Scan and View Imaging formats!
- Raster-XChange SDK - Create and manipulate Image files from any Windows document or application.
- TIFF-XChange SDK - Create and manipulate TIFF files from any Windows document or application.
- **OCR-XChange SDK** (Coming soon - 2008) - Add Optical Character Recognition to your applications functionality.

Trial Versions

All of our products are available as fully functional evaluation downloads for you to try before you

buy - usually printing a demo watermark/stamp to differentiate between output created with the evaluation or licensed versions. We recommend that all users test the product they wish to buy first - thus ensuring you only buy when you are satisfied that the product meets your needs.

Refunds will not be issued except in exceptional circumstances once your license codes have been issued.

Trial versions are available from our web site - www.docu-track.com

For more details visit our web site or contact us by email.

4.3 Licensing

The PDF-XChange Drivers API SDK allows a developer to distribute their application with the Developers version of the PDF-XChange Drivers - Royalty Free to up to 100,000 users desktops without further cost - subject to our standard terms and conditions as outlined in the license agreement below.

Should you need a greater number of distribution licenses than outlined above - please do contact us and we will do all we can to provide a workable solution to suit your needs.

Please note a developer's license information should always be kept private and no developer key should ever be entered or used in the PDF-XChange Drivers GUI. Further a developer license never entitles the distribution of the End User version of the PDF-XChange drivers with an end user license key, this must always be specifically licensed for such purposes by our OEM sales department and a separate OEM License agreement is completed.

Once Licensed

Once you have purchased the PDF-XChange Drivers API SDK, you will receive your REG.KEY and DEV.CODE and these must be embedded in your project code and your project recompiled for 'Live' use before distribution to your Clients, otherwise all PDF pages created will bear the evaluation watermarks placed on the top corners of each page.

License Agreement.

License Agreement : PDF-XChange & PDF-Tools Software Developer Kits (SDK's) from Tracker Software Products Ltd 2001-8

PRINTED BELOW IN ITS ENTIRETY IS THE LICENSE AGREEMENT GOVERNING YOUR USE OF THE SOFTWARE. PLEASE READ THE LICENSE AGREEMENT.

IMPORTANT

TRACKER SOFTWARE PRODUCTS LTD. IS WILLING TO LICENSE THE ENCLOSED SOFTWARE TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THE LICENSE AGREEMENT PRINTED BELOW. PLEASE READ THE TERMS CAREFULLY BEFORE OPENING THE PACKAGE CONTAINING THE DISKETTE(S)/CD-R(S), Electronic File OR CLICKING THE ACCEPT BUTTON DURING INSTALLATION, AS SUCH CONDUCT INDICATES YOUR ACCEPTANCE TO ALL OF THE TERMS OF THIS LICENSE AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS, TRACKER SOFTWARE PRODUCTS LTD IS UNWILLING TO LICENSE THE SOFTWARE TO YOU, IN WHICH CASE YOU MUST IMMEDIATELY RETURN THE PACKAGE AND ALL ACCOMPANYING MATERIAL TO TRACKER SOFTWARE PRODUCTS LTD. OR YOUR AUTHORIZED DEALER FOR A FULL REFUND.

This License Agreement ("Agreement") is a legal agreement between Tracker Software Products Ltd, (Tracker), a Company registered in England, principally located in Turners Hill, West Sussex, England, and you, the user ("Licensee"), and is effective the date Licensee opens the package containing the diskette(s)/CD-R(s) or otherwise uses the enclosed

software product.

This Agreement covers all materials associated with Tracker's PDF-XChange API/SDK developer's toolkit products including the enclosed software product ("Software") **with the exception of the PDF-XChange PDF Viewer SDK for which there is a specific and separate license agreement.**

1. GRANT OF DEVELOPMENT LICENSE

TRACKER grants Licensee a non-exclusive, non-transferable, worldwide license for one (1) programmer to install the Software on a single personal computer and use the Software and one copy of the associated user documentation contained in the accompanying user manual, "online" help and Acrobat files ("Documentation") in the development of End User software application's as contemplated in section 2 below (herein, the "Application Software"). If additional programming seats are needed, Licensee should contact TRACKER for discounted license pricing. The license granted hereunder applies only to the designated version of the enclosed Software. If the Software is an upgrade or cross grade, it, and the product that was upgraded/cross graded constitute a single copy of the Software for purposes hereof and the new version and product that was upgraded/cross graded cannot be used by two people at the same time.

2. END USER APPLICATION

The Application Software developed by Licensee must be an "End User Application." An "end user application" is a specific application program that is licensed to a person or firm for business or personal use and not with a view toward redistributing the application or any part of the application, and may be either an application that is used by Licensee internally, or an application that is commercially distributed to end users for their use. A user of an end user application may not modify or redistribute the application and may not copy it (other than for archival purposes). Licensee's license agreement covering the Application Software must contain restrictions prohibiting redistribution, modification and copying of the Application Software. **The license rights hereunder do not apply to the development and deployment of software products such as Printer Drivers, ActiveX controls, plug-ins, authoring tools, development toolkits, compilers, operating systems and also software products where a primary or significant function is to generate or view 'PDF' format files (as defined by Adobe Systems Inc') and other file formats from 3rd party software applications not developed by the licensee, indirectly or otherwise, - such as Microsoft's 'Office' suite and component applications other than for the purpose of creating and then storing such files within a structured application for the archival and management of documents that is developed by the licensee and any other software not falling within the definition of an end user application.** If Licensee wishes to develop a product outside the scope of this license, Licensee should contact TRACKER'S OEM Sales department to see if a special license is available.

3. GRANT OF DUPLICATION AND DISTRIBUTION LICENSE

The Software includes certain runtime libraries and files intended for duplication and distribution by Licensee within the Application Software to the user of Application Software ("Redistributable's"). The Redistributable components of the Software are those files specifically designated as being distributable in the "Files to be Included with Your Application" section of the Online Help file, the terms of which are hereby incorporated herein by reference. Licensee should refer to the Documentation and specifically the "Online Help" file for additional information regarding the Redistributable's. Under TRACKER'S copyright, and subject to all the restrictions and conditions set forth in this Agreement and the Documentation, TRACKER hereby grants Licensee (and only Licensee) a non-exclusive, non-transferable, worldwide license to reproduce exact copies of the Redistributable's and include such files in the Application Software, and to deploy the Application Software internally and/or distribute the Application Software, directly or through customary distribution channels, to end users on a royalty free basis. (The foregoing sentence does not apply if Licensee has licensed Tracker's PDF-XChange Printer Driver for Windows – for 'End User' use (as Opposed to the API/SDK Toolkit Licensed products. This product requires additional run time licensing based on use/distribution of the Application Software: see Section 4, "Duplication and Distribution of Royalty Bearing Versions " below.) If Licensee wishes to use an OEM who will modify the Application Software and copy it, Licensee must first obtain an OEM distribution license from TRACKER or must require the OEM to obtain a license from TRACKER. Duplication or Redistribution of the Application Software, or any portion thereof, by the users of the Application Software, without a separate written redistribution license from TRACKER is prohibited. If the enclosed Software is packaged "For Evaluation Only," no right to copy and/or distribute the Redistributable's is granted. No rights to copy or redistribute the Application Software are granted until such time as Licensee has properly registered the Software with TRACKER and otherwise complied with this Agreement. Unless otherwise agreed in writing by Tracker, developer must distribute any Print drivers included using the Tracker Installation executable file provided for this purpose to ensure correct distribution and licensing adherence.

Once licensed, you may create and distribute a maximum of 100,000 end user application licenses incorporating any part of the allowed elements of this developer's kit - should the number of licenses you intend to distribute (or have already distributed) exceed this figure (other than for demonstration, evaluation or publicity purposes) then you must contact Tracker Software Products immediately and prior to (further) distribution or as soon as it becomes known to you that this figure has or will exceed this number, to discuss alternative licensing options. Further you agree at any time, on request and within 30 days of such request, to supply a duly audited and notarised account of application licenses delivered/sold where components of this licensed developer kit have been used and distributed – other than for

demonstration, evaluation or publicity purposes.

4. DUPLICATION AND DISTRIBUTION OF ROYALTY BEARING VERSIONS OF THE SOFTWARE

If the enclosed Software is Trackers PDF-XChange Printer Driver designed for 'End User' use as opposed to a component of the SDK kits available, (collectively known as "Royalty Bearing Versions") Licensee is required to enter into a separate duplication and distribution license prior to deployment of the Application Software. No duplication or distribution rights are granted hereunder with respect to the Royalty Bearing Versions.

5. OTHER RESTRICTIONS

The licenses granted under this Agreement are expressly conditioned upon Licensee's compliance with all the terms and conditions of this Agreement. Licensee may not use, copy, rent, lease, sell, sublicense, assign or otherwise transfer the Software except as expressly provided for in this Agreement. Licensee may make a reasonable number of archival copies of the Software. Except for the Redistributable's, Licensee shall not distribute any files contained in the Software, including without limitation, .CLW, .INC, .TPL, .CHM, .DRV, .LIB, .H, .MAK, .DEF, .TXT, .PDF or .HLP files. Licensee shall not reproduce, copy or transfer any Documentation, except Licensee may use the sample source code examples contained in the Documentation for the purpose of developing the Application Software. Upon TRACKER'S request, Licensee agrees to send TRACKER one demonstration copy of the Application Software. If the Software is PDF-XChange, the Application Software may only access the ActiveX and DLL file(s) directly and not through the PDF-XChange Print driver (.DRV file(s).) Licensee may only directly access the ActiveX and DLL file(s) if Licensee has a license in good standing for a PDF-XChange product with an API/SDK License such as PDF-XChange SDK or PDF-Tools SDK, and only then the appropriate components relevant to each SDK. Any distributor or reseller of Application Software appointed by Licensee must be subject to a binding agreement that includes provisions no less protective of TRACKER'S intellectual property rights in the Software as it is protective of Licensee's rights in its own software. Licensee acknowledges that the Software, in source code form, remains a confidential trade secret of TRACKER and/or its suppliers and therefore Licensee agrees that it shall not modify, de-compile, disassemble or reverse engineer the Software or attempt to do so except as permitted by applicable legislation. Licensee agrees to refrain from disclosing the Software (and to take reasonable measures with its employees to ensure they do not disclose the Software) to any person, firm or entity except as expressly permitted herein. Specifically, Licensee will not disclose or publish any unlock codes or instruction sets provided by TRACKER relating to the Software. If Licensee wishes to use the Software in a manner prohibited by this Agreement, Licensee should contact TRACKER'S OEM department to determine whether a special license may be obtained.

6. Use in Evaluation versions of Licensee Developer's End User Applications.

Where the Licensee provides a limited use evaluation version of their End User software – distribution of the PDF-XChange & Tools SDK redistributable components shall not count towards the licensee's allowed limited redistribution totals - provided that:

- 1: The functionality provided by the PDF-XChange or Tools SDK will cease on expiry of the evaluation period.
- 2: The evaluation period is no longer than 90 days – otherwise it must count towards the overall limited redistribution rights provided by the clients purchased license and may not be provided under the umbrella of this evaluation concession.
- 3: In the event that the Licensee allows some functionality of their application to continue after the evaluation period expires – this will not include the PDF-XChange or Tools SDK functions - otherwise it must count towards the overall limited redistribution rights provided by the clients purchased license and may not be provided under the umbrella of this evaluation concession.

If the Licensee is unable to ensure the above limited use restrictions are adhered to – Licensee may not incorporate the PDF-XChange & Tools SDK functionality in evaluation versions of their software applications – without accounting for each installation provided towards the total license count, allowed under the terms of this license.

7. PROPRIETARY RIGHTS; COPYRIGHT NOTICES

Except for the limited license granted herein, TRACKER, and its suppliers, retains exclusive ownership of all proprietary rights (including all ownership rights, title, and interest) in and to the Software. Licensee agrees not to represent that TRACKER is affiliated with or approves of Licensee's Application Software in any way. Except as required hereby, Licensee shall not use TRACKER'S name, trademarks, or any TRACKER designation in association with Licensee's Application Software. The Application Software may contain the following copyright notice in the "About box": "Portions of this product were created using PDF-XChange From Tracker Software Products Ltd ©2001-8, ALL RIGHTS RESERVED."

8. EXPORT LAW

Licensee acknowledges and agrees that the Software and Application Software may be subject to restrictions and controls imposed by the United States Export Administration Act, as amended (the "ACT"), and the regulations there under. Licensee agrees and certifies that neither the Software nor any direct product thereof (e.g. the Application Software) is being or will be acquired, shipped, transferred or re-exported, directly or indirectly, into any country

prohibited by the ACT and the regulations there under or will be used for any purpose prohibited by the same. Licensee acknowledges that the Software may include "technical data" subject to export and re-export restrictions imposed by U.S. law. Licensee bears all responsibility for export law compliance and will indemnify TRACKER against all claims based on Licensee's exporting of the Application Software.

9. U.S. GOVERNMENT RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19, as applicable. Manufacturer/Contractor is TRACKER SOFTWARE PRODUCTS LTD, Units 1-3, Burleigh Oaks, East Street, Turners Hill, West Sussex. England.RH10 4PZ

10. TERM

The license granted hereby is effective until terminated. Licensee may terminate the license by returning the Software and Documentation to TRACKER, without refund, and destroying all copies thereof in any form. TRACKER may terminate the licenses if Licensee fails to comply with any term or condition of this Agreement or any corresponding duplication and distribution agreement for Printer Driver Products. Upon such termination, Licensee shall cease using the Software and cease using or distributing the Application Software containing the Redistributable's. All restrictions prohibiting Licensee's use of the Software and intellectual property provisions relating to Software running to the benefit of TRACKER will survive termination of the license pursuant hereto. Termination will not affect properly granted end user licenses of the Application Software distributed by Licensee prior to termination.

11. EXCLUSION OF WARRANTIES

TRACKER and its suppliers offer and Licensee accepts the Software "AS IS." TRACKER and its suppliers do not warrant the Software will meet Licensee's requirements or will operate uninterrupted or error-free. ALL WARRANTIES, EXPRESS OR IMPLIED, ARE EXCLUDED FROM THIS AGREEMENT AND SHALL NOT APPLY TO ANY SOFTWARE LICENSED UNDER THIS AGREEMENT, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

12. LICENSEE'S REMEDIES: LIMITATIONS

LICENSEE'S SOLE AND EXCLUSIVE REMEDIES AGAINST TRACKER ON ANY AND ALL LEGAL OR EQUITABLE THEORIES OF RECOVERY SHALL BE, AT TRACKER'S SOLE DISCRETION, (A) REPAIR OR REPLACEMENT OF DEFECTIVE SOFTWARE; OR (B) REFUND OF THE LICENSE FEE PAID BY LICENSEE.

13. NO LIABILITY FOR CONSEQUENTIAL DAMAGES

In no event shall TRACKER, or its suppliers, be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information or other pecuniary loss) arising out of use of or inability to use the Software, even if TRACKER or its dealer have been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of certain implied warranties or the exclusion or limitation of incidental or consequential damages, in which case and to the extent such exclusion or limitation is not allowed, some of the foregoing limitations and exclusions may not apply to Licensee.

14. GENERAL

This Agreement shall be interpreted, construed, and enforced according to the laws of England. In the event of any action under this Agreement, the parties agree that courts located in England will have exclusive jurisdiction and that a suit may only be brought in England, and Licensee submits itself for the jurisdiction and venue of the courts located in England. This Agreement constitutes the entire agreement and understanding of the parties and may be modified only in writing signed by both parties. No officer, salesman, or agent has any authority to obligate TRACKER by any terms, stipulations or conditions not expressed in the Agreement. All previous representations and agreements, if any, either verbal or written, referring to the subject matter of this Agreement are void. If any portion of this Agreement is determined to be legally invalid or unenforceable, such portion will be severed from this Agreement and the remainder of the Agreement will continue to be fully enforceable and valid. This Agreement, and the rights hereunder, may not be transferred or assigned by Licensee, whether by oral or written assignment, sale of assets, merger, consolidation or otherwise, without the express written consent of TRACKER. Licensee agrees to be responsible for any and all losses or damages arising out of or incurred in connection with the Application Software. Licensee agrees to defend, indemnify and hold TRACKER harmless from any such loss or damage, including attorney's fees, arising from the use, operation or performance of the Application Software or Licensee's breach of any terms of this Agreement. Licensee shall be responsible for paying all state and federal use, sales or value added taxes, duties or governmental charges, whether presently in force or which come into force in the future, related to the distribution and sale of the Application Software and will indemnify TRACKER against any claim made against TRACKER relating to any such taxes or assessments.

Copyright © 2001-8 Tracker Software Products Ltd, Units 1-3, Burleigh Oaks, East Street, Turners Hill, West Sussex. England. RH10 4PZ

ALL RIGHTS RESERVED. All Other Trademark's acknowledged & are the property of their respective owners.

PDF-XChange Templates & Classes for Clarion for Windows (PDF-XChange-API/SDK customers only)

PDF-XChange API/SDK (PDF-XChange-API/SDK customers only)

PDF-XChange SDK Printer Driver (PDF-XChange-Print Driver customers only)

PDF-Tools SDK Templates & Classes for Clarion for Windows (PDF-Tools-API/SDK customers only)

Delphi Components for PDF-XChange and/or PDF-Tools SDK products.

All Demo/Evaluation components and examples for PDF-XChange and/or PDF-Tools SDK products.

Windows and MS Office are trademarks of Microsoft Inc. All other trademarks recognised and acknowledged.

Please note – no rights to use the PDF-XChange PDF Viewer SDK are contained herein, there is a specific and separate license agreement for the PDF-XChange Viewer SDK to which you must specifically complete and return prior to use for anything other than evaluation use.

Index

- C -

Contact 60

Contact Us 60

- N -

Notes - Clarion Developers 1

- O -

Other Products 61

- P -

Property Printer 5

- T -

Tracker Software Products 60

- W -

Watermarks 29