



PDF-XCHANGE



# PDF-XChange Drivers API USER MANUAL

## Head Office:

PDF-XChange Co. Ltd,  
P.O. Box 79,  
9622 Chemainus Road,  
Chemainus, British Columbia  
V0R 1K0  
Canada

## Sales:

Tel: (+00) 1-250-324-1621  
Fax: (+00) 1-250-324-1623  
sales@pdf-xchange.com  
<https://www.pdf-xchange.com>

## In Europe:

PDF-XChange Co. Ltd,  
Horsmanshoad,  
Pickwell Lane,  
Bolney, West Sussex  
RH17 5RH  
United Kingdom

## Sales:

Tel: (+00) 1-250-324-1621  
Fax: (+00) 1-250-324-1623  
sales@pdf-xchange.com  
<https://www.pdf-xchange.com>

## Copyright:

PDF-XChange is an internationally-registered trademark. International registration number: 1461163.  
© 2025. All rights reserved.



<b>1.</b>	<b>Welcome</b>	<b>5</b>
1.1	Introduction .....	6
1.2	Redistributable File Installation .....	9
1.2.1	System Requirements .....	14
<b>2.</b>	<b>Printing Process</b>	<b>15</b>
2.1	IPXCControlEx Object .....	16
2.2	IPXCPrinter Interface .....	19
2.2.1	Properties .....	20
2.2.1.1	Property Name .....	22
2.2.1.2	Property Options .....	23
2.2.1.2.1	Property Table Definition .....	24
2.2.1.2.2	Paper .....	25
2.2.1.2.3	Custom Forms .....	31
2.2.1.2.4	General .....	32
2.2.1.2.5	Compression .....	35
2.2.1.2.6	Graphics .....	38
2.2.1.2.7	Fonts .....	43
2.2.1.2.8	Watermarks .....	45
2.2.1.2.9	Overlays .....	46
2.2.1.2.10	Security .....	49
2.2.1.2.11	Links .....	52
2.2.1.2.12	Bookmarks .....	54
2.2.1.2.13	Default File Append .....	56
2.2.1.2.14	HeadersFooters .....	58
2.2.1.2.15	DocInfo (Document Information) .....	61
2.2.1.2.16	Save .....	63
2.2.1.2.17	EEmail .....	67
2.2.1.2.18	DigSig (Digital Signatures) .....	69
2.2.1.2.18.1	Optimization .....	72
2.2.1.2.19	Saver .....	75
2.2.1.2.20	RenderMode .....	76
2.2.2	Functions .....	77
2.2.2.1	SetRegInfo .....	78
2.2.2.2	ResetDefaults .....	79
2.2.2.3	ApplyOptions .....	80
2.2.2.4	AddBookmarkItem .....	81
2.2.2.5	AddTextWatermark .....	85
2.2.2.6	AddImageWatermark .....	91
2.2.2.7	StorePrinterOptions .....	94
2.2.2.8	ReStorePrinterOptions .....	95
2.2.2.9	SetAsDefaultPrinter .....	96
2.2.2.10	RestoreDefaultPrinter .....	97

2.2.2.11	AddSMTPServer .....	98
2.2.3	Events .....	100
2.2.3.1	Asynchronous Events .....	102
2.2.3.2	Synchronous Events .....	106
2.2.4	Unsupported Property Declarations .....	110
<b>2.3</b>	<b>Macros .....</b>	<b>111</b>
<b>Index</b>		<b>115</b>

## 1 Welcome

---



### PDF-XChange Drivers API V10 Manual

---

Welcome to the **PDF-XChange Drivers API V10** user manual. Use the **Table of Contents** on the left to browse the topics of this manual. Click topics to expand them.

We offer several further cutting-edge applications for the manipulation of PDF and image files. See the [PDF-XChange Products Page](#) for more information.

If you have any queries then please [Contact Us](#). We aim to respond to all communication within eight hours - and we are usually much faster. Additionally, the [User Forums](#) are an excellent resource for troubleshooting, and our [Knowledgebase](#) contains over four hundred articles about our software and other relevant information.

Examples and sample apps for this SDK and others are available [here](#).

A PDF version of this manual is available [here](#).

---

## 1.1 Introduction

---



### Introduction

---

The **PDF-XChange Drivers API** is an excellent choice for developers who want to add PDF-creation capability to their applications and/or manage the output-to-PDF of third party applications such as **MS Office**, **AutoCAD** and **Internet Explorer**. This is made possible via a set of 32/64-bit virtual print drivers that create amongst the smallest and most reliable PDF facsimiles available worldwide.

Additional developer kits are available from **PDF-XChange** to create, view, edit and manipulate PDF documents, including:

- The [PDF-XChange Core API SDK](#), which provides developers with libraries and API for the creation and manipulation of fully-native, industry standard PDF files.
- The [PDF-XChange Editor SDK](#), which enables the incorporation of a fully-licensed version of our best-selling software [PDF-XChange Editor](#) into developers' applications.
- The [PDF-XChange PRO SDK](#) bundle, which combines the features of the [PDF-XChange Drivers API SDK](#) and the [PDF-XChange Core API SDK](#), and also features a programmatic OCR module and access to all library DLL functions - including the creation of programmatic, fillable forms and digital signature capabilities.

See [here](#) for further information about our developer products and a comprehensive list of available applications.

Please note:

- The **PDF-XChange Drivers API SDK** is not a royalty-free tool kit. A fixed amount of licenses are included in each SDK product. Additional bulk-license packs are available for a reasonable price - see [here](#) for further information.
- This toolkit must not be used to develop toolkits/components for non-licensed developers.
- This toolkit must not be used to assist in the creation of printer drivers.
- The license agreement contains all relevant terms and conditions of use. If you are unsure whether your intended use would be in breach of license then please [Contact Us](#) for clarification. Our terms of licensing are flexible and we can often tailor them to meet the specific needs of our clientele.

- We recommend experimenting with the evaluation version prior to purchase. Evaluation versions are fully functional - but watermarks will be included on output pages. This means that you can develop your applications fully before making a purchase from **PDF-XChange**. When a licensed version is purchased all output becomes watermark-free. Our hope is that this can guarantee your satisfaction as we cannot offer refunds once a purchase has taken place.

## Licensing

Please use [IPXCControlEx Object](#)<sup>[16]</sup> to add your registration key to the product.

## Support

- Our [Developer Forums](#) are an excellent resource for troubleshooting.
- The [Adobe Website](#) is a very useful resource for developers working on applications for the purpose of creating and manipulating PDF formats.
- **PDF-XChange** also provides end-user and developer tool kits for creation/manipulation of PDF and raster image files and virtual printer drivers. See [here](#) for further information.

This manual is broken down into the following sections:

- The [Welcome](#)<sup>[5]</sup> section contains [System Requirements](#)<sup>[14]</sup> and [Installation](#)<sup>[9]</sup> information.
- The [Printing Process](#)<sup>[15]</sup> outlines the steps that take place when the **PDF-XChange Drivers API** is used to print documents. The following chapters are included:
  - The [IPXCControlEx Object](#)<sup>[16]</sup> is the property that enables virtual printing and returns the printing interface detailed below.
  - The [IPXCPrinter Interface](#)<sup>[19]</sup> is used to determine optional parameters in the PDF creation process:
    - The [Properties](#)<sup>[20]</sup> determine customizable settings such as page layout/ordering, PDF specification, text/image compression options, graphics processing and options for fonts, security, digital signatures, overlays, links, watermarks, headers and footers, document information (information fields and metadata), save options, automatic emailing, optimization and bookmarks.
    - The [Functions](#)<sup>[77]</sup> are used to perform **ActiveX** controls such as defining registration information, resetting properties to default values, applying customized properties, adding bookmarks/watermarks, storing/utilizing customized printer options, determining/restoring the default printer and defining the SMTP servers that the automatic email function uses.
    - The [Events](#)<sup>[100]</sup> are returned during printing via the [IPXCPrinter Interface](#)<sup>[19]</sup> and detail the stages of PDF creation. This section lists and defines all events.
    - The [Unsupported Property Declarations](#)<sup>[110]</sup> page lists properties that are not supported in this application.
  - The [Macros](#)<sup>[111]</sup> section details available macroinstructions that can be used in conjunction with the **PDF-XChange Drivers API** to name files and insert headers/footers.

## Windows Support

**PDF-XChange** has discontinued drivers for all Windows versions prior to Windows 2000 since version 4 of our products. All **PDF-XChange** version 3 products remain available - which means that Windows 9x/ME/NT is still supported - but it should be noted that no further development will take place for these products. Version 7 supports all Windows versions from Windows 2000.

## Troubleshooting

When installation and registration are complete the COM facilities provided are transparently available in most environments. However, this may be problematic for some 32-bit based IDEs (such as **Microsoft Visual Basic 6**) running in 64-bit Windows. If this is the case then then the type library (**pxccom.tlb**) should be manually referenced. It is located in the **PDF-XChange** installation directory.

---



## 1.2 Redistributable File Installation

---



### Redistributable File Installation

---

**PDF-XChange** provide a comprehensive installation executable for developers to distribute with their applications. This is the only method authorized for distribution. Reverse-engineering or reconfiguration of any kind is strictly prohibited and liable to immediate prosecution due to copyright infringement. Parties unwilling to adhere to this condition must not use **PDF-XChange Drivers API V10** within their applications.

All required driver options from **Windows 2000** are contained in our installer at both 32 and 64 bit. Developers can either pack the installer within their own installation package or provide it separately. If the latter option is used then it is necessary to call from the installer script. The command line switches detailed below can be used for this purpose.

The **PDF-XChange Drivers API** is available for redistribution from our [Website](#) and updated immediately when new versions are released. (The driver's API at 32/64 bit MSM is also available at this link and can be integrated into any msi installers). Alternatively, click [here](#) to download the driver. Please note that additional items are present in the installation folder, the default location of which is:

**"C:\Program Files\Tracker Software\PDF-XChange 7 API"**

Additional items are as follows:

- The **APIHelp** folder, which contains the license agreement and a link to the online version of the help manual.
- The **Examples** folder, which contains demo applications (and the files of code used to create them) in:
  - **C+/C++**
  - **C#**
  - **Delphi**
  - **Visual Basic**
  - Please note that executables for the demo applications are located in the **Bin** folder.
- The **Languages** folder, which contains language files that can be used to set the default UI language. (See below for further information).
- The **DS Error Lookup Tool**, which can be used to identify error codes.

- **PDF Saver**, a feature of [PDF-XChange Standard](#) that enables the combination of multiple print jobs into a single PDF document.

Note that the installer is based on **Inno Setup**, and exit codes are described [here](#).

## Silent Installation

This functionality is provided for users with multiple-user licenses and developer SDKs. Please note that the duplication of single-user licenses to multiple machines is illegal. A license is required for each machine on which the **PDF-XChange Drivers API** is installed with the exception of developer SDKs and the use of the application in evaluation mode.

When the installation of the **PDF-XChange Drivers API** is taking place on multiple workstations it may be useful for system administrators to perform a silent installation. This removes the need for user interaction and enables the installation to take place from the command line. Several command line options (detailed below) are available. Additionally, it is possible to create a .bat file for distribution with the main installation executable of the host software, which removes the need for a system administrator. However, it should be noted that full administrator privileges are required to install on later 'Win32' versions of Microsoft Windows.

A sample command line is detailed below. Please note that the string content section must begin and end with double quotation marks:

```
C:\PDFX7SA_sm.exe /VERYSILENT /NORESTART /COMPONENTS="pdfSaver,PDF-XChange driver, Help,Languagess" /DIR="C:\Program Files\PDF-XChange 5\" "/UserName:I am" "/Organization:Tracker software" "/UserEmail:my@email.com"
```

Available switches and their parameters/functions are detailed below. Please note that all components are case-sensitive:

SWITCH	DEFINITION
<b>MAIN</b>	<p>This is the full path to the installation executable that all forms of installation utilize. It is required in all cases.</p> <p><b>C:\PDFX7SA_sm.exe</b></p>
<b>/VERYSILENT</b>	<p>Designates a silent installation that requires no user interaction.</p>

<p><b>/SILENT</b></p>	<p>Designates an installation that requires basic user interaction.</p>
<p><b>/NORESTART</b></p>	<p>Disables the automatic, post-installation reboot. This is not recommended as the <b>PDF-XChange Drivers API</b> requires a reboot in order for changes to take effect.</p>
<p><b>/DIR=</b></p>	<p>The full path at which all required files are installed. If it is not available at the time of installation then it will be created. It is required in all cases.</p> <p><b>/DIR="C:\Program Files\PDF-XChange\"</b></p>
<p><b>/Group=</b></p>	<p>The Windows 'Start Menu' folder in which to locate the menu options provided during installation. It will be created as a child menu option of the 'Programs Files' menu and this cannot be changed.</p> <p><b>/GROUP="Tracker Software\PDF-XChange 7 API"</b></p>
<p><b>/COMPONENTS=</b></p>	<p>If this switch is specified then only listed components will be installed. If this switch is not specified then all components will be installed. Available components are:</p> <ul style="list-style-type: none"> <li>• <b>pdfSaver</b> (always required)</li> <li>• <b>PDF-XChange driver</b> (always required)</li> <li>• <b>Help</b></li> <li>• <b>Languagess</b> (please note 'ss' is required)</li> <li>• <b>Autounload</b></li> </ul> <p><b>/COMPONENTS="pdfSaver, PDF-XChange driver, Help,Languagess"</b></p>
<p><b>/AutoUnload:nn</b></p>	<p>If this option is specified then the PDFSaver4 module of the <b>PDF-XChange Drivers API</b> will unload after the specified timeout (nn, in minutes). This is a useful feature when the application is installed on a Windows Terminal Server/Citrix Metaframe with multiple users. It minimizes the memory in use when PDF creation is idle and automatically reloads as required.</p>

	<p><b>"/Autounload:03"</b></p>
<p><b>/Key</b></p>	<p>Allows license key registration during silent installation. If this information is not present then <b>PDF-XChange Drivers API</b> will function in demo mode. Please note that this switch must only be used in conjunction with an end-user enabled license and it is illegal to include it in a developer SDK license.</p> <p><b>"/Key:XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX"</b></p>
<p><b>/UserName</b></p>	<p>Allows username registration during silent installation.</p> <p><b>Username"/UserName:John Doe"</b></p>
<p><b>/Organization</b></p>	<p>Allows the registration of company details during silent installation.</p> <p><b>"/Organization:PDF-XChange"</b></p>
<p><b>/PDEFAULT=</b></p>	<p>Sets the <b>PDF-XChange Drivers API</b> as the default system printer.</p>
<p><b>/PName=</b></p>	<p>Sets the <b>PDF-XChange Drivers API</b> printer name.</p> <p><b>/PName="PDF-XChange for my application"</b></p>
<p><b>/UserEmail</b></p>	<p>Allows the registration of the user's email address during silent installation.</p> <p><b>"/UserEmail:JohnDoe@email.com"</b></p>
<p><b>/Lang</b></p>	<p>Sets a language as the UI default for all installed programs and print drivers. See installation folder for a list of available languages. The example below sets the language as German:</p> <p><b>PDFX7SA_sm.exe [other parameters] /LANG=gr [Additional parameters]</b></p>

<b>/DRVwoPRN</b>	Installs the print drivers without a visible "printer".

---

### 1.2.1 System Requirements

---



## System Requirements

---

The **PDF-XChange Drivers API SDK** supports all **Windows** (32/64 bit) operating systems from **Windows 7** and later. Please note the following:

- All versions from **V4** to **V10** are **Microsoft/Citrix Terminal Server** compatible.
  - All versions from **V5** to **V7** are **Windows XP/Vista** compatible.
  - We recommend that users install the latest **Microsoft Windows** service packs and updates before using our products, as doing so will ensure the greatest possible performance of the software.
  - Our products - in particular the printer drivers that **PDF-XChange Standard** and **PDF-XChange Lite** utilize - are not designed to work in virtualized environments such as the **XenApp** software.
  - There are some limitations to product support for **Windows XP** as **Microsoft** have stopped supporting it. Further information about this issue is available [here](#).
  - If you are using **Windows 7** then please ensure you have the latest release and all available fixes - otherwise you may encounter issues, as detailed [here](#).
-

## 2 Printing Process

---



### Printing Process

---

The **PDF-XChange Drivers API** is supplied as an **ActiveX** component that enables the control of the virtual printing element from our virtual printer [PDF-XChange Standard](#). Developers can embed the functionality of the virtual printer within their applications. The following process takes place when printing PDF documents in developers' applications:

1. A new instance of the virtual printer is created via the **IPXCControlEx** interface. See [here](#)<sup>[16]</sup> for further information.
  2. The virtual printer (which will display as an **IPXCPrinter** interface) is used to set up properties for printing documents. See [here](#)<sup>[20]</sup> for a list of available properties and their parameters.
  3. Documents are printed via the virtual printer (from either the developer's application or a third-party application run from within the application).
  4. The virtual printer instance is closed and destroyed.
-

## 2.1 IPXControlEx Object

---



### IPXControlEx Object

---

The **IPXControlEx Object** is used to create a new **PDF-XChange Drivers API** virtual printer instance and then returns the **IPXCPrinter** object. This allows the host application to control the properties of the printer instance and print the relevant documents. When the operation is complete the virtual printer instance is destroyed.

Please note that all properties, code declarations and parameters are case-sensitive.

#### IPXControlEx.Printer

This property enables the creation of a new printer instance from the **PDF-XChange Drivers API** and returns the printed **IPXCPrinter** object:

#### 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)
```

## Parameters

### *pServerName*

This argument is reserved for future usage and should be a NULL or empty string.

### *pPrinterName*

Specifies the name of the created printer. If a printer of the same name already exists then a suffix will be generated for the new printer.

### *pRegKey*

This is the string that contains the developer registration key. If this parameter is absent/invalid then demo labels will be added to all generated pages.\*

### *pDevCode*

This is the string that contains the developer code. If this parameter is absent/invalid, then demo labels will be added to all generated pages. Please note that *pDevCode* is only needed when the older style license keys are used. If a new style of key is being used then the *pDevCode* parameter should be *NULL* or just an empty string. This is because the new style keys include the information necessary for the *pDevCode* parameter, but we must retain the property as there are still users who have the older style license keys.

\* This item is provided when a license is purchased. If projects are developed in evaluation mode then they must be recompiled in order to include the license strings. All output created in evaluation mode will need to be recreated.

## Return Value

If the function fails then the return value is a NULL.

If the function succeeds then the return value is an **IPXCPrinter** object.

## Example (VB)

```
Dim PDFPFFactory As New PXCComLib5.CPXCControlEx
Dim WithEvents PDFPrinter As PXCComLib5.CPXCPrinter

Public Sub Form_Load()
Set PDFPrinter = PDFPFFactory.Printer("", "PDF-XChange 2012 Sample", "<reg>", "<dev>");
End Sub

Public Sub Form_Unload(Cancel As Integer)
```

```
Set PDFPrinter = Nothing  
End Sub  
</dev></reg>
```

---

## 2.2 IPXCPrinter Interface

---



### IPXCPrinter Interface

---

The **IPXCControlEx Object** returns the **IPXCPrinter Interface**, which is used to control all options of the PDF creation process. The **IPXCPrinter Interface** is released when printing is complete. This process also removes the associated virtual printer.

Events that the **IPXCPrinter Interface** provides can be used to control the printing process. The **IPXCPrinter Interface** provides the following options:

- [Properties](#) 20
  - [Functions](#) 77
  - [Events](#) 100
-

### 2.2.1 Properties

---



## Properties

---

The **IPXCPrinter Interface** provides the following properties:

- **Property Name**<sup>[22]</sup> returns the printer name.
- **Property Options**<sup>[23]</sup> is used to specify most of the printer properties. Options are detailed below:
  - **Paper**<sup>[25]</sup> defines the page layout customization of printed documents:
    - **Normal** is used to specify the position and size of the page on the sheet.
    - **n-UP** is used to define the amount and order of pages.
    - **Booklet** is used to define the parameters of booklet/book creation.
  - **Custom Forms**<sup>[31]</sup> is used to create forms of custom sizes.
  - **General**<sup>[32]</sup> defines the specifications to which printed PDFs conform and the PDF viewing application options.
  - **Compression**<sup>[35]</sup> defines text and image compression options:
    - **Color** specifies the compression for True color images.
    - **Indexed** specifies the compression for images that feature a color palette.
    - **Mono** specifies the compression method for monochrome/black & white images.
  - **Graphics**<sup>[38]</sup> defines how images are processed before they are inserted into documents:
    - **Color** determines image downsampling and conversion options for True color images:
      - **Color.Downsampling** defines downsampling options.
      - **Color.Convert** defines conversion options for color images into grayscale/monochrome/black & white.
    - **Indexed** determines downsampling and conversion options for indexed images that feature palettes:
      - **Indexed.Downsampling** defines downsampling options.
      - **Indexed.Convert** defines conversion options for indexed images into grayscale/monochrome/black & white.
    - **MonoAndDrawings** determines downsampling and conversion options for monochrome images and black & white drawings:
      - **MonoAndDrawings.Downsampling** defines downsampling options.
      - **MonoAndDrawings.Convert** defines conversion options for color images into grayscale/monochrome/black & white.

- [Fonts](#)<sup>[43]</sup> defines font properties.
  - [Watermarks](#)<sup>[45]</sup> defines and enables/disables watermarks.
  - [Overlays](#)<sup>[46]</sup> defines the overlay of existing files onto printed documents.
  - [Security](#)<sup>[49]</sup> defines security properties.
  - [Links](#)<sup>[52]</sup> defines link properties.
  - [Bookmarks](#)<sup>[54]</sup> defines bookmark creation options.
  - [Default File Append](#)<sup>[56]</sup> is used used to append/prepend files to existing documents.
  - [HeadersFooters](#)<sup>[58]</sup> defines headers and footers.
  - [DocInfo](#)<sup>[61]</sup> defines PDF document information fields and XMP metadata.
  - [Save](#)<sup>[63]</sup> defines the folder in which PDFs are stored, and additional save data.
  - [EMail](#)<sup>[67]</sup> defines automatic emailing properties for printed documents.
  - [Optimization](#)<sup>[72]</sup> defines optimization properties.
  - [DigSig](#)<sup>[69]</sup> defines digital signature properties.
  - [Saver](#)<sup>[75]</sup> defines the **ShowProgress** property of the **pdfsaver** application.
-

### 2.2.1.1 Property Name

---



## Property Name

---

This is a read-only property that returns the name of the printer associated with the object. Please note that declarations are case-sensitive:

### Declaration

#### C#

```
string Name;
```

#### C++

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

#### (VB)

```
Property Name As String
```

### Comments

If the requested printer name already exists the the returned name will be different from the name passed in the printer properties section of the [IPXCControlEx Object](#).<sup>[16]</sup>

---

### 2.2.1.2 Property Options

---



## Property Options

---

This is a write-only property option that uses parameters to specify most of the **PDF-XChange Drivers API** properties. Please note that declarations are case-sensitive:

### 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)
```

### Comments

Strings for the relevant *pOptionName* are used to identify properties. Properties contain strings, integers, boolean or binary values. A comprehensive list of properties is available [here](#).<sup>[20]</sup>

---

### 2.2.1.2.1 Property Table Definition

---



## Property Table Definition

---

Tables are used to detail available property values. There are three columns in each table:

- **Name** is the property name to which values are assigned.
- **Values** details available property values.
- **Definition** defines the property function.

Please note that values are described as numbers and corresponding strings, either of which can be used to set property values. For example, **0 (Normal)** means the property can be set as the integer value **0** or the string "**Normal**" and achieve the same result:

```
pPrinter.Option["Paper.LayoutType"] = 0  
pPrinter.Option["Paper.LayoutType"] = "Normal"
```

### Boolean Values

If the **Values** section of the property's table is a Boolean value then the following options apply:

- When the value is **0** the following strings are acceptable: **No, False, Disabled.**
- When the value is **1** the following strings are acceptable: **Yes, True, Enable.**

Therefore the following strings achieve the same result:

```
pPrinter.Option["Compress.Graphics"] = 1  
pPrinter.Option["Compress.Graphics"] = "Enable"
```

---



## 2.2.1.2.2 Paper



### Paper

The **Paper** properties define the page layout of printed documents. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>LayoutType</b>	<b>0 (Normal)</b> <b>1 (n-UP)</b> <b>2 (Booklet)</b>	Defines the page layout of printed documents. See tables below for further information.  Default value: <b>0 (Normal)</b>
<b>MirrorX</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables/disables the horizontal mirroring of all drawings.  Default value: <b>False</b>
<b>MirrorY</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables/disables the vertical mirroring of all drawings.  Default value: <b>False</b>
<b>SheetSizeIndex</b>	All values from <b>-2 to 41</b>	Defines the sheet used for PDF creation.  Value <b>-2</b> is a custom sheet size that the <b>SheetWidth</b> and <b>SheetHeight</b> properties define (see below).  Value <b>-1</b> is an automatic sheet size defined by the layout type and page size.  All other values are indices of standard paper sizes. See the <a href="#">List of Standard Paper Sizes</a> <sup>[28]</sup> table below.

		Default value: <b>-1</b>
<b>SheetWidth</b>	All values from <b>13-5080</b>	Defines sheet width. Values are in tenths of millimeters. Default value: <b>2100</b>
<b>SheetHeight</b>	All values from <b>13-5080</b>	Defines sheet height. Values are in tenths of millimeters. Default value: <b>2970</b>

## Normal Layout

These properties define the parameters when the standard layout is used. Please note that the maximum height/width of PDF pages is 5080mmx5080mm.

NAME	VALUES	DEFINITION
<b>LeftPos</b>	Integer	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.
<b>TopPos</b>	Integer	This property is used when the <b>Centre</b> and <b>Fit</b> properties have <b>False</b> values.  Default value: <b>0</b>
<b>PageScale</b>	Integer	Determines the scale level when positioning pages on sheets. Values are in 0.01%. This property is only used if the <b>Fit</b> property (below) is <b>False</b> .  Default value: <b>1000</b>
<b>Center</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	If this property is <b>True</b> then pages are centered on sheets. Otherwise the <b>LeftPos</b> and <b>TopPos</b> properties (above) determine page positioning. If the <b>Fit</b> property is <b>True</b> then this property is not used.  Default value: <b>True</b>

<b>Fit</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Fits the page size to the sheet size.  Default value: <b>True</b>
------------	---	---

## n-UP Layout

These properties define the amount and order of pages on each sheet:

NAME	VALUES	DEFINITION
<b>Nup</b>	<b>0</b> - one page per sheet <b>1</b> - two pages per sheet <b>2</b> - four pages per sheet <b>3</b> - six pages per sheet <b>4</b> - eight pages per sheet <b>5</b> - nine pages per sheet <b>6</b> - sixteen pages per sheet	Defines the number of pages per sheet.  Default value: <b>0</b>
<b>NupOrder</b>	<b>0 (AcrossFromLeft)</b> <b>1 (DownFromLeft)</b> <b>2 (AcrossFromRight)</b> <b>3 (DownFromRight)</b>	Defines the order in which pages are placed.  Default value: <b>0 (AcrossFromLeft)</b>
<b>Borders</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables/disables page borders.  Default value: <b>False</b>

## Booklet Layout

These properties define booklet creation parameters:

NAME	VALUES	DEFINITION
<b>BookletType</b>	<b>0 (Brochure)</b>	

	<b>1 (Book)</b>	<p>If the value of this property is <b>1 (Book)</b> then pages will be grouped typographically and according to the number specified in the <b>SheetsInBook</b> property.</p> <p>If the value of this property is <b>0 (Brochure)</b> then then all pages will be included in a single book.</p> <p>Default value: <b>0 (Brochure)</b></p>
<b>SheetsInBook</b>	<b>1-64</b>	<p>Defines the number of sheets in one book when the <b>BookletType</b> property is set to <b>Book</b>.</p> <p>Default value: <b>4</b></p>
<b>GutterSize</b>	Positive integer	<p>Defines the booklet gutter size. Values are in tenths of millimeters.</p> <p>Default value: <b>127</b></p>

**Standard Paper Sizes**

This table details predefined, standard paper sizes in the **PDF-XChange Drivers API**. Please note:

- The figures in the **Index** column can be used to specify sheet size via the **Paper.SheetSizeIndex** property (see above).
- The figures in the **Standard Index** column are the index values for paper as defined in the **Windows API**. Undefined values are specific to the **PDF-XChange Drivers API** and do not have a standard index.
- The figures in the **Dimension** column define the paper size in standard (portrait) orientation.
- Please note that the **Custom Forms** properties can be used to create custom forms.

<b>INDEX</b>	<b>STANDARD INDEX</b>	<b>PAPER NAME</b>	<b>DIMENSION</b>
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	85 x 14"
8	6	Statement	5.5 x 8.5"
9	7	Executive	7.25 x 10.5"
10	-	A0	841 x 1189mm
11	-	A1	594 x 841mm
12	62	A2	420 x 594mm
13	8	A3	297 x 420mm
14	9	A4	210 x 297mm
15	11	A5	148 x 210mm
16	-	B0	1000 x 1414mm
17	-	B1	7070 x 1000mm
18	-	B2	5000 x 7070mm
19	42	B4 (ISO)	250 x 353.6mm
20	12	B4 (JIS)	257 x 364mm
21	34	B5	176 x 250mm
22	13	B5 (JIS)	182 x 257mm
23	14	Folio	8.5 x 13"
24	15	Quarto	215 x 275mm
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 220mm
31	28	Envelope C5	162 x 229mm
32	29	Envelope C3	324 x 458mm
33	30	Envelope C4	229 x 324mm
34	31	Envelope C6	114 x 162mm
35	32	Envelope C65	114 x 229mm
36	35	Envelope B6	125 x 176mm
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"

---

## 2.2.1.2.3 Custom Forms

---



### Custom Forms

---

The **Custom Forms** properties can be used to create forms of custom sizes. The options set **CustomForms.Items** is an indexed container (array) that features the following option sets:

NAME	VALUES	DEFINITION
<b>ID</b>	String	The identifier of the form.
<b>Name</b>	String	The display name of the form.
<b>Unit</b>	<b>0 (millimeters)</b> <b>1 (inches)</b>	Specifies the units of measurement used.
<b>Width</b>	All values from <b>0</b> <b>to 5080.</b>	Specifies the width of the form.
<b>Height</b>	All values from <b>0</b> <b>to 5080.</b>	Specifies the height of the form.

Please note that all properties are case-sensitive:

---

## 2.2.1.2.4 General



### General

The **General** properties determine the specifications of printed PDF files and the parameters of the viewing application:

NAME	VALUES	DEFINITION
Specification	<b>0 (Auto)</b> <b>3 (1.3)</b> <b>4 (1.4)</b> <b>5 (1.5)</b> <b>6 (1.6)</b> <b>7 (1.7)</b> <b>-1 (PDF/A-1b)</b> <b>-2 (PDF/A-1a)</b> <b>-3 (PDF/A-2a)</b> <b>-4 (PDF/A-2b)</b> <b>-5 (PDF/A-2u)</b> <b>-6 (PDF/A-3a)</b> <b>-7 (PDF/A-3b)</b> <b>-8 (PDF/A-3u)</b>	<p>Defines the specification level to which the PDFs conform.</p> <p>The value <b>Auto (0)</b> enables the <b>PDF-XChange Drivers API</b> to choose the most efficient specification based on other specifications.</p> <p>When a negative value is specified, the <b>PDF-XChange Drivers API</b> creates a PDF file that conforms to an ISO standard PDF/A. Please note that some content/functions are unavailable in the PDF/A format.</p> <p>Default value: <b>0 (Auto)</b></p>
PageLayout	<b>0 (Default)</b> <b>1 (SinglePage)</b> <b>2 (OneColumn)</b> <b>3 (TwoColumnLeft)</b> <b>4 (TwoColumnRight)</b> <b>5 (TwoPagesLeft)</b> <b>6 (TwoPagesRight)</b>	<p>Default value: <b>0 (Default)</b></p>
PageMode	<b>0 (Default)</b> <b>1 (ShowNone)</b> <b>2 (Bookmarks)</b>	<p>Default value: <b>0 (Default)</b></p>



	<b>3 (Thumbnails)</b> <b>4 (Layers)</b> <b>5 (Attachments)</b> <b>6 (FullScreen)</b>	
<b>FullScreenMode</b>	<b>0 (Default)</b> <b>1 (ShowNone)</b> <b>2 (Bookmarks)</b> <b>3 (Thumbnails)</b> <b>4 (OptionalContent)</b>	Default value: <b>0 (Default)</b>
<b>MagnificationType</b>	<b>0 (Default)</b> <b>1 (ActualSize)</b> <b>2 (FitPage)</b> <b>3 (FitWidth)</b> <b>4 (FitHeight)</b> <b>5 (Level)</b>	Default value: <b>0 (Default)</b>
<b>MagnificationLevel</b>	All values from <b>1-6400</b>	Default value: <b>100</b>
<b>OpenToPage</b>	Positive integers from zero to the last document page number.	Default value: <b>0</b>
<b>HideToolbars</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Default value: <b>False</b>
<b>HideMenubar</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Default value: <b>False</b>
<b>HideUI</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Default value: <b>False</b>
<b>FitWindow</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Default value: <b>False</b>
<b>CenterWindow</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Default value: <b>False</b>

<b>ShowDocTitle</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Default value: <b>False</b>
---------------------	---	-----------------------------

Please note that all properties are case-sensitive:

---

## 2.2.1.2.5 Compression



### Compression

The **Compression** properties determine the compression format used for text and images. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Graphics</b>	<a href="#">Boolean Value</a> <sup>24</sup>	Specifies PDF document compression. If this value is true then compression methods for different image types are defined in the <b>Color</b> , <b>Indexed</b> and <b>Mono</b> subsections.  Default value: <b>True</b>
<b>Text</b>	<a href="#">Boolean Value</a> <sup>24</sup>	If this value is true then text-based content of PDF pages are compressed in order to reduce the resultant file size.

### Color

These properties specify the compression for True color images:

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> <sup>24</sup>	Default value: <b>True</b>
<b>Method</b>	0 (Auto) 1 (JPEG)	

	<p>2 (ZIP) 3 (JPEGZIP) 4 (J2K) 5 (J2KZIP)</p>	<p>Specifies the compression method used for the compression of True color images.</p> <p>Please note that if the PDF/A format is being used then the compression formats <b>J2K</b> and <b>J2KZIP</b> cannot be used.</p> <p>Default value: <b>0 (Auto)</b></p>
JPEGQuality	1-100	<p>Specifies the image quality when JPEG compression is used (both <b>JPEG</b> and <b>JPEGZIP</b>). Please note that increasing quality will increase image definition but result in a larger compressed stream, resulting in a larger file.</p> <p>Default value: <b>75</b></p>


**Indexed**

These properties specify the compression for indexed images:

NAME	VALUES	DEFINITION
Enabled	<p><a href="#">Boolean Value</a><sup>[24]</sup></p>	<p>If this property is true then indexed images will be compressed in the manner specified in the <b>Method</b> section. Otherwise they will be stored in PDF and without compression.</p> <p>Default value: <b>True</b></p>
Method	<p>0 (Auto) 1 (Packbits) 2 (ZIP) 3 (LZW)</p>	<p>Defines the compression method to be used to compress images with palette.</p> <p>Please note that <b>LZW</b> compression cannot be used with the <b>PDF/A</b> output specification.</p> <p>Default value: <b>0 (Auto)</b></p>

**Mono**

These properties specify the compression for monochrome/black & white images:

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> 	Default value: <b>True</b>
<b>Method</b>	<b>0 (Auto)</b> <b>1 (ZIP)</b> <b>2 (CCITT3)</b> <b>3 (CCITT4)</b> <b>4 (RLE)</b> <b>5 (JBIG2)</b>	<p>Defines the compression method to be used to compress monochrome/black &amp; white images.</p> <p>Please note that <b>JBIG2</b> compression cannot be used with the <b>PDF/A</b> output specification.</p> <p>Default value: <b>0 (Auto)</b></p>
<b>JBIG2Method</b>	<b>0 (Standard)</b> <b>1 (CropBorders)</b> <b>2 (Symbols)</b>	<p>Specifies preprocessing method of images compressed using JBIG2 compression.</p> <p>Default value: <b>0 (Standard)</b></p>

## 2.2.1.2.6 Graphics



### Graphics

The **Graphics** properties contains three sections that define how images are processed before they are inserted. Please note that all properties are case-sensitive:

- **Color** determines image downsampling and conversion options for **True Color** images.
- **Indexed** determines image downsampling and conversion options for indexed images with palettes:
- **MonoAndDrawings** determines downsampling and conversion options for monochrome images and black & white drawings:

#### Color

This property determines image downsampling and conversion options for **True Color** images:

- The **Downsample** subsection defines when and how images are downsampled.
- The **Convert** subsection defines color reduction parameters.

#### Color.Downsampling

This subsection determines when and how images are downsampled:

NAME	VALUES	DEFINITION
<b>Method</b>	<b>0 (None)</b> <b>1 (Linear)</b> <b>2 (Bilinear)</b> <b>3 (Bicubic)</b>	Specifies the downsampling method. If the value of this property is <b>0 (None)</b> then color images will not be downsampled.  Default value: <b>0 (None)</b>
<b>IfAbove</b>	All values from <b>51-2400</b>	

		<p>If the original DPI of the image is equal to or above the value specified for this property then images are downsampled to the DPI specified for the <b>To</b> property.</p> <p>Default value: <b>128</b></p>
<b>To</b>	All values from <b>50-2399</b>	<p>The size to which color images are downsampled.</p> <p>Default value: <b>96</b></p>

## Color.Convert

This subsection determines conversion options for color images:

NAME	VALUES	DEFINITION
<b>Method</b>	<p><b>0 (None)</b>  <b>1 (ToGrayscale)</b>  <b>2 (ToMonochrome)</b>  <b>3 (ToBlackAndWhite)</b></p>	<p>Specifies how color images are converted. If the value of this property is <b>0 (None)</b> then color images will not be converted.</p> <p>Default value: <b>0 (None)</b></p>
<b>Dither</b>	<p><a href="#">Boolean Value</a> <sup>[24]</sup></p>	<p>Enables/disables dithering during conversion. This is relevant for only the <b>ToMonochrome</b> and <b>ToBlackAndWhite</b> values.</p> <p>Default value: <b>True</b></p>

## Indexed

This section determines image downsampling and conversion options for indexed images with palettes:

- The **Downsample** subsection defines when and how images are downsampled.
- The **Convert** subsection specifies color reduction and conversion options for indexed images.

## Indexed.Downsampling

This subsection defines downsampling options:

NAME	VALUES	DEFINITION
<b>Method</b>	<b>0 (None)</b> <b>1 (Linear)</b> <b>2 (Bilinear)</b> <b>3 (Bicubic)</b>	Specifies the downsampling method. If this property is set to <b>0 (None)</b> then indexed images will not be downsampled.  Default value: <b>0 (None)</b>
<b>IfAbove</b>	All values from <b>51-2400</b>	If the original DPI of the image is equal to or above the value specified for this property then images are downsampled to the DPI specified for the <b>To</b> property.  Default value: <b>128</b>
<b>To</b>	All values from <b>50-2399</b>	The size to which indexed images are downsampled.  Default value: <b>96</b>

**Indexed.Convert**

This subsection determines conversion options for indexed images:

NAME	VALUES	DEFINITION
<b>Method</b>	<b>0 (None)</b> <b>1 (ToGrayscale)</b> <b>2 (ToMonochrome)</b> <b>3 (ToBlackAndWhite)</b>	Specifies the conversion method. If this property is set to <b>0 (None)</b> then indexed images will not be converted.  Default value: <b>0 (None)</b>
<b>Dither</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Specifies whether or not images are dithered during conversion. It is relevant to only the <b>ToMonochrome</b> and <b>ToBlackAndWhite</b> values.  Default value: <b>True</b>



## MonoAndDrawings

This section determines downsampling and conversion options for monochrome images and black & white drawings:

- The **Downsample** subsection determines when and how images are downsampled.
- The **Convert** subsection determines conversion options for monochrome/black & white images.

### MonoAndDrawings.Downsampling

This subsection defines downsampling options:

NAME	VALUES	DEFINITION
<b>Method</b>	<b>0 (None)</b> <b>1 (Linear)</b> <b>2 (Bilinear)</b> <b>3 (Bicubic)</b>	Specifies the downsampling method. If the value of this property is <b>0 (None)</b> then monochrome images and black & white drawings will not be downsampled.  Default value: <b>0 (None)</b>
<b>IfAbove</b>	All values from <b>51-2400</b>	If the original DPI of the image is equal to or above the value specified for this property then images will be downsampled to the DPI specified for the <b>To</b> property.  Default value: <b>128</b>
<b>To</b>	All values from <b>50-2399</b>	The size to which monochrome images and black & white images are downsampled.  Default value: <b>96</b>

### MonoAndDrawings.Convert

This subsection determines conversion options for monochrome images and black & white drawings.

		DEFINITION

NAME	VALUES	
<b>Method</b>	<b>0 (None)</b> <b>1 (ToGrayscale)</b> <b>2 (ToMonochrome)</b> <b>3 (ToBlackAndWhite)</b>	<p>Defines how mono images and line art are converted. If the value is set to <b>0 (None)</b> then no conversions will take place.</p> <p>Default value: <b>0 (None)</b></p>
<b>Threshold</b>	All values from 1-255	<p>Defines the level behind which line art colors are treated as black. The grayscale level of line art color is used as a comparison.</p> <p>Default value: <b>128</b></p>

---

## 2.2.1.2.7 Fonts



### Fonts

The **Fonts** properties determine document font options. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>EmbedAll</b>	<a href="#">Boolean Value</a> <sup>24</sup>	<p>If this property is set to <b>True</b> then all fonts are embedded. This increases file size but also ensures that documents can be viewed correctly on all computers.</p> <p>Default value: <b>False</b></p>
<b>EmbedProtected</b>	<a href="#">Boolean Value</a> <sup>24</sup>	<p>If this property is set to <b>True</b> then only protected fonts are embedded.</p> <p>Default value: <b>True</b></p>
<b>WriteToUnicode</b>	<a href="#">Boolean Value</a> <sup>24</sup>	<p>If this property is set to <b>True</b> then fonts are written to unicode.</p> <p>Default value: <b>False</b></p>
<b>Subset</b>	<a href="#">Boolean Value</a> <sup>24</sup>	<p>If this property is set to <b>True</b> and the percentage of font characters used is less than that specified in the <b>SubsetLevel</b> property (below) then only used characters will be embedded.</p> <p>If this property is set to <b>False</b> then the complete font file will be embedded, which will increase file size but ensure formatting consistency and that fonts can always be viewed correctly.</p> <p>Default value: <b>True</b></p>

<b>SubsetLevel</b>	All values from <b>1-100</b>	Default value: <b>100</b>

There are two identical subsections: **AlwaysEmbed** and **NeverEmbed**. These define lists of fonts that should always/never be embedded. These following properties are available in each subsection:

NAME	VALUES	DEFINITION
<b>Enable</b>	<a href="#">Boolean Value</a> <sup>24</sup>	Enable using the list defined by the <b>Fonts</b> property (below). Default value: <b>False</b> .
<b>Fonts</b>	String	Contains a list of fonts delimited with commas. Default value: <b>empty list</b> .

## 2.2.1.2.8 Watermarks



### Watermarks

The **Watermarks** properties enable/disable and define watermarks in printed documents. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	If set to <b>True</b> then watermarks will be added to printed documents as defined in the <b>Watermarks</b> property below.  Default value: <b>False</b>
<b>Watermarks</b>	String	A list of watermarks, delimited with semicolons, to be added to the document. The <a href="#">AddTextWatermark</a> <sup>[85]</sup> or <a href="#">AddImageWatermark</a> <sup>[91]</sup> functions can be used to add watermarks.  Default value: <b>empty string</b>

## 2.2.1.2.9 Overlays



### Overlays

The **Overlays** properties determine how existing PDF files are used to overlay new documents. Please note that overlays and the **PDF/A** format are not compatible. Additionally, all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables/disables overlay properties. If set to <b>False</b> then no further properties from this section are used.  Default value: <b>False</b>
<b>OverlayFile</b>	String	Specifies the file used as the overlay. The value must be a fully-qualified path name.  Default value: <b>empty string</b>
<b>OverlayPassword</b>	String	If the file specified as the <b>OverlayFile</b> property is encrypted then this property must contain the password.  Default value: <b>empty string</b>
<b>AsForeground</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Defines the location of overlay pages. If this property is set to <b>True</b> then the default location is the document background.  Default value: <b>False</b>
<b>Repeat</b>	<b>0 (NoRepeat)</b> <b>1 (RepeatLastPage)</b>	Specifies the repeat option used for the overlay:

	<p><b>2 (RepeatContinuous)</b></p>	<p><b>0 (NoRepeat):</b> applies each page of the overlay file to the corresponding page of the new file. If the new file has a greater number of pages than the overlay file then the additional pages will not feature an overlay.</p> <p><b>1 (RepeatLastPage):</b> applies each page of the overlay file to the corresponding page of the new PDF file. If the new file has a greater number of pages than the overlay file then the additional pages will all feature the last page of the overlay file. If the overlay file features only one page then it will be placed on all the pages of the new file.</p> <p><b>2 (RepeatContinuous):</b> applies each page of the overlay file to the corresponding page of the new PDF file. If the new file has a greater number of pages than the overlay file then the overlay file will repeat from the beginning for the remaining pages. If the overlay file features only one page then it will be placed on all the pages of the new file.</p> <p>Default value: <b>0 (NoRepeat)</b></p>
<p><b>AlignH</b></p>	<p><b>0 (Left)</b> <b>1 (Center)</b> <b>2 (Right)</b></p>	<p>Specifies (relative to printed pages) the horizontal alignment of overlay pages.</p> <p>Default value: <b>1 (Center)</b></p>
<p><b>AlignV</b></p>	<p><b>0 (Top)</b> <b>1 (Middle)</b> <b>2 (Bottom)</b></p>	<p>Specifies (relative to printed pages) the vertical alignment of overlay pages.</p> <p>Default value: <b>1 (Middle)</b></p>
<p><b>Fit</b></p>	<p><a href="#">Boolean Value</a><sup>[24]</sup></p>	<p>Fits the pages of the overlay to match those of the printed file.</p> <p>Default value: <b>False</b></p>
<p><b>KeepAspect</b></p>	<p><a href="#">Boolean Value</a><sup>[24]</sup></p>	<p>Retains the proportions of the overlay file when using the options detailed above.</p> <p>Default value: <b>False</b></p>





## 2.2.1.2.10 Security



### Security

The **Security** properties determine document security options. Please note that if the **PDF/A** format is being used then security options will be ignored and therefore should be disabled in order to increase processing speed. The security options detailed below are available. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables/disables document security. If set to <b>False</b> then no further properties from this section are used.  Default value: <b>False</b>
<b>HasUserPassword</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables/disables the <b>User Password</b> .  If documents do not have user passwords then they can be opened at any time - even if an owner password is set. Enable both user and owner passwords to maximize security.  Default value: <b>False</b>
<b>HasOwnerPassword</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables/disables the <b>Owner Password</b> .  Default value: <b>False</b>
<b>UserPassword</b>	String	Defines the <b>User Password</b> .  This password is used for opening documents

		Default value: <b>empty string</b>
<b>OwnerPassword</b>	String	<p>Defines the <b>Owner password</b>. This password is required to change document permissions/security settings. If this value is empty, or the same as <b>User password</b>, then it will not be possible to change permissions/security settings.</p> <p>Default value: <b>empty string</b></p>
<b>Type</b>	<p><b>0 (40RSA)</b>  <b>1 (128RSA)</b>  <b>2 (128AES)</b>  <b>3 (256AES)</b>  <b>4 (256AES_R6)</b></p>	<p>Defines the security method used for encryption. This property specifies the subsection (<b>Perms40</b> or <b>Perms128</b>) used to define user permissions.</p> <p>Default value: <b>0 (40RSA)</b></p>
<b>Perms40.AllowPrinting</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Enables/disables printing permissions.</p> <p>Default value: <b>False</b></p>
<b>Perms40.AllowChanging</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Enables/disables editing permissions.</p> <p>Default value: <b>False</b></p>
<b>Perms40.AllowCopying</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Enables/disables copy permissions.</p> <p>Default value: <b>False</b></p>
<b>Perms40.AllowComments</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Enables/disables comment permissions.</p> <p>Default value: <b>False</b></p>
<b>Perms128.ContentAccess</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Enables/disables content permissions.</p> <p>Default value: <b>False</b></p>
<b>Perms128.AllowCopying</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Enables/disables copy permissions.</p> <p>Default value: <b>False</b></p>

<b>Perms128.Changes</b>	<b>0 - 4</b>	<p>Specifies the level of editing security:</p> <p><b>0</b> = No changes are permitted.</p> <p><b>1</b> = Document assembly is permitted.</p> <p><b>2</b> = Form-filling and document signing is permitted.</p> <p><b>3</b> = Comment authoring, form filling and signing of the document is permitted.</p> <p><b>4</b> = General editing, comments and form-field authoring are permitted.</p> <p>Default value: <b>0</b></p>
<b>Perms128.Printing</b>	<b>0 - not allowed 1 - low resolution 2 - fully allowed</b>	<p>Specifies printing permissions.</p> <p>Default value: <b>0</b></p>
<b>Perms128.EncryptMeta</b>	<a href="#">Boolean Value</a> <sup>24</sup>	<p>Enables/disables metadata encryption.</p> <p>Please note that this property is applicable to only PDF specifications 1.6 and higher.</p> <p>Default value: <b>True</b></p>

## 2.2.1.2.11 Links



### Links

The **Links** property determines the parameters for document links. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Detects valid URL and mail links. Links must occupy only a single line of the page. Embedded links are not recognized.</p> <p>Default value: <b>False</b></p>
<b>DetectFiles</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Detects links to files.</p> <p>Default value: <b>False</b></p>
<b>CheckExistance</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Checks the validity of identified links.</p> <p>Default value: <b>False</b></p>
<b>Type</b>	<p><b>0 (None)</b>  <b>1 (Underline)</b>  <b>2 (Rectangle)</b></p>	<p>Determines the border format of links.</p> <p>Default value: <b>0 (None)</b></p>
<b>Style</b>	<p><b>0 (Solid)</b>  <b>1 (Dashed)</b>  <b>2 (Dotted)</b></p>	<p>Determines the border style of links.</p> <p>Default value: <b>0 (Solid)</b></p>

<b>Color</b>	Integer	Determines the RGB color of the border used for links. Default value: <b>255</b> (red)
<b>Width</b>	All values from <b>5</b> - <b>50</b>	Determines the width of the border used for links. The units used are tenths of a millimeter. Default value: <b>10</b>

---

## 2.2.1.2.12 Bookmarks



### Bookmarks

The **Bookmarks** properties activate automatic bookmark tree creation for printed documents and define the parameters used in the process. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Enables the automatic creation of a bookmark tree within the document. If set to <b>False</b> then a bookmark tree is not created and no further properties from this section are used.</p> <p>Default value: <b>False</b></p>
<b>AllowMultiLine</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Activates the detection of multiline bookmark items.</p> <p>Default value: <b>True</b></p>
<b>DocTitleAsRoot</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>If set to <b>True</b> then the root element of the bookmarks tree will be used as the document title. All other top-level bookmarks will be children of this root.</p> <p>Default value: <b>False</b></p>
<b>MatchLevel</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>If set to <b>True</b> then non-top-level items will be added to the tree only if a previously-detected item of a higher level already exists.</p> <p>If set to <b>False</b> then all detected items will be added to the tree. Dummy parents will be added when no parent is detected at a higher level.</p>

		Default value: <b>False</b>
<b>DisplayMode</b>	<b>0 (AllOpened)</b> <b>1 (AllClosed)</b> <b>2 (ByItem)</b> <b>3 (OpenToLevel)</b>	<p>Defines which items of the bookmarks tree are expanded automatically when documents are opened.</p> <p><b>0 (AllOpened)</b> expands all items.</p> <p><b>1 (AllClosed)</b> expands no items.</p> <p><b>2 (ByItem)</b> expands items according to the <b>Template</b> definition (see below).</p> <p><b>3 (OpenToLevel)</b> expands items up to the level that <b>UpToLevel</b> specifies.</p> <p>Default value: <b>2 (ByItem)</b></p>
<b>UpToLevel</b>	Integer	<p>Defines the level of the bookmarks tree up to which items are opened when the <b>DisplayMode</b> property is set to <b>3 (OpenToLevel)</b>.</p> <p>Default value: <b>1</b></p>
<b>Template</b>	String	<p>Defines the template used for detection and creation of bookmark trees. This property is not controlled via <b>ActiveX</b> and should be an empty string, which means <i>current template</i>. The <a href="#">AddBookmarkItem</a><sup>[81]</sup> function can be used to create current templates for this property.</p> <p>Default value: <b>empty string</b></p>

## 2.2.1.2.13 Default File Append



### Default File Append

The **Default File Append** properties can be used to append/prepend printed files to existing documents. Please note that all properties are case-sensitive:

#### Prepend Files

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables the prepend feature. If set to <b>False</b> then the feature is disabled and no further properties from this section are used.  Default value: <b>False</b>
<b>FileName</b>	String	The file specified will be used to prepend printed documents.
<b>Password</b>	String	The password (where necessary) of the file used for the <b>File Name</b> property.

#### Append Files

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	



		<p>Enables the apend feature. If set to <b>False</b> then the feature is disabled and no further properties from this section are used.</p> <p>Default value: <b>False</b></p>
<b>FileName</b>	String	The file specified will be used to apend printed documents.
<b>Password</b>	String	The password (where necessary) of the file used for the <b>File Name</b> property.

---

## 2.2.1.2.14 HeadersFooters



### HeadersFooters

The **HeadersFooters** properties define the placement of headers and footers in printed documents. Headers and footers contain three sections: left, center and right. Each section specifies the text color and font in the associated area of printed documents. [Macros](#)<sup>[111]</sup> can be used as desired. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables headers and footers. If set to <b>False</b> then headers and footers are not created and no further properties from this section are used.  Default value: <b>False</b>
<b>FirstPageNumber</b>	Integer	Defines the page at which headers and footers start. The value of this property is used in the <a href="#">%[Page]</a> <sup>[112]</sup> macro.  Default value: <b>1</b>
<b>OffsetLeft</b> <b>OffsetTop</b> <b>OffsetRight</b> <b>OffsetBottom</b>	Integer	These properties define the location of headers and footers. The units used are tenths of a millimeter.  Default value: <b>127</b>

Headers and footers are defined with two sections: **Header** and **Footer**. These sections are identical and contain three identical subsections: **Left**, **Center** and **Right**. The following properties apply to all subsections:

		DEFINITION
--	--	------------

NAME	VALUES	
<b>Font</b>	String	Specifies the font used for headers and footers. Default value: <b>Arial</b>
<b>FontSize</b>	All values from <b>10 - 32768</b>	Specifies font size in tenths of a point. (1 point is 1/72 inch). Default value: <b>100</b>
<b>FontBold</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables/disables bold for the text used. Default value: <b>False</b>
<b>FontItalic</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables/disables italics for the text used. Default value: <b>False</b>
<b>Color</b>	<b>RGB COLORREF</b>	Defines the color of the text used. Default value: <b>0 (black)</b>
<b>Value</b>	String	Defines the text used. <a href="#">Macros</a> <sup>[117]</sup> can be used in this property as desired. Default value: <b>empty string</b>

## Example

The following example will place a red label with the current page number at the right-top corner of each page, with the font Verdana, in bold and at 12 pt size:

### (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
```

---

## 2.2.1.2.15 DocInfo (Document Information)



### DocInfo (Document Information)

The **DocInfo** properties enable the specification of document information fields and XMP metadata. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	If this property is <b>False</b> then no information fields are specified except the <b>Producer</b> field, which is always specified and contains the <b>PDF-XChange Drivers API</b> version information.  Default value: <b>False</b>
<b>AskForDocInfo</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	If this property is <b>True</b> then a dialog box will be displayed after documents are spooled. The dialog box can be used to enter document information fields.  Default value: <b>False</b>
<b>AddXMP</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	If this property is <b>True</b> then XMP metadata will be added to printed documents. Metadata includes values of information fields, date of document creation etc.  Please note that if the <b>PDF/A</b> format is used then XMP metadata is always added to documents.  Default value: <b>False</b>
<b>Title</b>	String	Specifies the document <b>Title</b> .

		Default value: <b>empty string</b>
<b>Subject</b>	String	Specifies the document <b>Subject</b> . Default value: <b>empty string</b>
<b>Author</b>	String	Specifies the document <b>Author</b> . Default value: <b>empty string</b>
<b>Keywords</b>	String	Specifies document <b>Keywords</b> . Commas must be used to delimiter keywords. Default value: <b>empty string</b>
<b>Creator</b>	String	Specifies the document <b>Creator</b> . Default value: <b>empty string</b>

---

## 2.2.1.2.16 Save



### Save

The **Save** properties determine the folder in which documents are stored, the name of documents and further options that relate to the saving of documents. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<p><b>SaveType</b></p>	<p>0 (<b>pdfSaver</b>)                      1 (<b>AppendToLast</b>)                      2 (<b>Save</b>)                      3 (<b>JustEmail</b>)</p>	<p>Specifies the action when documents are spooled:</p> <p><b>0 (pdfSaver)</b> specifies that <b>pdfSaver</b> (the document-creation application within the <b>PDF-XChange Drivers API</b>) displays the <b>Job Management</b> window, where the desired action can be specified.</p> <p><b>1 (AppendToLast)</b> appends new documents to the last job in the jobs list. <b>pdfSaver</b> can then be used to create a PDF file from the job.</p> <p><b>2 (Save)</b> generates and saves the PDF file from the printed document with the properties defined in this section.</p> <p><b>3 (JustEmail)</b> creates the PDF file and emails it using the properties defined in the <a href="#">EMail</a><sup>[67]</sup> section.</p> <p>Default value: <b>2 (Save)</b></p>
<p><b>ShowSaveDialog</b></p>	<p><a href="#">Boolean Value</a><sup>[24]</sup></p>	<p>If this property is <b>True</b> then <b>pdfSaver</b> will display the <b>Save As</b> dialog box in order to facilitate saving. If the <b>Save As</b> dialog box is cancelled then the PDF file creation will also be cancelled.</p> <p>Default value: <b>True</b></p>

<p><b>Path</b></p>	<p>String</p>	<p>Defines the path where documents are saved. If this property is an empty string then the <b>My Documents</b> folder will be used.</p> <p>Default value: <b>empty string</b></p>
<p><b>File</b></p>	<p>String</p>	<p>Defines file name of the new document. This name can contain <a href="#">macros</a><sup>[111]</sup> and a path. If this property contains a path then it will be used instead of the path specified in <b>Path</b> property.</p> <p>Default value: <b>%[DocName]</b></p>
<p><b>StripPath</b></p>	<p><a href="#">Boolean Value</a><sup>[24]</sup></p>	<p>Defines how the <a href="#">%[DocName]</a><sup>[111]</sup> macro value is calculated. If this property is <b>True</b> and the document name passed from the printing application contains the path then it will be stripped when the <b>%[DocName]</b> property is assigned.</p> <p>Default value: <b>False</b></p>
<p><b>WhenExists</b></p>	<p> <b>0 (ShowWarning)</b>  <b>1 (Overwrite)</b>  <b>2 (AutoNumber)</b>  <b>3 (Append)</b>  <b>4 (InsertBefore)</b> </p>	<p>Specifies the action taken when a file is saved and the name specified in the <b>Path</b> and/or <b>File</b> properties already exists.</p> <p><b>0 (ShowWarning)</b> launches a dialog box in which the user can select the action taken.</p> <p><b>1 (Overwrite)</b> overwrites the existing file.</p> <p><b>2 (AutoNumber)</b> uses the existing name and appends a numeric value to the new file. The <b>AutoNumber.Start</b> and <b>AutoNumber.NumDigits</b> properties (see below) can be used to format numbering.</p> <p><b>3 (Append)</b> adds the new file at the end of the existing file.</p> <p><b>4 (InsertBefore)</b> adds the new file as the beginning of the existing file.</p> <p>Please note that these options are not available when the <b>PDF/A</b> format is used.</p> <p>Default value: <b>0 (ShowWarning)</b></p>



<b>AutoNumber.Start</b>	<b>0 - 999999</b>	<p>Defines the start number when the <b>2 (AutoNumber)</b> property is used.</p> <p>Default value: <b>1</b></p>
<b>AutoNumber.Num Digits</b>	<b>1 - 6</b>	<p>Defines the number format when <b>2 (AutoNumber)</b> is used. The number of digits selected defines the minimum number of digits used. For example if <b>3</b> is selected and the <b>2 (AutoNumber)</b> property is set to <b>1</b> then numbering will start at 001.</p> <p>Default value: <b>3</b></p>
<b>RunApp</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Specifies if new documents are opened after they are printed.</p> <p>Default value: <b>True</b></p>
<b>RunCustom</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>Defines the application used to open new documents. If set to <b>True</b> then the application specified in the <b>AppName</b> property will be used. If set to <b>False</b> then the default PDF application will be used. This property is used only when <b>RunApp</b> is set to <b>True</b>.</p> <p>Default value: <b>False</b></p>
<b>AppName</b>	String	<p>This property is used only when <b>RunApp</b> and <b>RunCustom</b> are set to <b>True</b>. It designates the application used to open files. The value of this property is the fully-qualified path name of the application.</p> <p>Default value: <b>empty string</b></p>
<b>AppParams</b>	String	<p>This property defines the parameters passed to the application specified in the <b>AppName</b> property when PDF files are opened. The string <b>%f</b> represents the full path name of printed files.</p> <p>Default value: <b>%f</b></p>



## 2.2.1.2.17 EMail



### EMail

The **EMail** properties are used to determine the settings of the email feature, which emails documents automatically when they are printed. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Enabled</b>	<a href="#">Boolean Value</a> <sup>24</sup>	Enables the email feature. If set to <b>False</b> then no further properties from this section are used.  Default value: <b>False</b>
<b>Method</b>	<b>0 (LaunchClient)</b> <b>1 (SendViaClient)</b> <b>2 (SendViaSMTP)</b>	Specifies how documents are emailed.  <b>0 (LaunchClient)</b> launches the default email client. Users can then make changes as desired and send the email.  <b>1 (SendViaClient)</b> launches the default email client sends the email automatically.  <b>2 (SendViaSMTP)</b> uses the SMTP server specified in the <b>Server</b> property to send the email.  Default value: <b>0 (LaunchClient)</b>
<b>From</b>	String	Specifies the email account used to send emails.
<b>To, CC, Bcc</b>	String	Specifies email recipients. Semicolons must be used to delimiter recipients. The email address or the name of recipients can be used. If the name of recipients is used

		<p>(for example: John Doe) then the email address is taken from the email client's address book.</p> <p>It is also possible to enter a combination of the name and email address of recipients, for example: John Smith &lt;john@server.com&gt;.</p> <p>Default values: <b>empty string</b></p>
<b>Subject</b>	String	<p>Defines the email subject.</p> <p>Default value: <b>empty string</b></p>
<b>Content</b>	String	<p>Defines the email content.</p> <p>Default value: <b>empty string</b></p>
<b>ZIP</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	<p>If this property is set to <b>True</b> then PDF files are sent as ZIP archives. Otherwise files are sent in their existing format.</p> <p>Default value: <b>True</b></p>
<b>Server</b>	String	<p>Specifies the SMTP server used to send emails when the <b>Method</b> property is set to <b>SendViaSMTP</b>.</p> <p>The SMTP server must be defined using the <a href="#">AddSMTPServer</a><sup>[98]</sup> function.</p> <p>Default value: <b>empty string</b></p>

## 2.2.1.2.18 DigSig (Digital Signatures)



### DigSig (Digital Signatures)

The **DigSig** properties determine digital signature properties for documents and document signatures. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
Enabled	<a href="#">Boolean Value</a> <sup>24</sup>	Enables/disables digital signatures properties. If set to <b>False</b> then no further properties from this section are used.  Default value: <b>False</b>
Store	<b>0 (System)</b> <b>1 (File)</b>	Defines the certificate used.  If set to <b>0 (System)</b> then the default system certificate will be used.  If the value of this property is <b>1 (File)</b> then an external PFX file must be specified (see below).  Default value: <b>0 (System)</b>
Hash	Binary	Defines the hash value of the certificate used for signing the document.  If <b>Store</b> is set to <b>0 (System)</b> then the The <b>PDF-XChange Drivers API</b> will use the corresponding certificate on the local computer.  If <b>Store</b> is set to <b>1 (File)</b> then this property is not used.
PFXFile	String	

		Specifies the PFX file that contains the digital signature when <b>Store</b> is set to <b>1 (File)</b> . If <b>Store</b> is set to <b>0 (System)</b> then this property is not used.
<b>PFXPassword</b>	String	Specifies the password for the PFX file when <b>Store</b> is set to <b>1 (File)</b> . If <b>Store</b> is set to <b>0 (System)</b> then this property is not used.
<b>Reason Location Contact</b>	String	These properties represent additional information that can be included with digital signatures.  Default values: <b>empty string</b>
<b>Graphic</b>	<b>0 (None) 1 (Image) 2 (Name)</b>	Defines how the signature field graphic is displayed.  If set to <b>0 (None)</b> then this feature will not be used, and information about the digital signature will be placed in the digital signature field.  If set to <b>1 (Image)</b> then the <b>GraphicFileName</b> file is used (see below).  If set to <b>2 (Name)</b> then the signer's name will be used.  Default value: <b>2 (Name)</b>
<b>GraphicFileName</b>	String	The fully-qualified path name for the image used when the <b>Graphic</b> property is set to <b>1 (Image)</b> .
<b>ShowFlags</b>	Integer	A combination of flags that define the signature field appearance. See below for further information.
<b>Page</b>	Integer	The zero-based page number on which the digital signature is placed. A value of <b>-1</b> places the signature on the last page.  Default value: <b>-1</b>
<b>Left</b>	Integer	

		<p>Defines the signature position (relative to the top-left corner of the page). The units used are tenths of a millimeter.</p> <p>Default value: <b>254</b></p>
<b>Top</b>	Integer	<p>Defines the signature position (relative to the top-left corner of the page). The units used are tenths of a millimeter.</p> <p>Default value: <b>254</b></p>
<b>Width</b>	Integer	<p>Defines the width of the signature field. The units used are tenths of a millimeter.</p> <p>Default value: <b>508</b></p>
<b>Height</b>	Integer	<p>Defines the height of the signature field. The units used are tenths of a millimeter.</p> <p>Default value: <b>168</b></p>

## 2.2.1.2.18.1 Optimization



### Optimization

The **Optimization** properties determine the optimization of printed files. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Blending</b>	One of following integer values:  <b>0 - Normal</b> <b>1 - Multiply</b> <b>2 - Screen</b> <b>3 - Overlay</b> <b>4 - Darken</b> <b>5 - Lighten</b> <b>6 - ColorDodge</b> <b>7 - ColorBurn</b> <b>8 - HardLight</b> <b>9 - SoftLight</b> <b>10 - Difference</b> <b>11 - Exclusion</b>	Specifies the blending mode used for line art when printing from certain applications (for example <b>AutoCAD</b> when line overlay is enabled). The functions of blending modes are detailed below this table.  Default value: <b>4</b>
<b>DeTessellation</b>	<b>0 (None)</b> <b>1 (Simple)</b>	Enables/disables detessellation for sequences of triangles. Detessellation is useful when printing from <b>AutoCAD</b> and enables the creation of smaller, optimized PDF files.  Default value: <b>1 (Simple)</b>
<b>UseTransparency</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables transparency drawing. The <b>PDF-XChange Drivers API</b> can detect relevant drawing sequences and replace them with transparent versions. Please note that this property is not available in <b>PDF/A</b> format.



		Default value: <b>True</b>
<b>seMaskedImages</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables the use of masked images, which produces smaller files for some drawing sequences.  Default value: <b>True</b>
<b>GrayPaletteAsRGB</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables the storage of grayscale image palettes as RGB palettes. The same value is used for each element of the grayscale palette in the RGB palette.  Default value: <b>False</b>
<b>ImagesOptimization</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables the optimization of image sequences. This process requires multiple image manipulations but significantly reduces the file size and increases viewing quality.  Default value: <b>True</b>
<b>EnableSwapping</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	Enables the swapping of images to disk during image optimization. When this property is set to <b>False</b> the process of optimization is faster but requires more memory.  Default value: <b>True</b>
<b>SwapToDiskLimit</b>	<b>1-8</b>	Defines the image size (in megabytes) at which images are swapped to disk when <b>EnableSwapping</b> is set to <b>True</b> .  Default value: <b>1</b>

## Blending Modes

The blending modes available in the **PDF-XChange Drivers API** are as follows:

- **Normal** maintains the **Fill Color**.

- **Multiply** multiplies the **Fill Color** value with the color value of the underlying content. The resulting color will be at least as dark as one of the original values.
  - **Screen** multiplies the complements of the **Fill Color** value with the color value of underlying content and then complements the result. The resulting color will be at least as light as one of the original values.
  - **Overlay** either multiplies or screens the **Fill Color** value with the color value of underlying content, depending on the color value of the latter. The **Fill Color** will overlay the underlying content while preserving its highlights and shadows. The color of underlying content will be blended with the **Fill Color** to reflect its shade.
  - **Darken** selects the darker of the **Fill Color** value and the color value of underlying content.
  - **Lighten** selects the lighter of the **Fill Color** value and the color value of underlying content.
  - **Color Dodge** brightens the color of the underlying content to reflect the **Fill Color**. Please note that if black is used as the **Fill Color** then there will be no effect.
  - **Color Burn** darkens the color of the underlying content to reflect the **Fill Color**. Please note that if white is used as the **Fill Color** then there will be no effect.
  - **Hardlight** either multiplies or screens the **Fill Color** value with the color value of underlying content, depending on the **Fill Color** value. This creates the effect of shining a strong spotlight on the underlying color.
  - **Softlight** either darkens or lightens the colors, depending on the **Fill Color** value. This creates the effect of shining a diffused spotlight on the underlying color.
  - **Difference** subtracts the darker of the constituent colors from the lighter color.
  - **Exclusion** performs a less contrasted version of the **Difference** option.
-

## 2.2.1.2.19 Saver

---



### Saver

---

The **Saver** properties determine the **ShowProgress** property of the **PDFSaver** application. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Show Progress</b>	<a href="#">Boolean Value</a> <sup>[24]</sup>	If this property is set to <b>True</b> then the progress dialog box will be displayed when new documents are printed.  If this property is set to <b>False</b> then the progress dialog box will not be displayed.  Default value: <b>True</b>

---

## 2.2.1.2.20 RenderMode

---



### Rendermode

---

The **Rendermode** property determines the printing mode used to render documents. Please note that all properties are case-sensitive:

NAME	VALUES	DEFINITION
<b>Rendermode</b>	<b>Auto</b> <b>GDI</b> <b>XPS</b>	Determines the printing mode used.  <b>Auto</b> sets the mode automatically and depends on the printing application used.  <b>GDI</b> sets the printing mode to GDI.  <b>XPS</b> sets the printing mode to XPS.

---

## 2.2.2 Functions

---



### Functions

---

The **Functions** are used to perform **ActiveX** controls:

- The [SetRegInfo](#)<sup>[78]</sup> function sets registration information.
  - The [ResetDefaults](#)<sup>[79]</sup> function resets all properties to their default values.
  - The [ApplyOptions](#)<sup>[80]</sup> function applies all specified [Property Options](#)<sup>[23]</sup>.
  - The [AddBookmarkItem](#)<sup>[81]</sup> function adds a new item to the bookmarks tree template.
  - The [AddTextWatermark](#)<sup>[85]</sup> function adds text-based watermarks.
  - The [AddImageWatermark](#)<sup>[91]</sup> function adds image-based watermarks.
  - The [StorePrinterOptions](#)<sup>[94]</sup> function stores existing printer options in an external storage device. Please note that storage devices must support the **IStream** interface.
  - The [ReStorePrinterOptions](#)<sup>[95]</sup> function restores printer options saved with the [StorePrinterOptions](#)<sup>[94]</sup> property.
  - The [SetAsDefaultPrinter](#)<sup>[96]</sup> function sets the printer associated with the [IPXCPrinter Object](#)<sup>[16]</sup> as the default system printer.
  - The [RestoreDefaultPrinter](#)<sup>[97]</sup> function restores the default system printer stored during a call to the [SetAsDefaultPrinter](#)<sup>[96]</sup> function.
  - The [AddSMTPServer](#)<sup>[98]</sup> function adds a definition of an SMTP server, which can be used to email files.
-

### 2.2.2.1 SetRegInfo

---



## SetRegInfo

---

The **SetRegInfo** function is used to set printer registration information. It is intended for use when registration information is not passed automatically during printer creation. Please note that declarations are case-sensitive:

### 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)
```

---

### 2.2.2.2 ResetDefaults

---



## ResetDefaults

---

The **ResetDefaults** function is used to reset all properties to their default values, as defined in the [Properties](#) <sup>[20]</sup> section. Please note that declarations are case-sensitive:

### Declaration

#### C#

```
void ResetDefaults();
```

#### C++

```
HRESULT ResetDefaults();
```

#### VB

```
Sub ResetDefaults()
```

---

### 2.2.2.3 ApplyOptions

---



## ApplyOptions

---

The **ApplyOptions** function is used to apply all properties adjusted using the **Property Options**. It should be called before printing via the printer associated with the [IPXCPrinter](#)<sup>[16]</sup> object. Please note that declarations are case-sensitive:

### Declaration

#### C#

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

#### C++

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

#### VB

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

---



#### 2.2.2.4 AddBookmarkItem

---



### AddBookmarkItem

---

The **AddBookmarkItem** function is used to add new items to the bookmarks tree template. Values returned from this function can be used to call the same function again. Please note that declarations are case-sensitive:

#### Declaration

##### C#

```
int AddBookmarkItem(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 AddBookmarkItem([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 AddBookmarkItem(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
```

## Parameters

### *nParent*

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

### *dwDetectFlags*

This parameter is used to specify patterns of text. Text in printed documents that matches the specifications will be converted into bookmark items. Please note that the value of this parameter cannot be zero - at least one parameter must be used, and multiple values can be used as desired. The following values are available:

BIT	VALUE	DEFINITION
0	1	If this bit is selected then text of the font specified in <b>sDet_FontName</b> is used to create bookmark items.  <b>sDet_FontName</b> cannot be set to NULL or be an empty string when this flag is used.
1	2	If this bit is selected then text of the style specified in <b>nDet_FontStyle</b> is used to create bookmark items.
2	4	If this bit is selected then text of the size specified in the <b>nDet_Size</b> and <b>nDet_SizeDelta</b> parameters is used to create bookmark items. Text of the size <b>nDet_Size ± nDet_SizeDelta</b> will be used.
3	8	If this bit is selected then the <b>nDet_Textcolor</b> parameter is also used to detect bookmark items.

### *sDet\_FontName*

Specifies the font name used to determine bookmark items.

*nDet\_FontStyle*

Specifies the font style used to determine bookmark items. The following values are available and can be combined as desired:

- 0 - Normal Font**
- 1 - Bold**
- 2 - Italic**
- 3 - Bold and Italic**

*nDet\_Size*

Specifies the font size used to determine bookmark items. Font sizes must be specified in tenths of a point.

*nDet\_SizeDelta*

Specifies the font size delta used to determine bookmark items. The delta value must be specified in tenths of a point.

*nDet\_Textcolor*

Specifies the text color used to determine bookmark items.

*dwDisplayFlags*

Specifies how bookmark items are displayed. The following values are available and can be combined as desired:

BIT	VALUE	DEFINITION
0	1	If this bit is selected then bookmark items will be displayed in bold.
1	2	If this bit is selected then bookmark items will be displayed in italics.
2	4	If this bit is selected then bookmark items will be expanded by default. This flag is used only when <b>Bookmarks</b> property <a href="#">DisplayMode</a> <sup>[55]</sup> is set to the value <b>ByItem</b> .

*dwDisplaycolor*

Specifies the color of matching bookmark items as they appear in the bookmark tree. The **RGB Macro** defined in Windows API is used to specify the color.

## Return Value

This function returns an internal ID for the bookmark item. Internal IDs can be used as a **nParent** parameter for later calls to the [AddBookmarkItem](#)<sup>[81]</sup> function.

## Example

The following example creates a simple bookmarks detection tree determined according to the following parameters:

- 12pt, bold Times New Roman will be used to create top-level items.
- 12pt Times New Roman will be used to create second-level items.
- 14pt, bold Arial will be used to create top-level items and will be displayed in red.

## VB

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

---

### 2.2.2.5 AddTextWatermark

---



## AddTextWatermark

---

The **AddTextWatermark** function is used to define text-based watermarks. The parameters in this function correspond to the watermark parameters available in the **PDF-XChange Drivers API's GUI**. Please note that declarations are case-sensitive:

### 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)
```

## Parameters

### *sName*

Specifies the name of the new watermark. This name is used to identify the watermark in the [Watermarks](#)<sup>[45]</sup> property. If there is already a watermark with the name specified then it will be overwritten.

### *sText*

Specifies the watermark text.

### *sFontName*

Specifies the watermark font. Please note that this is a required parameter.

### *dwFontWeight*

Specifies the weight of the font used. The available range is 400 (normal) to 700 (bold).

*bItalic*

If this parameter is set to **True** then the italic variant of the font that **sFontName** identifies will be used.

*bOutline*

If this parameter is set to **True** then text is displayed as an outline. If it is set to **False** then text will be displayed as standard.

*nLineWidth*

Specifies the line width used when *bOutline* is set to **True**. Line width must be specified in tenths of points.

*nFontSize*

Specifies the font size in tenth of points. For example, 120 must be passed to generate 12pt text. If this parameter is set to 0 (zero) then the font size will be adjusted to fit the entire page.

*nTextcolor*

Specifies the text color used in the watermark.

*dwAlign*

Specifies the text alignment. If *nFontSize* is set to 0 (zero) then this parameter is ignored and watermarks are centered. The following parameters are available:

BIT	VALUE	DEFINITION
0	1	If this bit is used then text will align to the horizontal center.
1	2	If this bit is used then text will align on the right. Please note that if <b>BIT 0</b> is selected then this value cannot be used. Additionally, if neither <b>BIT 0</b> nor <b>BIT 1</b> is set then text will align to the left.
4	16	If this bit is used then text will align to the vertical center.
5	32	If this bit is used then text will be aligned to the bottom of the page. Please note that if <b>BIT 4</b> is selected then this value cannot be used. Additionally, if neither <b>BIT 4</b> nor <b>BIT 5</b> is used then text will align to the top of the page.

*xOffset*

Specifies the horizontal offset (relative to the top-left page corner) of the watermark. Values must be specified in tenths of millimeters.

*yOffset*

Specifies the vertical offset (relative to the top-left page corner) of the watermark. Values must be specified in tenths of millimeters.

*nAngle*

Specifies the watermark angle in degrees. All values from -90 to 90 are available. Positive values specify a counterclockwise rotation.

*dwOpacity*

Specifies the level of opacity used in the watermark. All values from 0 (fully transparent) to 100 (fully opaque) are available. If a value outside this range is used then the watermark will be clipped.

*dwFlags*

Specifies watermark placement and visibility. The following parameters are available and can be combined as desired:

BIT	VALUE	DEFINITION
0	1	This bit specifies that the parameter is not used.
1	2	This bit specifies that watermarks are placed as a background. If this bit is not used then watermarks will be placed in the foreground.
2	4	This bit specifies that watermarks are visible when viewed in PDF applications.
3	8	This bit specifies that watermarks are visible when documents are exported (for example as images) from the viewing application.
4	16	This bit specifies that watermarks are printed when documents are printed from the PDF application.





*dwPlaceType*

Specifies the pages on which the watermark is placed. The following parameters are available:

VALUE	DEFINITION
0	The watermark will be placed on all pages.
1	The watermark will be placed on only the first page.
2	The watermark will be placed on only the last page.
3	The watermark will be placed on only even pages.
4	The watermark will be placed on only odd pages.
5	The watermark will be placed on pages that the <i>sRange</i> parameter specifies. (See below).

*sRange*

This parameter is a string that defines the page range(s) onto which the watermark is placed.

Available page ranges are as follows:

- The page number, in order to specify a single page.
- A range of pages in the format [p1]-[p100], where the pages specified are included.
- A range of pages in the format [p1]- where all pages from the page specified are included.
- A range of pages in the format -[p100] where all pages from the beginning of the documented to the page specified are included.
- Please note that if multiple page ranges are specified then they must be delimited with either commas or semicolons.

For example the string: "**-3, 5, 7-11, 15-**" will place watermarks on pages 1, 2, 3, 5, 7, 8, 9, 10, 11 and from page 15 to the end of the document.



### 2.2.2.6 AddImageWatermark

---



## AddImageWatermark

---

The **AddImageWatermark** function is used to define image watermarks. The parameters of this function correspond to the parameters available in the **PDF-XChange Drivers API's UI** when new image watermarks are created. Please note that declarations are case-sensitive:

### 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)
```

## Parameters

### *sName*

Specifies the watermark name. This name can be used in the [Watermarks](#)<sup>[45]</sup> property. If there is already a watermark with the name specified then it will be overwritten.

### *sImageFileName*

Specifies the path to the image file. The following formats are supported: **BMP, PNG, JNG, JPEG, JPEG 2000, TIFF, JBIG, JBIG2, GIF, PCX, DCX, EMF** and **WMF**. If a multipage image is specified then the first page will be used.

### *dwTranscolor*

Specifies the transparent color for the image used. The pixels in the image of the color specified will be transparent in generated files. Specify **-1** to disable this feature.

### *dwWidth*

Specifies the width of the watermark. The image specified in **sImageFileName** will be resized to fit the area specified and will retain its aspect ratio.

### *dwHeight*

Specifies the height of the watermark. The image specified in **sImageFileName** will be resized to fit the area specified and will retain its aspect ratio.

All other parameters of this function have the same meaning as the corresponding parameters of the [AddTextWatermark](#)<sup>[85]</sup> function with the exception of the **dwFlags** parameter, which features two additional flags:

*dwFlags*

Specifies watermark placement and visibility. The following parameters are available and can be combined as desired:

BIT	VALUE	DEFINITION
8	256	If this bit is specified then the image will be rescaled to fit the page. The <i>dwWidth</i> and <i>dwHeight</i> parameters will be ignored.
9	512	If this bit is specified then the image used will not keep its aspect ration when scaling takes place.

**Return Values**

If the function fails then the return value is an error code.

If the function succeeds then the return value is `S_OK`.

---

### 2.2.2.7 StorePrinterOptions

---



## StorePrinterOptions

---

The **StorePrinterOptions** function is used to store printer options in an external location. Please note that storage locations must support the **IStream** interface. Please note that declarations are case-sensitive:

### Declaration

#### C#

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

#### C++

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

#### VB

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

### Parameters

*pStream*

Pointer to the **IStream** object into which printer properties are stored.

### Return Values

If the function fails then the return value is an error code.

If the function succeeds then the return value is `S_OK`.

---

### 2.2.2.8 ReStorePrinterOptions

---



## ReStorePrinterOptions

---

The **ReStorePrinterOptions** function is used to restore printer options that the [StorePrinterOptions](#)<sup>[94]</sup> function has been used to save. Please note that declarations are case-sensitive:

### Declaration

#### C#

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

#### C++

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

#### VB

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

### Return Values

If the function fails then the return value is an error code.

If the function succeeds then the return value is `S_OK`.

---

### 2.2.2.9 SetAsDefaultPrinter

---



## SetAsDefaultPrinter

---

The **SetAsDefaultPrinter** function is used to set the **IPXCPrinter** as the default printer. The [RestoreDefaultPrinter](#)<sup>[97]</sup> function can be used to restore previous default printers, which will restore automatically when the **IPXCPrinter** object is released. Please note that declarations are case-sensitive:

### Declaration

#### C#

```
void SetAsDefaultPrinter();
```

#### C++

```
HRESULT SetAsDefaultPrinter();
```

#### VB

```
Sub SetAsDefaultPrinter()
```

### Return Values

If the function fails then the return value is an error code.

If the function succeeds then the return value is S\_OK.

---



### 2.2.2.10 RestoreDefaultPrinter

---



## RestoreDefaultPrinter

---

The **RestoreDefaultPrinter** function is used to restore previous default printers that the **SetAsDefaultPrinter** function was used to override. If the **SetAsDefaultPrinter** function has not been previously called then this function will be ignored. Please note that declarations are case-sensitive:

### Declaration

#### C#

```
void RestoreDefaultPrinter();
```

#### C++

```
HRESULT RestoreDefaultPrinter();
```

#### VB

```
Sub RestoreDefaultPrinter()
```

### Return Values

If the function fails then the return value is an error code.

If the function succeeds then the return value is S\_OK.

---

### 2.2.2.11 AddSMTPServer

---



## AddSMTPServer

---

The **AddSMTPServer** function is used to add the parameters of a defined *SMTP* server, which can then be used to email files. See [Email](#)<sup>[67]</sup> for further information. Please note that declarations are case-sensitive:

### 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)
```

## Parameters

*sName*

Specifies the name/IP address of the SMTP server.

*nPort*

Specifies the port number used to communicate with the SMTP server.

*bNeedAuth*

If this parameter is set to **True** then it determines that the server requires authorization in order to send emails.

*sUser*

Specifies the username required for server authorization. If **bNeedAuth** is set to **False** then this parameter is not used.

*sPassword*

Specifies the password required to access the server. If **bNeedAuth** is set to **False** then this parameter is not used.

*bSecureConnection*

If this parameter is set to **True** then a secure connection must be used to access the server. If it is set to **False** then a secure connection is not required.

## Return Values

If the function fails then the return value is an error code.

If the function succeeds then the return value is `S_OK`.

---

### 2.2.3 Events

---



#### Events

---

The **IPXCPrinter** interface details events that developers can use to determine the stages of PDF creation within their applications. Events are grouped into two categories:

- **Asynchronous** events fire when documents are spooled. The **PDF-XChange Drivers API** does not wait for the host application to receive and return them. They are fired independently in order to prevent dead-locking and host applications receive them immediately.
- **Synchronous** events are fired when documents are generated. The **PDF-XChange Drivers API** waits for host applications to complete them before proceeding further.

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

The following events are supported in the **PDF-XChange Drivers API**:

#### Asynchronous Events:

- **Event OnStartDoc** is fired when printing starts for a new document.
- **Event OnStartPage** is fired when printing starts to for a new page.
- **Event OnEndPage** is fired when printing is complete for a page.
- **Event OnEndDoc** is fired when the GDI spools a complete document.

#### Synchronous Events:

- **Event OnDocSpooled** is fired immediately when the **PDF-XChange Drivers API** starts processing documents printed by the GDI.
- **Event OnFileSaved** is fired when documents are saved.
- **Event OnFileSent** is fired when automatic document emailing is enabled and the associated email has been sent. See [Email](#) [67] for further information.
- **Event OnError** is fired when an error occurs during document processing.



### 2.2.3.1 Asynchronous Events

---



## Asynchronous Events

---

The following asynchronous events are supported in the **PDF-XChange Drivers API**:

- **Event OnStartDoc** is fired when printing starts for a new document.
- **Event OnStartPage** is fired when printing starts to for a new page.
- **Event OnEndPage** is fired when printing is complete for a page.
- **Event OnEndDoc** is fired when the GDI spools a complete document.

Please note that declarations and parameters are case-sensitive:

**Event OnStartDoc** is fired when printing starts for a new document.

### 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)
```

### Parameters

#### *JobID*

Internal job identifier.

#### *lpszDocName*

Specifies the document name when passed from the printing application.

*lpszAppName*

Specifies the module name of the printing application.

### Return Values

The **PDF-XChange Drivers API** ignores return values from this event.

**Event OnStartPage** is fired when printing starts for a new page.

### Declaration

#### C#

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

#### C++

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

#### VB

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

### Parameters

*JobID*

Internal job identifier.

*nPageNumber*

Zero-based page number when spooling.

### Return Values

The **PDF-XChange Drivers API** ignores return values from this event.

**Event OnEndPage** is fired when printing is complete for a page.

### Declaration

#### C#

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

**C++**

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

**VB**

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

**Parameters***JobID*

Internal job identifier.

*nPageNumber*

Zero-based page number spooled.

**Return Values**

The **PDF-XChange Drivers API** ignores return values from this event.

**Event OnEndDoc** is fired when the GDI spools a complete document.

**Declaration****C#**

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

**C++**

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

**VB**

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

**Parameters***JobID*

Internal job identifier.

*bOK*



If this parameter is **True** then documents have fully spooled and the **PDF-XChange Drivers API** will continue processing and produce the files. If an error occurred during printing (such as the cancellation of the print job) then documents will not be processed further.

## Return Values

The **PDF-XChange Drivers API** ignores return values from this event.

---

### 2.2.3.2 Synchronous Events

---



## Synchronous Events

---

The following synchronous events are supported in the **PDF-XChange Drivers API**:

- **Event OnDocSpooled** is fired immediately when the **PDF-XChange Drivers API** starts processing documents printed by the GDI.
- **Event OnFileSaved** is fired when documents are saved.
- **Event OnFileSent** is fired when automatic document emailing is enabled and the associated email has been sent.
- **Event OnError** is fired when an error occurs during document processing.

**Event OnDocSpooled** is fired immediately when the **PDF-XChange Drivers API** starts processing documents printed by the GDI.

Please note that declarations and parameters are case-sensitive:

### 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)
```

### Parameters

#### *JobID*

Internal job identifier.

*lpzDocName*

Specifies the document name when passed from the printing application.

*lpzAppName*

Specifies the module name of the printing application.

Please note that applications can set document-specific properties when this event is handled. (For example the path where documents are stored, as defined by the printing application).

## Return Values

The application should return `S_OK` from this event handler.

**Event OnFileSaved** is fired when documents are saved. It enables the host application to perform application-based processing on documents as desired.

## Declaration

### C#

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

### C++

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

### VB

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

## Parameters

*JobID*

Internal job identifier.

*lpzFileName*

Specifies the fully qualified path name where document were saved.

## Return Values

The **PDF-XChange Drivers API** ignores values returned from this event.

**EventOnFileSent** is fired when automatic document emailing is enabled and the associated email has been sent.

## 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

Parameters for this event are the same as those of the **OnFileSaved** event detailed above.

## Return Values

The **PDF-XChange Drivers API** ignores values returned from this event.

**Event OnError** is fired when an error occurs during document processing. Additionally, it means that events **OnFileSaved** and/or **OnFileSent** will not be fired and that 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)
```

## Parameters

*JobID*

Internal job identifier.

*dwErrorCode*

Specifies an error (code) occurred during document processing.

## Return Values

The **PDF-XChange Drivers API** ignores values returned from this event.

---

## 2.2.4 Unsupported Property Declarations

---



## Unsupported Property Declarations

---

The following functions return an `E_NOTIMPL` error code:

- Event **LastPrinterEvent**
  - Function **PrintDocument**
  - Function **WaitForPrintEvent**
-

## 2.3 Macros



### Macros

The **PDF-XChange Drivers API** supports the use of **macros** to create name files and insert headers/footers. All macros use the following syntax:

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

Where **Macro** is the macro name and **extra** defines optional specifiers. Available macros are detailed in the table below. Please note that not all macros use specifiers:

NAME	DEFINITION
<b>AppName</b>	<p>Specifies the executable module of the printing application.</p> <p>Please note that this macro should contain only the file name of the executable and not the path of the module. For example if printing from <b>Microsoft Word</b> then this macro has the value <b>Winword.exe</b>.</p>
<b>DocName</b>	<p>Specifies the document name when the printing application passes it (via the <b>DOCINFO</b> structure at the start of the process) to the <b>PDF-XChange Drivers API</b>.</p> <p>Please note that if the <b>Save.StripPath</b> property is <b>True</b> and the document name contains a path then the path will be stripped before the macro is assigned.</p>
<b>DocPath</b>	<p>Specifies the paths of document names that the printing application passes.</p>
<b>Date</b>	<p>Specifies the document printing date. The <b>{:extra}</b> specifier can be used to format date values. If an <b>{:extra}</b> specifier is not given then the default</p>

	regional settings are used. A format string must be used for this macro - see below for string syntax.
<b>Year</b>	Specifies the year. Macro format depends on regional settings.
<b>Month</b>	Specifies the month. Macro format depends on regional settings.
<b>Day</b>	Specifies the day of the month. Macro format depends on regional settings.
<b>Time</b>	Specifies the time of printing as a string. The <b>{:extra}</b> specifier can be used to format date values. If an <b>{:extra}</b> specifier is not given then the default regional settings are used. A format string must be used for this macro - see below for string syntax.
<b>Hour</b>	Specifies the hour of printing. Macro format depends on regional settings.
<b>Minute</b>	Specifies the minute of printing. Macro format depends on regional settings.
<b>Second</b>	Specifies the seconds of printing. Macro format depends on regional settings.
<b>Computer</b>	The name of the local computer.
<b>User</b>	The login name of the current user.
<b>Page</b>	The current page number. This macro can be used for only headers and footers.  The <a href="#">HeadersFooters.FirstPageNumber</a> <sup>[58]</sup> property defines the starting page.
<b>Pages</b>	The total number of document pages.



<b>Env</b>	<p>The value of this macro is an environment variable specified with the <b>{:extra}</b> specifier. If the specifier is not present, or there is no associated environment variable, then the value of the macro will be an empty string.</p> <p>For example <b>%[Env:UserName]</b> specifies the login name of the user when printing takes place.</p>

---



**- A -**

AddBookmarkItem 81  
AddImageWatermark 91  
AddSMTPServer 98  
AddTextWatermark 85  
ApplyOptions 80  
Asynchronous Events 102

**- B -**

Bookmarks 54

**- C -**

Compression 35  
Custom Forms 31

**- D -**

Default File Append 56  
Digital Signatures 69  
DigSig 69  
DocInfo (Document Information) 61

**- E -**

Email 67  
Events 100

**- F -**

Fonts 43  
Functions 77

**- G -**

General 32  
Graphics 38

**- H -**

HeadersFooters 58

**- I -**

Installation 9  
Introduction 6  
IPXControlEx Object 16  
IPXCPrinter Interface 19

**- L -**

Links 52

**- M -**

Macros 111

**- O -**

Optimization 72  
Overlays 46

**- P -**

Paper 25  
Printing Process 15  
Properties 20  
Property Name 22  
Property Options 23  
Property Table Definition 24

**- R -**

ResetDefaults 79  
RestoreDefaultPrinter 97  
ReStorePrinterOptions 95

**- S -**

Save 63  
Saver 75  
Security 49  
SetAsDefaultPrinter 96  
SetRegInfo 78  
StorePrinterOptions 94  
Synchronous Events 106

System Requirements 14

## - T -

Troubleshooting 6

## - U -

Unsupported Property Declarations 110

## - W -

Watermarks 45

Welcome 5