PDF Xpansion SDK 13

Reference

soft Xpansion GmbH & Co. KG ● Königsallee 45, 44789 Bochum

+49 234 58801264 ● pdf@soft-xpansion.com ● www.soft-xpansion.com

TABLE OF CONTENTS

INTRODUCTION

This reference contains the documentation of the **PDF Xpansion SDK**: interfaces, properties and methods, structures and types declared and used in the PDF Xpansion libraries.

Certain properties and methods that may be discoverable through SDK's introspection facilities are not documented here. Undocumented properties and methods should not be used. They are entirely unsupported and subject to change without notice at any time.

This reference is intended for developers familiar with C++, C# or VB.NET. The intended audience includes, but is not limited to these languages. Familiarity with the PDF and/or XPS file formats and the object models can be helpful.

Different development platforms in this Reference have been designated as:

- **UWP** – applications for Universal Windows Platform
- **Windows** – applications and services written in C++ as the classic Windows x86/x64 programs
- **DotNET** – .NET Framework applications
- **COM** – applications written in Delphi, Builder and the languages which support OLE Automation

Store / Phone Store: For applications compatible with **Windows 8.1 / 8.0**
we provide PDF Xpansion SDK 11 / PDF Xpansion SDK 10. Please request us if required.

DotNET: PDF Xpansion SDK includes assemblies built in **Visual Studio 2017** for **.NET Framework 4.0 / 4.5 / 4.6 / 4.7**. If you need assemblies for other versions of .NET Framework, please request us.
Important! These assemblies require the redistributable files for **Visual Studio 2017** to be installed.

COM: All objects of this library implement IDispatch interface and can be used from many scripting languages: JScript, VBScript, VBA, etc.
The library provides the document viewer as an ActiveX control.

System Requirements

PDF Xpansion SDK can be used without any functional restrictions and third party components on following Windows versions:

- Windows 10 and Windows Server 2016
- Windows 8.1 and Windows Server 2012 R2*
- Windows 8 and Windows Server 2012*

PDF Xpansion SDK can be used with restrictions on following Windows versions:

- Windows 7 and Windows Server 2008 R2*

  For displaying and/or printing must be installed Platform update additionally

*UWP part of PDF Xpansion SDK cannot be used under these OS versions. For Windows Store / Windows Phone Store applications compatible with Windows 8.1 / 8.0 we provide PDF Xpansion SDK 11 / PDF Xpansion SDK 10. Please request us if required.

Installation of SDK

You may install and use PDF Xpansion SDK without setup - just download ZIP file and unpack the archive into any folder of your hard drive.

UWP: You need to install **pdf-xpansion-uwp10.vsix** extension package using VSIX Installer, see **DevRes** folder.

Contents of SDK package

Folder "Docs"

The folder contains SDK documentation.

Folder "DevRes"

UWP: The folder contains the Visual Studio Extension for the projects of UWP applications.

Subfolder "include"

Windows: The folder contains header files for classic C++ applications.
You don't need these files on the other platforms.

Subfolder "samples"

The folder contains multiple sample projects for different developer platforms. Please read the chapter "Redistribution of PDF Xpansion" before starting the compiled sample because you should perform this procedure for a sample application also as for any other.

Folder "Redist"

The folder contains redistributable files of SDK, more detailed information about using these files you find in chapter "Redistribution of PDF Xpansion".

UWP: You do not need the files from this folder on specified platforms because all necessary files are installed with appropriate Visual Studio Extension.

Reference SDK libraries in Your Projects

UWP:

1) Add new reference **PDF Xpansion SDK** to your project before you start to use the SDK API. You must install appropriate Visual Studio Extension package if you can't find this reference in the list of available references.
2) If you want use the document viewer, please add **PDFXpansionControl** to your app. To add a control in XAML of your app, double-click it in the Toolbox under **PDF Xpansion SDK** category.

Windows: Folder **DevRes\include** contains the header files for C++ development desktop or server applications using PDF Xpansion SDK. You should include these files to your project before you start to use the SDK API.

Important! Please define **SX_WINDESKTOP_DLL** macro in your project or in your source files if you want use the document viewer as a Windows window or other classic elements of Windows provided by PDF Xpansion SDK. Please define **SX_USE_OBJ_PTR** macro in your project or in your source files if you want use the ObjPtr template class.

DotNET:

1) Add new reference **pdf-xpansion-net40-x86.dll** from **redist** folder of PDF Xpansion SDK to your project before you start to use the SDK API.
2) Add new reference **pdf-xpansion-wpf40-x86.dll** from **redist** folder of PDF Xpansion SDK to your project before you start to use WPF control.

COM:
1) Register **pdf-xpansion-com-x86.dll** or **pdf-xpansion-com-x64.dll** using regsvr32 utility
2) Register **pdf-xpansion-ctl-x86.dll** or **pdf-xpansion-ctl-x64.dll** using regsvr32 utility if you need to use an ActiveX control
3) Embarcadero developers must import necessary components in the project – use "Component\Import Component" function of Embarcadero environment.

Replace Trial License

After you purchase the PDF Xpansion SDK you get your corporate license file **pdf-xpansion.license** and you will need to overwrite trial license file in order to replace trial license by your own. If you have tried the SDK in several projects, you must overwrite the license file in these projects also.

Note! Simultaneously with license file you should replace the trial license key in you sources because every license have own license key, you cannot authorize your license with trial key and vice-versa.

UWP: On this platform, you must overwrite a license file in location where SDK Extension was installed because extension package installs trial license. Usually you can find installed Visual Studio Extension files (and trial license file **pdf-xpansion.license**) in the subdirectories of directories **…\Microsoft SDKs\Windows Kits\10\ExtensionSDKs\pdf-xpansion-uwp**

Important! You should overwrite trial license every time after update of SDK files and reinstallation of Visual Studio Extension because extension package contains trial license always.

Update of SDK Files

We publish systematic releases of SDK with new functions and fixed bugs. We recommend that you download an actual version of SDK regularly (at least every quarter) and simply unpack the archive into the same folder (overwrite old SDK files).

Also you need to overwrite all used redistributable SDK files in your products with actual files, excluding

license file **pdf-xpansion.license** because SDK package contains trial license always.

UWP: You must reinstall used Visual Studio Extensions after every SDK update.
Simply open the actual VSIX file.
Important! You should replace trial license after this operation.

COM: You must repeat registration and import of components (see detailed instruction here) after this operation.

SDK Samples

Folder "DevRes\samples" contains multiple sample projects for different developer platforms. Please read the chapter "Redistribution of PDF Xpansion" before starting the compiled sample because you should perform this procedure for a sample application also as for any other.

Redistribution of PDF Xpansion

Files that need to be redistributed along with the client application that uses PDF Xpansion SDK are placed in the **Redist** folder of SDK. Some files have to be always redistributed, other only in some specific cases. You should copy redistributable files to one directory – application folder or subfolder. More detailed information about every file see in table below.

UWP: At this platform the necessary redistributable files will be automatically included in your app package. You do not need to redistribute any other SDK files.

DotNET: Please note that assemblies require require the redistributable files for **Visual Studio 2017** to be installed on the target computers.

Important! Only the .NET references will be copied automatically by Visual Studio to your application folder. All other required redistributable SDK files **you must copy yourself**.

COM: You must register all redistributed COM components on the client PCs before use.
Important! The component **pdf-xpansion-com-x86.dll** (or **pdf-xpansion-com-x64.dll**) can be used without registration if you can call C-compatible entry point within your projects (see detailed information here).

pdf-xpansion-windesktop-x86.dll pdf-xpansion-windesktop-x64.dll	The main library of SDK, it should be redistributed with all applications excluding UWP applications. If you build only x86 binaries in your project, you may redistribute x86 library only.
pdf-xpansion-net40-x86.dll pdf-xpansion-net40-x64.dll	The assembly for use in .NET projects. You should redistribute it with .NET applications only. You may place these assemblies in Global Assembly Cache (GAC) also.
pdf-xpansion-wpf40-x86.dll pdf-xpansion-wpf40-x64.dll	The assembly with the document viewer as a WPF Control, it has reference to pdf-xpansion-net40-xxx.dll and you should redistribute second assembly also. You can place these assemblies in Global Assembly Cache (GAC) also.
pdf-xpansion-com-x86.dll pdf-xpansion-com-x64.dll	The COM-module for Delphi, Builder and the languages that support OLE interfaces.
pdf-xpansion-ctl-x86.dll pdf-xpansion-ctl-x64.dll	The ActiveX for Delphi, Builder and other languages that support embedding of ActiveX.
pdf-xpansion.license	The license file, it should be redistributed with all applications excluding UWP applications. Note! After you have purchased SDK license you will get your corporate license file and you will need to replace this file.
pdf-xpansion.pds	The resources store for the SDK library, it should be redistributed all applications excluding UWP applications.
pdf-xpansion-cjk.pds	This store contains resources for support of CJK languages.
pdf-xpansion-en.pds	This store contains localizable standard stamps. If you need other languages, please contact us.

PDF XPANSION SDK

Getting Started

1. Please install PDF Xpansion SDK at your computer.
2. If you already purchased SDK license, please replace the trial license with your own, in other case you can test PDF Xpansion SDK with trial license. The trial license has not any functional restrictions, but applies DEMO watermark on the document pages. In addition, it has time restriction: you can use the trial license 3 months, after this time you need to download actual version of SDK and continue your tests.
3. Now you may try to compile our samples or you may try to integrate PDF Xpansion SDK in your project, in a second case you need to add the SDK references in project.
4. Please read the chapter **SDK Entry Point** before integrate PDF Xpansion SDK in your project. Also read the chapter **Using Document Viewer** if you want integrate document viewer in your application.
5. Please read the chapter **SX::IRefObject Interface / IDisposable Interface** (for C++ and .NET applications only) about controlling lifetime of SDK objects and management of used resources.
6. Please read the chapter **Error Handling / Exceptions** about processing of SDK errors in your projects.
7. Please read the chapter **Redistribution of PDF Xpansion** before starting the compiled sample or your application because you should perform this procedure for any application that uses PDF Xpansion SDK.

Load SDK Library

On most platforms the PDF Xpansion SDK library will be loaded to your application automatically and you can get entry point of SDK library anytime.

Windows: Before you can use the library in your application, you must load it from **pdf-xpansion-windesktop-x86.dll** or **pdf-xpansion-windesktop-x86.dll** module using **LoadLibrary** or **LoadLibraryEx** function. After successful loading of library, you can use module handle to get entry point of SDK library anytime. See our sample as an example.

COM: Optionally you can use the library without registering at the client PC. In this case, you need to load it as described above if your environment allows this operation.

SDK Entry Point

We recommend you to get entry point of SDK library during the initialize phase of your application. See our samples as an example.

Important! PDF Xpansion SDK need to be authorized before you can use it.

Please call **IApplication::Authorize** method during the initialize phase of your application.

UWP: You should use **EntryPoint** method of the SX.UWP.PDFXpansionSDK class to get SX.UWP.Application class.

Windows: At first, you should get address of **EntryPoint** function using **GetProcAddress** system function and handle of loaded library. Than you can cast a retrieved address to **SX::LibraryEntryPoint** and call **EntryPoint** function - as a result you get SX::IApplication interface.

DotNET: you should use **EntryPoint** method of the SX.NET.PDFXpansionSDK class to get SX.NET.IApplication interface.

COM: you should create an object of PDFXpansionSDK class (ID is **SX.PDFXpansionSDK.12**) and use **EntryPoint** method of this object to get IApplication interface.
Note! If you have not registered this library in registry but you can load it dynamically, you can get address of **EntryPoint** function using GetProcAddress system function and handle of loaded library. Call of EntryPoint function returns IApplication interface.

Using Document Viewer

The PDF Xpansion SDK provides several platform oriented implementations of document viewer. Using these implementations, you can easy and quickly integrate a power viewer (and/or annotator, editor) in your application.

Error Handling / Exceptions

PDF Xpansion SDK uses the mechanism of exceptions and throws the exception of specific type which represent errors that occur during application execution. Please catch and process these exceptions everywhere you use PDF Xpansion SDK.

UWP: SDK throws exceptions of the Platform.COMException type. The **HResult** property of an exception object corresponds to the ErrorCode enumeration.

Windows: SDK throws C++ exceptions of the **SX::IException** type. The **ErrorCode** property value of this interface corresponds to the ErrorCode enumeration.

DotNET: SDK throws exceptions of the **SX.NET.Exception** type which inherits System.Exception class. The **ErrorCode** property value of this interface corresponds to the ErrorCode enumeration.

COM: SDK do not throws exceptions on this platform because traditionally uses **HRESULT** return value. All methods on this platform return a value of the type **HRESULT** which corresponds to the ErrorCode enumeration. Please find more information about error handling in COM.

Combine, Convert, Import and Export Documents

The PDF Xpansion SDK provides a simple and powerful possibility to combine PDF and/or XPS documents. It combines not only the pages, but the hyperlinks, bookmarks and outline of document. Within the PDF format can be correctly combined the annotations of document, form fields and layers. Using this functionality one document can be split up into multiple documents (for example, one page – one separate document).

The library makes quick and high quality conversion from PDF to XPS or OXPS format and backward conversion. Using the library many other formats can be imported to the PDF:

- plain text can be formatted and layouted at the pages in PDF document
- raster or SVG image can be imported as new page or placed on the existing page
- GDI or GDI+ metafile can be imported as new page or placed on the existing page

Using the library the pages of PDF or XPS document can be rendered and exported to the raster images, also the plain text can be extracted from such documents.

The library makes conform PDF/A (versions 1, 2 and 3) and ZUGFeRD files from PDF files or images.

Create, Edit or Explore the Page Content

The PDF Xpansion SDK provides direct access to the page content in PDF and/or XPS documents. See details about the Rich Content API within SDK. The viewer provides interactive tool for creating or editing page content by end user.

Use Metadata (XMP) in PDF Files

You can get metadata of PDF document or edit such metadata using XMP API which is part of PDF Xpansion SDK.

ZUGFeRD Invoices

The PDF Xpansion SDK provides complete tool for processing ZUGFeRD invoices, see more detailed information here.

Document Viewer

The PDF Xpansion SDK provides several platform oriented implementations of document viewer. Using these implementations you can easy and quickly integrate power viewer and editor of **PDF**, **XPS**, **ZUGFeRD** and other documents in your application. Also you can display the documents of own format using customization possibilities of viewer.

The viewer implements a wide spectrum of traditional functions: different layouts, zooming, scrolling, etc. It has many tools for processing content: text marking, creating and editing of annotations (different types), page content editor.

The viewer uses modern and high-performance Windows API: **Direct2D** for displaying of documents. Direct2D takes advantage of hardware acceleration via the graphics processing unit and can minimize CPU usage and utilize hardware rendering on a graphics card.

The viewer processes all activity of end-user, including keyboard, mouse, touch and pen inputs.

Using power and effective event mechanism provided by the viewer, your application could extend standard functionality of viewer and implement the custom tools (see **ToolMode** property of the viewer) for visual processing of PDF or other documents.

Windows: On this platform you can create and use document viewer as a classic Windows window object represented by **SX::Viewer::WinViewer** interface. This viewer displays a loaded document and processes end-user activity.

As an alternative, you can create viewer object without window object. In this case you can use **SX::Viewer:IViewer** interface to manage viewport with layout of loaded document in your own window.

DotNET: On this platform you can use document viewer as an **Forms Control** or **WPF Control**. These controls are derived from the appropriate control objects within .NET Framework and you can integrate it in your applications in the traditional way.

UWP: On this platform you can use document viewer as a **custom control** that derived from the native user control object and you can integrate it in your applications in the traditional way.

COM: On this platform you can use document viewer as a classic **ActiveX** and you can integrate it in your applications in the traditional way.

Coordinate System and Unit of Measure

The viewer coordinate system is based on the coordinate system of the display devices. The basic unit of measure is the device unit (typically, the pixel). The logical size of device unit within a viewer you can control using configuration interface of the viewer, especially DPI property. The x-coordinates increase to the right; y-coordinates increase from top to bottom.

The viewer operates with three similar coordinate systems:

- coordinate system of viewport
- coordinate system of viewed document canvas
- coordinate system of document page placed on the canvas

The origin of these coordinate systems coincides with the upper left corner of the viewport, canvas and page area accordingly. The offset of the viewport coordinate system realtive to the canvas coordinate system is the scroll values and can be positive only. The placement of document page (page coordinate system) relative to the canvas coordinate system provides **IViewerCanvas::GetPageRect** method.

The pages in the PDF or XPS document have own native coordinate systems, these systems are different

and do not coincide with the unified page coordinate system in the viewer. You can transform the page coordinates between unified and native systems using **PDF::IPage::CalcMatrix** or **XPS::IPage::CalcMatrix**.

.NET Forms Control

This control is derived from System::Windows::Forms::Control.

Application Property

Type: IApplication. Access: readonly.

The root of the PDF Xpansion API.

Document Property

Type: IViewableDocument. Access: full.

The property represents a document displayed in the viewer. Please note that you may not attach one IViewableDocument object in two or more viewers simultaneously. If you want display one document in the several viewers, you must request the necessary number of viewable document objects (one for every viewer).

ToolMode Property

Type: tool_mode. Access: full.

The property represents actual tool mode of the viewer.

Important! Before you set the value of this property to something else than "tool_mode_none" you must define handler of viewer events and associate it with viewer.

Canvas Property

Type: IViewerCanvas. Access: readonly.

The property provides the canvas of the viewer. Using this interface you can control layouting of the document in the viewer and navigate over the document.

Config Property

Type: IViewerConfig. Access: readonly.

The property provides the configuration settings of the viewer.

EnableBarScroll Property

Type: sx_bool. Access: full.

The property enables processing of scrollbar events and scrolling of document in the viewer.

EnableKeysScroll Property

Type: sx_bool. Access: full.

The property enables processing of keyboard events and scrolling of document in the viewer.

EnableWheelScroll Property

Type: sx_bool. Access: full.

The property enables processing of mouse wheel events and scrolling of document in the viewer.

Relayout Method

The method recalculates layout of displayed document in the viewer if the document structure (number of pages, page size or page order) was changed.

NewActPage Parameter

Type: sx_uint.

The index of active page (see Canvas property above) after changing of document structure.

For example, before changes the active page was 7, than page 4 was deleted in document, after this change, you should call Realyout method and specify firts parameter as 6. In this case, the viewer recalculates the document layout but preserves (if possible) the active page and scrolling position.

If active page no more available, set this prameter to 0.

AdoptPageOffset Parameter

Type: sx_bool.

True if you want preserve visible part of active page after relayout.

Invalidate Method

The method invalidates the viewport image.

Pages Parameter

Type: sx_bool.

True if content of pages was changed and page images in the viewer cache must be deleted.

Pages Parameter

Type: sx_bool.

True if PDF annotations in the document was changed.

OnEvent Event

The viewer events.

Delegate: IViewerEvents.

.NET WPF Control

This control is derived from System.Windows.Controls::UserControl.

Application Property

Type: IApplication. Access: readonly.

The root of the PDF Xpansion API.

Document Property

Type: IViewableDocument. Access: full.

The property represents a document displayed in the viewer. Please note that you may not attach one IViewableDocument object in two or more viewers simultaneously. If you want display one document in the several viewers, you must request the necessary number of viewable document objects (one for every viewer).

ToolMode Property

Type: tool_mode. Access: full.

The property represents actual tool mode of the viewer.

Important! Before you set the value of this property to something else than "tool_mode_none" you must define handler of viewer events and associate it with viewer.

Canvas Property

Type: IViewerCanvas. Access: readonly.

The property provides the canvas of the viewer. Using this interface you can control layouting of the document in the viewer and navigate over the document.

Config Property

Type: IViewerConfig. Access: readonly.

The property provides the configuration settings of the viewer.

EnableBarScroll Property

Type: sx_bool. Access: full.

The property enables processing of scrollbar events and scrolling of document in the viewer.

EnableKeysScroll Property

Type: sx_bool. Access: full.

The property enables processing of keyboard events and scrolling of document in the viewer.

EnableWheelScroll Property

Type: sx_bool. Access: full.

The property enables processing of mouse wheel events and scrolling of document in the viewer.

Relayout Method

The method recalculates layout of displayed document in the viewer if the document structure (number of pages, page size or page order) was changed.

NewActPage Parameter

Type: sx_uint.

The index of active page (see Canvas property above) after changing of document structure.

For example, before changes the active page was 7, than page 4 was deleted in document, after this change, you should call Realyout method and specify firts parameter as 6. In this case, the viewer recalculates the document layout but preserves (if possible) the active page and scrolling position.

If active page no more available, set this prameter to 0.

AdoptPageOffset Parameter

Type: sx_bool.

True if you want preserve visible part of active page after relayout.

Invalidate Method

The method invalidates the viewport image.

Pages Parameter

Type: sx_bool.

True if content of pages was changed and page images in the viewer cache must be deleted.

Pages Parameter

Type: sx_bool.

True if PDF annotations in the document was changed.

OnEvent Event

The viewer events.

Delegate: IViewerEvents.

UWP Control

This control is derived from Windows.UI.Xaml.Controls::UserControl.

Application Property

Type: IApplication. Access: readonly.

The root of the PDF Xpansion API.

Document Property

Type: IViewableDocument. Access: full.

The property represents a document displayed in the viewer. Please note that you may not attach one IViewableDocument object in two or more viewers simultaneously. If you want display one document in the several viewers, you must request the necessary number of viewable document objects (one for every viewer).

ToolMode Property

Type: tool_mode. Access: full.

The property represents actual tool mode of the viewer.

Important! Before you set the value of this property to something else than "tool_mode_none" you must define handler of viewer events and associate it with viewer.

Canvas Property

Type: IViewerCanvas. Access: readonly.

The property provides the canvas of the viewer. Using this interface you can control layouting of the document in the viewer and navigate over the document.

Config Property

Type: IViewerConfig. Access: readonly.

The property provides the configuration settings of the viewer.

EnableBarScroll Property

Type: sx_bool. Access: full.

The property enables processing of scrollbar events and scrolling of document in the viewer.

EnableKeysScroll Property

Type: sx_bool. Access: full.

The property enables processing of keyboard events and scrolling of document in the viewer.

EnableWheelScroll Property

Type: sx_bool. Access: full.

The property enables processing of mouse wheel events and scrolling of document in the viewer.

EnableNavigationButtons Property

Type: sx_bool. Access: full.

The property enables the “page back” and “next page” buttons on the left and right edges of the viewer. The viewer hides the buttons automatically after three seconds of inactivity.

SnapToPageNavigationButtons Property

Type: sx_bool. Access: full.

If true, than navigation buttons (see above) navigate to the top edge of document page. If false, than navigation buttons navigate to the previous or next “screen page”.

EnableZoomButtons Property

Type: sx_bool. Access: full.

The property enables a zoom control in the bottom right hand corner of the viewer. The viewer hides the buttons automatically after three seconds of inactivity.

EnablePageIndicator Property

Type: sx_bool. Access: full.

The property enables the page info in the bottom left hand corner of the viewer. The viewer hides the buttons automatically after three seconds of inactivity.

EnableDrawProgress Property

Type: sx_bool. Access: full.

The property enables a draw progress in the the viewer until the redraw complete.

Suspended Property

Type: sx_bool. Access: full.

The property allow to break and block any interaction with document (f.e. long time redraw) if you need to access the document from other thread (f.e. from semantic zoom view). Set this property to false if you want continue displaying the document in the viewer.

Relayout Method

The method recalculates layout of displayed document in the viewer if the document structure (number of pages, page size or page order) was changed.

NewActPage Parameter

Type: sx_uint.

The index of active page (see Canvas property above) after changing of document structure.

For example, before changes the active page was 7, than page 4 was deleted in document, after this change, you should call Realyout method and specify firts parameter as 6. In this case, the viewer recalculates the document layout but preserves (if possible) the active page and scrolling position.

If active page no more available, set this prameter to 0.

AdoptPageOffset Parameter

Type: sx_bool.

True if you want preserve visible part of active page after relayout.

Invalidate Method

The method invalidates the viewport image.

Pages Parameter

Type: sx_bool.

True if content of pages was changed and page images in the viewer cache must be deleted.

Pages Parameter

Type: sx_bool.

True if PDF annotations in the document was changed.

OnEvent Event

The viewer events.

Delegate: IViewerEvents.

ActiveX Control

This is classic ActiveX control.

Application Property

Type: IApplication. Access: readonly.

The root of the PDF Xpansion API.

Viewer Property

Type: IViewer. Access: readonly.

The multifunctional viewer object.

SX Namespace

SX namespace contains declarations of base types, common enumerations, structures and interfaces used in the PDF Xpansion SDK.

UWP: For this platform, **SX.UWP** is a main namespace of the PDF Xpansion SDK, use it everywhere in your projects instead of documented namespace **SX**.
Note! Customers of PDF Xpansion SDK 11 / 10 need to rename used before **SX.RT** namespace to **SX.UWP** within all projects.

DotNET: For this platform, **SX.NET** is a main namespace of the PDF Xpansion SDK, use it everywhere in your projects instead of documented namespace **SX**.

COM: This platform does not use any namespaces, please ignore main namespace of the PDF Xpansion SDK on this platform.

IRefObject Interface / IDisposable Interface

Windows: This is fundamental interface at this platform, but it do not present used at other platforms. It implements object lifetime management through reference counting. Most interfaces of PDF Xpansion SDK are inherited, directly or indirectly, from this interface.
See description of **AddRef** and **Release** methods below for more detailed information.
We recommend you use the instances of an ObjPtr Class everywhere you use **IRefObject** based interfaces to save pointer in a safe manner.

DotNET: Many objects of PDF Xpansion SDK use unmanaged resources and allocate unmanaged memory. Therefore most interfaces at this platform are inherited, directly or indirectly, from **System.IDisposable** interface. Use possibilities of this interface to control the used / available resources.

AddRef Method

The method increments the reference count for an object. Call this method for every new copy of an interface pointer that you make. You do not need to call this method if you use instances of an **ObjPtr Class** everywhere you use **IRefObject** based interfaces.

Important! All methods or properties of PDF Xpansion SDK call **AddRef** on a pointer before return it,

so you must call Release method when you no longer need to use an interface pointer returned by PDF Xpansion SDK.

Important! Please, **do not call AddRef method** additionally before passing it as an in parameter to any method or by setting a property within PDF Xpansion SDK.

Release Method

Call this method when you no longer need to use an interface pointer returned by PDF Xpansion SDK. You do not need to call this method if you use instances of an **ObjPtr Class** everywhere you use **IRefObject** based interfaces.

ObjPtr Class (Smart Ptr)

Windows: This template class is used at this platform only. Using the instances of this class provide some advantages over using a raw pointer of **IRefObject** based interface:
- you can create instances tailored for a specific type of interface pointer
  (for any interface derived from **IRefObject**)
- call **AddRef** on the interface pointer received during an assignment operation
- provide different constructors to initialize a new instance through convenient mechanisms
- call **Release** for the encapsulated interface pointer when the class destructor executes

Note! Please define **SX_USE_OBJ_PTR** macro in your project or in your source files if you want use this class.

PDFXpansionSDK Class

The class provides a starting point of the PDF Xpansion SDK on the UWP, .NET and COM platforms. Under starting point we understand a static method **EntryPoint** on the UWP and .NET platforms. On the

COM: On this platform, you can create an object of this class (ID is **SX.PDFXpansionSDK.12**) and use **EntryPoint** method as a starting point.

Windows: Please use **EntryPoint** function as a starting point within all C++ projects.

EntryPoint Method

This method returns SX::IApplication interface.

DotNET:
If you place .NET assemblies of PDF Xpansion SDK in one directory with other redistibutable files (main libraries, license file, etc., than you may use **EntryPoint** method to get SX::IApplication interface.
If you place .NET assemblies of PDF Xpansion SDK in GAC, you must use **EntryPointEx** method instead of EntryPoint and specify a parameter - an absolute path to the main library, for example **C:\Program Files\Company\Product\sx\pdf-xpansion-windesktop-x86.dll**.
Both methods generate an exception ErrorPathNotFound, if assembly cannot load the main library.

COM:
If you place COM modules of PDF Xpansion SDK in one directory with other redistibutable files (main libraries, license file, etc., than you may use **EntryPoint** method to get SX::IApplication interface.
If you place COM modules of PDF Xpansion SDK in other place, you must use **EntryPointEx** method instead of EntryPoint and specify a parameter - an absolute path to the main library, for example **C:\Program Files\Company\Product\sx\pdf-xpansion-windesktop-x86.dll**.
Both methods generate an error ErrorPathNotFound, if assembly cannot load the main library.

IApplication Interface

At the root of the PDF Xpansion API is the **IApplication** interface.

Note! PDF Xpansion SDK need to be authorized before you can use it. Please call Authorize method.

Version Property

Type: sx_str. Access: readonly.

The property provides technical version of SDK.

Authorized Property

Type: sx_bool. Access: readonly.

The property provides state of SDK authorization. You can use authorized library only. See Authorize method for details.

Factory Property

Type: SX::IAppFactory. Access: readonly.

The property provides factory interface, you need it to instantiate any SDK objects.

Settings Property

Type: SX::IAppSettings. Access: readonly.

The property provides settings interface, you need it to set up SDK properties.

Authorize Method

The method authorizes the holder of a license to access the permitted part API of PDF XpansionSDK. It must be called at least once, and is usually called only once, for application process that uses the library. Multiple calls to this method are allowed but not recommended as long as they do nothing after first successful call.

The method will attempt to locate the license file by looking for a file named "pdf-xpansion.license" in the same directory as the main library of SDK or by path specified as a second parameter.

LicenseKey Parameter

Type: sx_str.

The license key, you get it together with your own license, but unlike the license file, you should use license key content in your sources to call this method only. Please, don't redistribute key file with you application. See our samples as an example (its use trial license key!).

LicenseFilePath Parameter

Type: sx_str.

If the redistributable license has a different filename and/or is not in the same directory as the main library of SDK, you should specify absolute path of license file here, otherwise this parameter must be null or empty string.

Errors

The method produces an exception

- ErrorPathNotFound if can't find the license file.
- ErrorAuthFailed if used license key doesn't match the license or license file is corrupted.
- ErrorAccessDenied if can't find **pdf-xpansion.pds** file.
- ErrorInvalidUse if can't find **pdf-xpansion-windesktop-xXX.dll** file.
- ErrorOutOfRange if trial license is expired, please download actual version of PDF Xpansion SDK.

IAppFactory Interface

The interface represents factory of SDK objects, it instantiates SDK objects permitted by license.

CreateSequentialStream Method

The method creates new instance of memory based data stream. It returns ISequentialStream interface.

CreateReadStream Method

The method creates new instance of file based, data stream (read access only). It returns ISequentialStream interface.

UWP / WinRT / WinPhone: The method isn't available on this platform, but you can use ReadStreamAsync / ReadFileAsync methods of memory based ISequentialStream implementation.

FilePath Parameter

Type: sx_str.

The file path (can be filename, relative or absolute path).

Errors

The method produces an exception ErrorPathNotFound if can't find or open the file.

CreateWriteStream Method

The method creates new instance of file based, data stream (write access only, create always). It returns ISequentialStream interface.

UWP / WinRT / WinPhone: The method isn't available on this platform, but you can use WriteStreamAsync / WriteFileAsync methods of memory based ISequentialStream implementation.

FilePath Parameter

Type: sx_str.

The file path (can be filename, relative or absolute path).

Errors

The method produces an exception ErrorPathNotFound if can't find or open the file.

CreateRefStream Method

The method creates new instance of wrapper for the platform specific data stream. It returns ISequentialStream interface. The wrapper object redirect all calls to the similar methods of base stream

object.

UWP / WinRT / WinPhone: The method isn't available on this platform, but you can use ReadStreamAsync / ReadFileAsync / WriteStreamAsync / WriteFileAsync methods of memory based ISequentialStream implementation.

Stream Parameter

Windows: The IStream interface.

DotNET: The System.IO.Stream class.

CreateCryptoCert Method

The method creates new instance of X.509 certificate. It returns ICryptoCert interface.

CreateCryptoKey Method

The method creates new instance of private key object. It returns ICryptoKey interface.

CreateRegion Method

The method creates new instance of graphic region object. It returns Graphics::IRegion interface.

Color Method

The method returns auxiliary Graphics::IColor interface.

Matrix Method

The method returns auxiliary Graphics::IMatrix interface.

CreateBitmapFromStream Method

The method creates new instance of bitmap object and loads bitmap data from the specified image stream

(file). The method returns Graphics::IBitmapData interface.

Stream Parameter

Type: ISequentialStream.

The image data stream.

CreateBitmapFromSource Method

The method creates new instance of bitmap object and loads bitmap data from IWICBitmapSource interface. The method returns Graphics::IBitmapData interface.

UWP / WinRT / WinPhone: The method isn't available on this platform.

DotNET: The method isn't available on this platform.

CreateDrawingContext Method

The method creates new instance of SDK drawing context object. The method returns Graphics::IDrawingContext interface.

UWP / WinRT / WinPhone: The method isn't available on this platform.

Drawer Parameter

Type: Graphics::Drawer.

The type of context.

CreatePDFDocument Method

The method creates new instance of PDF document object. The method returns PDF::IDocument interface.

Errors

The method produces an exception ErrorAuthFailed if your license does not permit use the PDF documents.

CreateXPSPackage Method

The method creates new instance of XPS document object. The method returns XPS::IPackage interface.

Errors

The method produces an exception ErrorAuthFailed if your license does not permit use the XPS documents.

CreateTextDocument Method

The method creates new instance of document object with plain text. The method returns ITextDocument interface.

CreateImageDocument Method

The method creates new instance of image document object. The method returns IImageDocument interface.

CreateFragmentDocument Method

The method creates new instance of document object which is wrapper to other document. The method returns IFragmentDocument interface.

CreateCompositeDocument Method

The method creates new instance of document object which is composition of several other documents. The method returns ICompositeDocument interface.

CreateAcquisitionDocument Method

The method creates new instance of document object which is content provider for the acquisition devices (scanners, web cams, etc.). The method returns IAcquisitionDocument interface.

CreateXMPDocument Method

The method creates new instance of XMP document which contains the metadata. The method returns XMP::IDocument interface.

CreateInvoiceDocument Method

The method creates new instance of ZUGFeRD document object. The method returns ZUGFeRD::IDocument interface.

CreateInvoiceStream Method

The method creates new instance of ZUGFeRD stream object. The method returns ZUGFeRD::IStream interface.

CreateViewer Method

The method creates new instance of viewer object. The method returns Viewer::IViewer interface.

Windows: We recommend you create window based viewer instead of this object.

UWP / WinRT / WinPhone: The method isn't available on this platform. Please read "Using Document Viewer" for detailed information about using viewer on this platform.

DotNET: The method isn't available on this platform. Please read "Using Document Viewer" for detailed information about using viewer on this platform.

Errors

The method produces an exception ErrorAuthFailed if your license does not permit use the document viewer.

CreateWinViewer Method

The method creates new instance of viewer window. The method returns Viewer::IWinViewer interface.

Windows: You must define **SX_WINDESKTOP_DLL** macro in your project if you need to use window based viewer.

UWP / WinRT / WinPhone: The method isn't available on this platform. Please read "Using Document Viewer" for detailed information about using viewer on this platform.

DotNET: The method isn't available on this platform. Please read "Using Document Viewer" for detailed information about using viewer on this platform.

Rc Parameter

Type: sx_rect.

The size and location of the window relative to the top-left corner of the parent window.

Parent Parameter

Type: sx_window (typedef of HWND).

A handle to the parent or owner window of the window being created.

Style Parameter

Type: sx_flags, see also Viewer::viewer_flags.

The optional styles of the viewer window.

IAppSettings Interface

The interface provides properties and options of PDF Xpansion SDK.

IBaseDocument Interface

The interfaces declares common functionality for all documents supported by SDK.

Windows: This interface is derived from IRefObject at this platform. We recommend you use an instance of an ObjPtr template class where you can use an IBaseDocument pointer in a safe manner.

GetViewableDocument Method

The method returns Viewer::IViewableDocument interface for using in the viewer. Please note that you may not attach one instance of this interface in two or more viewers simultaneously. If you want display one document in the several viewers, you must request the necessary number of instances (one for every viewer).

GetSearchProvider Method

The method provides search provider of the document. It returns ISearchProvider interface.

Errors

The method produces an exception ErrorAuthFailed if your license does not permit to search in the text of document.

GetPrintProvider Method

The method provides print provider of the document. It returns IPrintProvider interface.

UWP / WinRT / WinPhone: The method haven't any parameters on WinStoreplatform. The method is not available on WinPhone platform.

Note! On **Windows 7** and **Windows Server 2008 R2** must be installed **Platform Update** if you want print the documents using IPrintProvider interface.

JobName Parameter

Type: sx_str.

The document name (and printer job name also).

PrinterName Parameter

Type: sx_str.

The name of the printer, print server or printer object.

DevMode Parameter

Type: sx_str.

A pointer to a DEVMODE structure that the operating system uses to print the document, it is optional parameter, it can be null.

DotNET: On this platform the type of parameter is "array<unsigned char>^". Please copy unmanaged data to a managed memory block, size of block must be at least size of DEVMODE.

Output Parameter

Type: ISequentialStream.

The print output stream, it is optional parameter, it can be null. Using this parameter you can implement "print to file" functionality. The stream must be writable.

Errors

The method produces an exception ErrorAuthFailed if your license does not permit to print the documents.

Clear Method

The method completelly resets the document - all pages and other document ressources are removed.

ITextDocument Interface

The interfaces provides interface of plain text document object implemented by SDK.

This interface is derived from IBaseDocument that allow you to load text document in the document viewer

and/or combine it with PDF / XPS documents.

You can create new instance of document object using IAppFactory interface.

IImageDocument Interface

The interfaces provides interface of image document object implemented by SDK.

This interface is derived from IBaseDocument that allow you to load image document in the document viewer and/or combine it with PDF / XPS documents.

You can create new instance of document object using IAppFactory interface.

IFragmentDocument Interface

The interfaces provides interface of wrapper document object implemented by SDK. The wrapper references to other document, but you can define restricted page range or redefine page order of original document.

This interface is derived from IBaseDocument that allow you to load warpper document in the document viewer and/or combine it with PDF / XPS documents.

You can create new instance of document object using IAppFactory interface.

ICompositeDocument Interface

The interfaces provides interface of composite document object implemented by SDK. The document represents several documents as one document. The child documents can be of different types: text, image, PDF, XPS, etc.

This interface is derived from IBaseDocument that allow you to load composite document in the document viewer and/or combine it with PDF / XPS documents.

You can create new instance of document object using IAppFactory interface.

IAcquisitionDocument Interface

The interfaces provides interface of document object implemented by SDK. The document is content

provider for the acquisition devices (scanners, web cams, etc.). You can compose new document from paper sources or from web camera using this interface.

This interface is derived from ICompositeDocument that allow you to load obtained document in the document viewer and/or combine it with PDF / XPS documents.

You can create new instance of document object using IAppFactory interface.

ISearchProvider Interface

The interface provides possibility search text in the document.

Text Property

Type: sx_str. Access: readonly.

The property retrieves actual text pattern.

Page Property

Type: sx_uint. Access: readonly.

The property retrieves page index of last occurrence. It can be 0xFFFFFFFF if pattern wasn't found.

Selection Property

Type: Graphics::IRegion. Access: readonly.

The property retrieves location of last occurrence at the page. The coordinates of region are in the coordinate system of page.

Reset Method

The method resets the search parameters. It doesn't search first occurrence, you must call SearchForward or SearchBack for start.

Text Parameter

Type: sx_str.

The text pattern to find.

Page Parameter

Type: sx_uint.

The index of start page.

searchFlags Parameter

Type: sx_flags, see also search_opt enumeration.

The parameter specifies the options of search. You can combine the enumeration values by using the bitwise OR operator.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SearchForward Method

The method starts or continues search in forward direction. It returns true if text pattern was found or false if search reaches end of document. See Text, Page and Selection properties if next occurrence was found.

StatusHandler Parameter

Type: IStatusHandler.

The optional parameter sets a callback handler of search process. The IStatusHandler interface must implemented by application to get status events of search.

UWP / WinRT / WinPhone: At these platforms the IStatusHandler delegate is declared instead of callback interface. Please define an event handler method with the same signature as the IStatusHandler delegate and add an instance to the **OnPage** event of the search provider.

SearchBack Method

The method starts or continues search in backward direction. It returns true if text pattern was found or false if search reaches begin of document. See Text, Page and Selection properties if next occurrence was found.

StatusHandler Parameter

Type: IStatusHandler.

The optional parameter sets a callback handler of search process. The IStatusHandler interface must implemented by application to get status events of search.

UWP / WinRT / WinPhone: At these platforms the IStatusHandler delegate is declared instead of callback interface. Please define an event handler method with the same signature as the IStatusHandler delegate and add an instance to the **OnPage** event of the search provider.

GetContext Method

The method retrieves text context of current occurrence. It returns IStr.

MaxLen Parameter

Type: sx_uint.

Max length of context string.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IPrintProvider Interface

The interface provides print functionality of the document.

PageSet Property

Type: print_page_set. Access: full.

The property defines whether to print all pages of the document

PageRange Property

Type: IStr. Access: full.

The property specifies the range of pages to print in the document (PageSet property must be print_page_set_range). Separate numbers in a range by using a hyphen, and separate multiple pages or ranges by using commas.

PageScale Property

Type: print_page_scale. Access: full.

The property defines scale mode: reduce, enlarge, or crop pages while printing.

ReverseOrder Property

Type: sx_bool. Access: full.

The property enables printing pages in reverse order. If page ranges was specified, the pages print opposite of the order in which they were entered.

AutoRotate Property

Type: sx_bool. Access: full.

The property adjusts the page orientation to match the paper (or printable area) orientation.

Centered Property

Type: sx_bool. Access: full.

The property specifies the alignment of the page with the center of the printable area.

PrintAs Property

Type: print_content_as. Access: full.

The property specifies how to print every page.

DrawOptions Property

Type: sx_flags, see also draw_opt enumeration.

The property specifies technical options of rendering.

PDFAnnots Property

Type: sx_flags, see also PDF::annot_type enumeration.

The property specifies the types of PDF annotations to be printed. You can combine the enumeration values by using the bitwise OR operator.

StartPrint Method

The method prints specified pages of document to the selected printer.

WinRT: At this platform method is not available.

StatusHandler Parameter

Type: IPrintStatus.

The optional parameter sets a callback handler of search process. The IStatusHandler interface must implemented by application to get status events of printing.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetSheetsCount Method

The method retrieves number of sheets to be printed.

WinRT: At this platform method is not available.

GetSheetThumbnail Method

The method retrieves thumbnail of specified sheet for preview. It returns Graphics::IBitmapData interface.

WinRT: At this platform method is not available.

Sheet Parameter

Type: sx_uint.

The index of sheet.

Width Parameter

Type: sx_uint.

Thumbnail max width in pixels.

Height Parameter

Type: sx_uint.

Thumbnail max height in pixels.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetPrintDocumentSource Method

The method retrieves document source for a print task (see PrintTaskRequest.CreatePrintTask). It returns Windows::Graphics::Printing::IPrintDocumentSource interface.

WinRT: The method is available at this platform only.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

InvalidatePreview Method

The method forces the invalidation of print preview.

WinRT: The method is available at this platform only.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

ISequentialStream Interface

The interface supports simplified sequential access to stream data. The stream can be memory or file based. Also it could be a wrapper to other stream object.

Windows: This interface is derived from IRefObject at this platform.

GetLength Method

The method return actual length in bytes of the stream.

Reset Method

The method sets the stream pointer to the beginning of the stream data.

Read Method

The method reads a sequence of bytes from the stream to the buffer and advances the pointer position within the stream by the number of bytes read. It returns number of bytes copied to the buffer.

UWP / WinRT / WinPhone: On this platform you can use the methods **ReadBytes** (synchronous read operation), **ReadStreamAsync** and **ReadFileAsync** (asynchronous read operation).

Buf Parameter

Type: sx_byte*.

The pointer to the memory buffer.

DotNET: On this platform the type of parameter is "array<unsigned char>^".

Size Parameter

Type: sx_uint.

The maximum number of bytes to be read from the stream.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Write Method

The method writes a sequence of bytes from the buffer to the stream and advances the pointer position within the stream by the number of bytes written. It returns number of bytes copied to the stream.

UWP / WinRT / WinPhone: On this platform you can use the methods **WriteBytes** (synchronous write operation), **WriteStreamAsync** and **WriteFileAsync** (asynchronous write operation).

Buf Parameter

Type: sx_byte*.

The pointer to the memory buffer.

DotNET: On this platform the type of parameter is "array<unsigned char>^".

Size Parameter

Type: sx_uint.

The number of bytes must be writed to the stream.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

ICryptoCert Interface

The interface supports simplified sequential access to stream data.

Windows: This interface is derived from IRefObject at this platform.

ICryptoKey Interface

The interface supports simplified sequential access to stream data.

Windows: This interface is derived from IRefObject at this platform.

Base Types

PDF Xpansion SDK supports the type system that's defined by providing typedefs for the platform specific fundamental types:

SX	Windows	WinRT	DotNET
sx_bool	bool	bool	System::Boolean
sx_byte	unsigned char	Byte	System::Byte
sx_int	int	int32	System::Int32
sx_uint	unsigned int	uint32	System::UInt32
sx_short	unsigned short	uint16	System::Int16
sx_qword	unsigned __int64	uint64	System::UInt64
sx_float	float	Float32	System::Single
sx_double	double	float64	System::Double
sx_str (IStr)	const wchar_t*	Platform::String^	System::String^
sx_flags	sx_uint		

IStr Interface

Windows: This interface is derived from IRefObject at this platform. The SDK API returns any dynamic strings as **IStr** interface, please do not forget to release dynamic strings when you no longer need to use it (see IRefObject description).

str Property

Type: sx_str. Access: readonly.

The property retrieves the string content.

Enumerations

ErrorCode Enumeration

The members of this enumeration are the possible error codes for exceptions generated by SDK libraries.

ErrorPathNotFound

The library cannot find the path specified.

ErrorAccessDenied

Access is denied.

ErrorOutOfMemory

Not enough memory is available to process this action.

ErrorOutOfSpace

The drive is full.

ErrorLibrary

Common library error.

ErrorSystem

Unknown system error.

ErrorAuthFailed

The library does not authorized or function does not permitted.

ErrorOutOfRange

Invalid index.

ErrorPermDenied

The action does not permitted by document permissions.

ErrorInvalidFormat

Invalid data format.

ErrorNotImplemented

The function does not implemented yet.

ErrorInvalidParam

Invalid parameter value.

ErrorInvalidUse

Invalid use of the function.

ErrorDocSecurity

Inavlid use of encrypted document.

ErrorCryptoEngine

Error occurred during crypto operation.

combine enumeration

The members of this enumeration are the options for combine document operation. You can combine the enumeration values by using the bitwise OR operator.

combine_named_dests

Copy all used named destinations to the target document.

combine_outlines

Copy outline structure to the target document.

combine_form_fields

Copy form fields to the target document.

combine_layers

Copy layer objects to the target document.

combine_links

Copy links to the target document.

combine_emb_unk_fonts

Embed all non embedded and non Windows standard fonts.

combine_emb_win_fonts

Embed all non embedded Windows standard fonts.

Structures

sx_point

Contains a set of two floating-point numbers that represent the point in a two-dimensional plane.

DotNET: The System::Drawing::PointF class used on this platform.

x

The x-coordinate of the point.

y

The y-coordinate of the point.

sx_size

Contains a set of two floating-point numbers that specify a height and width.

DotNET: The System::Drawing::SizeF class used on this platform.

width

The width component of the structure.

height

The height component of the structure.

sx_rect

Contains a set of four floating-point numbers that represent the location and size of a rectangle.

DotNET: The System::Drawing::RectangleF class used on this platform.

x

The x-coordinate location of the left side of the rectangle.

y

The y-coordinate location of the top side (or bottom side in PDF namespace, see details here) of the rectangle.

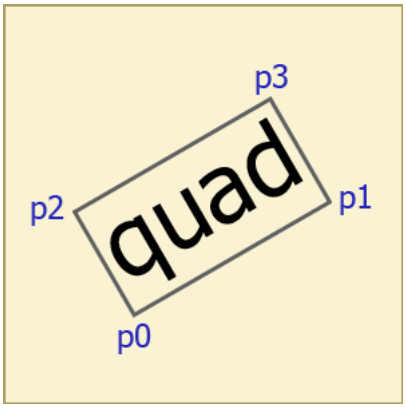
width

The width of the rectangle.

height

The height of the rectangle.

sx_quad



Contains a set of four sx_point structures that represent the parallelogram.

p0, p1, p2, p3

The coordinates of four corners.

sx_matrix

Contains a 3-by-2 matrix.

m11

The value in the first row and first column of the matrix.

m12

The value in the first row and second column of the matrix.

m21

The value in the second row and first column of the matrix.

m22

The value in the second row and second column of the matrix.

dx

The value of offset along the x-axis.

dy

The value of offset along the y-axis.

sx_time

Contains the local date and time (for your time zone). The current time-zone settings used by SDK you can control over IAppSettings::LocalTimeOffset property.

UWP / WinRT / WinPhone: The Windows::Foundation::DateTime class used on this platform.

DotNET: The System::DateTime class used on this platform.

wYear

The year. The valid values for this member are 1601 through 30827.

wMonth

The month. This member can be one of the following values.

wDayOfWeek

The day of the week. This member can be one of the following values.

wDay

The day of the month. The valid values for this member are 1 through 31.

wHour

The hour. The valid values for this member are 0 through 23.

wMinute

The minute. The valid values for this member are 0 through 59.

wSecond

The second. The valid values for this member are 0 through 59.

wMilliseconds

The millisecond. The valid values for this member are 0 through 999.

sx_color

It is base structure to define colors in different color spaces and formats. Base structure can be used to set color property of object in "none" value. If you need define color within grayscale, RGB or CMYK color space, you need to use derived structures:

- **sx_color_i_g** – grayscale color space, component values are integer in range 0..255
- **sx_color_i_rgb** – RGB color space, component values are integer in range 0..255
- **sx_color_i_rgba** – RGBA color space, component values are integer in range 0..255
- **sx_color_i_cmyk** – CMYK color space, component values are integer in range 0..255
- **sx_color_f_g** – grayscale color space, component values are float in range 0.0-1.0
- **sx_color_f_rgb** – RGB color space, component values are float in range 0.0-1.0
- **sx_color_f_rgba** – RGBA color space, component values are float in range 0.0-1.0

- **sx_color_f_cmyk** – CMYK color space, component values are float in range 0.0-1.0

Type Property

The property retrieves color space and format of color components. It is readonly property.

SX::Graphics Namespace

IRegion Interface

The interface describes an area (region) at the page. Region is a collection of quads. Regions are usually used in selection and hit-testing operations.

Windows: This interface is derived from IRefObject at this platform.

IBitmapData Data

The interface represents a raster image (bitmap). You can load this object from file or stream (see IAppFactory::CreateBitmapFromStream method) or you can get bitmap object from some methods of SDK interfaces, for example IPage::CopyToBitmap.

Windows: This interface is derived from IRefObject at this platform.

Width Property

Type: sx_uint. Access: readonly.

The property retrieves bitmap width, in pixels.

Height Property

Type: sx_uint. Access: readonly.

The property retrieves bitmap height, in pixels.

DPI Property

Type: sx_double. Access: full.

The property retrieves DPI of bitmap. When DPI is undefined, the value can be 0.

PixelFormat Property

Type:PixelFormat. Access: readonly.

The property retrieves pixel format of bitmap (color space and bit depth). Use CopyAs method to convert bitmap in specified format.

Palette Property

Type: IPaletteData. Access: readonly.

The property retrieves palette data for indexed formats of bitmap (see PixelFormat property).

GetPixels Method

The method retrieves the address of bitmap pixel data in the memory. The stride of lines padded out to a byte. The method can return null if data is too big for placing in memory, use GetScanLine method in this case.

DotNET: The method retrieves address of unmanaged memory on this platform.

UWP / WinRT / WinPhone: The method isn't available on this platform.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetScanLine Method

The method retrieves the address of pixel data for specified scan line of bitmap.

DotNET: The method retrieves address of unmanaged memory on this platform.

Line Parameter

Type: sx_uint.

The index of a scan line. Can be greater than or equal to 0 and less than bitmap height.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

CopyAs Method

The method creates new bitmap object and converts bitmap pixel data to specified format if it necessary. It returns IBitmapData interface.

Format Parameter

Type:PixelFormat.

The preferred pixel format of new bitmap.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SaveToStream Method

The method saves bitmap object to specified stream.

Stream Parameter

Type: ISequentialStream.

The stream for save. Please reset stream before save.

Format Parameter

Type: ImageFormat.

The preferred file format of saved image (PNG, JPEG, etc.).

Quality Parameter

Type: sx_uint.

The compression level when you save a JPEG image. Can be greater than 0 (bad quality / good compression) and less than or equal to 100 (good quality / bad compression).

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SX::Viewer Namespace

Viewer namespace is sub-namespace of main SDK namespace, it contains declarations of enumerations, structures and interfaces used in the document viewer.

IWinViewer Interface

This interface is available for applications written in C++ as the classic Windows x86- and/or x64 programs. It represents a normal Windows window which contains a viewer object (represented as IViewer interface). You can create new instance of viewer window using IAppFactory:: CreateWinViewer method. The viewer window you can place in the window hierarchy of your application.

Viewer Property

Type: IViewer. Access: readonly.

The multifunctional viewer object.

Window Property

Type: sx_window (typedef of HWND). Access: readonly.

The window handle for a viewer window.

Clipboard Property

Type: ISystemClipboard. Access: readonly.

The property provides an interface for interaction with system clipboard.

EnableBarScroll Property

Type: sx_bool. Access: full.

The property enables processing of scrollbar events and scrolling of document in the viewer.

EnableKeysScroll Property

Type: sx_bool. Access: full.

The property enables processing of keyboard events and scrolling of document in the viewer.

EnableWheelScroll Property

Type: sx_bool. Access: full.

The property enables processing of mouse wheel events and scrolling of document in the viewer.

SetEventHandler Method

The method sets a callback handler of viewer events.

EventHandler Parameter

Type: IViewerEvents.

The callback interface of event handler implemented by application.

EnableZoomGesture Property

Type: sx_bool. Access: full.

The property enables processing of touchscreen (if present) events for zooming of viewer content.

EnableScrollGesture Property

Type: sx_bool. Access: full.

The property enables processing of touchscreen (if present) events for scrolling of viewer content.

EnableRotateGesture Property

Type: sx_bool. Access: full.

The property enables processing of touchscreen (if present) events for the page rotation in viewer.

IViewer Interface

Windows: the interface used on this platform only. It represents main properties and functionality of viewer. We highly recommend you create the window based viewer where you can get IViewer interface using Viewer property of IWinViewer interface. As an alternative for professionals, you can create viewer object without Windows window (using IAppFactory::CreateViewer method) and use this object for embedding in your own window.

Document Property

Type: IViewableDocument. Access: full.

The property specifies a document displayed in the viewer. Please note that you may not attach one IViewableDocument object in two or more viewers simultaneously. If you want display one document in the several viewers, you must request the necessary number of IViewableDocument objects (one for every viewer).

Navigate Method

The method is an alternative for Document property. Using this method you can load new document and navigate to a specified page.

Doc Parameter

Type: IViewableDocument.

The document to be loaded.

Page Parameter

Type: sx_uint.

The index of page in the document. The viewer navigates to this page immediately.

TakeSnapshot Parameter

Type: sx_bool.

If this parameter is true, the viewer makes "snapshot" of viewer state (displayed document, layout and scroll properties of viewer) and places it in the viewer history. Using history functionality you can go back to this document.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Navigate Method

The method allows to restore previously fixed state of viewer (displayed document, layout and scroll properties of viewer) from screenshot object, see **TakeSnapshot** method also.

Snapshot Parameter

Type: IViewerSnapshot.

The snapshot object to be loaded.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

TakeSnapshot Method

The method makes "snapshot" of viewer state (displayed document, layout and scroll properties of viewer). It returns the IViewerSnapshot object.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Width Property

Type: sx_uint. Access: readonly.

The property provides viewport width in viewport units.

Height Property

Type: sx_uint. Access: readonly.

The property provides viewport height, in viewport units.

Resize Method

The method changes viewport size of the viewer.

Please **don't call this method directly** if you use a window based viewer.

Width Parameter

Type: sx_uint.

The new viewport width.

Height Parameter

Type: sx_uint.

The new viewport height.

Redraw Method

The method redraws actual viewport image on the specified drawing context. It returns false if viewport was completelly drawed or true if drawing isn't complete and you must call this method until its completed.

Please **don't call this method directly** if you use a window based viewer.

Ctx Parameter

Type:Graphics::IDrawingContext.

The target drawing context.

FirstBlock Parameter

Type: sx_bool.

True if you want begin to draw viewport image or false if you want continue uncomplete drawing (see above description of return value).

DrawRect Parameter

Type: sx_rect.

This optional parameter is for internal use. Must be a null.

Relayout Method

The method recalculates layout of displayed document in the viewer if the document structure (number of pages, page size or page order) was changed.

NewActPage Parameter

Type: sx_uint.

The index of active page (see Canvas property above) after changing of document structure.

For example, before changes the active page was 7, than page 4 was deleted in document, after this change, you should call Realyout method and specify firts parameter as 6. In this case, the viewer recalculates the document layout but preserves (if possible) the active page and scrolling position.

If active page no more available, set this prameter to 0.

AdoptPageOffset Parameter

Type: sx_bool.

True if you want preserve visible part of active page after relayout.

Invalidate Method

The method invalidates the viewport image.

Pages Parameter

Type: sx_bool.

True if page order or page content was changed and page cache in the viewer must be cleared.

PDFAnnots Parameter

Type: sx_bool.

True if PDF annotations in the document was changed and viewport cache must be updated.

Canvas Property

Type: IViewerCanvas. Access: readonly.

The property provides the canvas of the viewer. Using this interface you can control layouting of the document in the viewer and navigate over the document.

Config Property

Type: IViewerConfig. Access: readonly.

The property provides the configuration settings of the viewer.

ToolMode Property

Type: tool_mode. Access: full.

The property represents actual tool mode of the viewer.

Important! Before you set the value of this property to something else than "tool_mode_none" you must define handler of viewer events and associate it with viewer.

Focus Method

The method informs the viewer when window receives or loses the input focus.

Please **don't call this method directly** if you use a window based viewer.

Set Parameter

Type: sx_bool.

True if window receives the focus.

MouseActivity Method

The method informs the viewer about any mouse or touch events.

Please **don't call this method directly** if you use a window based viewer.

X Parameter

Type: sx_uint.

The x-coordinate of the cursor. The coordinate is relative to the upper-left corner of the client area.

Y Parameter

Type: sx_uint.

The x-coordinate of the cursor. The coordinate is relative to the upper-left corner of the client area.

Flags Parameter

Type: sx_flags, see also mouse_flags enumeration.

The type of mouse event and optional flags.

KeyboardActivity Method

The method informs the viewer about any keyboard events.

Please **don't call this method directly** if you use a window based viewer.

K Parameter

Type: sx_byte.

The virtual-key code of the key.

Flags Parameter

Type: sx_flags, see also key_flags enumeration.

The type of keyboard event and optional flags.

ShowPopup Method

The method shows or hides popup annotation within viewport. It is allowed if ToolMode property is tool_mode_none, tool_mode_tracker, tool_mode_select_text or tool_mode_annot. The popup mode in viewer configuration must be enabled also.

Page Parameter

Type: sx_uint.

The index of page where popup annotation is placed.

Popup Parameter

Type: PDF::IPopup.

The popup annotation object which must be shown or hidden.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SetEventHandler Method

The method sets a callback handler of viewer events.

Important! Please use the same method from IWinViewer interface instead this method if you use a window based viewer.

EventHandler Parameter

Type: IViewerEvents.

The callback interface of event handler implemented by application.

IViewerCanvas

This interface represents viewed canvas - a document layouted in the viewer according to viewer settings.

Also it provides basic viewer functions: navigation, scrolling, content marking, etc.

ActivePage Property

Type: sx_uint. Access: readonly.

The property retrieves the index of active page - it is partially or completelly visible page.

Columns Property

Type: sx_uint. Access: readonly.

The property retrieves the actual number of columns in layout.

Fit Property

Type: viewer_fit. Access: full.

The property specifies how a viewer will automatically zoom the pages to fit the size of window.

Height Property

Type: sx_uint. Access: readonly.

The property retrieves height of of the canvas, in viewport units.

HorizontalOffset Property

Type: sx_uint. Access: full.

The property specifies horizontal offset of the viewport in the canvas coordinate system, in viewport units.

LayoutMultiplicity Property

Type: sx_uint. Access: full.

The property specifies how many page columns (vertical layout) or page rows (horizontal layout) must have the layouted canvas.

MarkingType Property

Type: marking_type. Access: readonly.

The property retrieves the type of marked object. This property is used with **tool_mode_none** and **tool_mode_select_text** modes only.

MarkingRegion Property

Type: Graphics::IRegion. Access: readonly.

The property retrieves the region where marked object is located. The coordinates are in coordinate system of page in the viewer.

MarkedPage Property

Type: sx_uint. Access: readonly.

The property retrieves the index of the page marked object is located.

MarkingText Property

Type: IStr. Access: readonly.

The property retrieves the marked text if MarkingType is **marking_type_text**.

Rotation Property

Type: sx_rotation_angle. Access: full.

The property specifies the orientation (rotation angle) of the pages.

Rows Property

Type: sx_uint. Access: readonly.

The property retrieves the actual number of rows in layout.

Thumbs Property

Type: IViewerThumbs. Access: readonly.

The property retrieves extension of canvas interface for thumbnail mode of viewer (**viewer_mode_thumbs** value of IViewerConfig::Mode).

SeparateCover Property

Type: sx_bool. Access: full.

The property specifies that first page must be single in the first row of multicolumn layout.

VerticalLayout Property

Type: sx_bool. Access: full.

The property specifies the vertical or horizontal direction of layouting.

VerticalOffset Property

Type: sx_uint. Access: full.

The property specifies vertical offset of the viewport in the canvas coordinate system, in viewport units.

VisiblePages Property

Type: sx_uint. Access: readonly.

The property retrieves the number visible (partially or completelly) pages. See also the **GetVisiblePage** method.

Width Property

Type: sx_uint. Access: readonly.

The property retrieves width of the canvas, in viewport units.

Zoom Property

Type: sx_double. Access: full.

The property specifies the magnification scale of the pages in the viewer.

CanvasToPage Method

The method transform coordinates from canvas coordinate system to coordinate system of viewed page. For transformation in the reverse order, use **PageToCanvas** method. See also **GetPageMatrix** method.

Index Parameter

Type: sx_uint.

The index of a page, you can use **GetPageIndex** to get it.

PtCanvas Parameter

Type: sx_point.

The coordinates on the canvas (input parameter).

PtPage Parameter

Type: sx_point.

The coordinates on the specified page (output parameter).

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

ClearHistory Method

The method cleares history of the viewer operations (scrolling, navigation, relayout, etc). See also **History**

method.

GetPage Method

The method retrieves a page index for the specified column and row indexes.

Row Parameter

Type: sx_uint.

The row index in the page layout. It is zero-based.

Col Parameter

Type: sx_uint.

The column index in the page layout. It is zero-based.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetPageColumn Method

The method retrieves a column index of the page in the layout.

Index Parameter

Type: sx_uint.

The page index.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetPageIndex Method

The method retrieves a page located by specified coordinates on the canvas. It returns a page index or -1, if no page there.

X Parameter

Type: sx_uint.

The x-coordinate in the canvas coordinate system.

Y Parameter

Type: sx_uint.

The y-coordinate in the canvas coordinate system.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetPageMatrix Method

The method retrieves a matrix for transformation of coordinates from coordinate system of viewed page to canvas coordinate system. Method returns false if the viewer is in Flip mode and page located out of flip view canvas. See also **PageToCanvas** and **CanvasToPage** methods, its use the same matrix for transformation.

Index Parameter

Type: sx_uint.

The index of a page.

M Parameter

Type: sx_matrix.

The matrix for transformation (output parameter).

GetPageRect Method

The method provides rectange area of a specified page on the canvas. Method returns false if the viewer is in Flip mode and page located out of flip view canvas.

Index Parameter

Type: sx_uint.

The index of a page.

Rect Parameter

Type: sx_rect.

The page rectangle (output parameter), coordinates are in canvas coordinate system.

GetPageRow Method

The method retrieves a row index of the page in the layout.

Index Parameter

Type: sx_uint.

The page index.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetVisiblePages Method

The method retrieves the index of specified visible page. It returns the index of page in the document.

Index Parameter

Type: sx_uint.

The index of page in the list of visible pages. Must be smaller then **VisiblePages** property.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

History Method

The method checks history list or navigates in the history of viewer. See also **ClearHistory** method.

Back Parameter

Type: sx_bool.

The backward direction in the history.

Go Parameter

Type: sx_bool.

If this parameter is false, method returns availability of steps in the history in specified direction. If parameter is true and at least one step is available, the viewer navigates one step in specified direction.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Navigate Method

The method navigates to the specified location on the specified page.

Index Parameter

Type: sx_uint.

The page index.

X Parameter

Type: sx_uint.

The x-coordinate on the page (in viewer page coordinate system).

Y Parameter

Type: sx_uint.

The y-coordinate on the page (in viewer page coordinate system).

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Navigate Method

The method navigates to the specified ractangle area on the specified page.

Index Parameter

Type: sx_uint.

The page index.

Rect Parameter

Type: sx_rect.

The rectangle area on the page (coordinates are in viewer page coordinate system).

EnsureVisible Parameter

Type: sx_bool.

If parameter is false, method navigates always, if parameter is true and rectangle area is visible now the method makes nothing.

ZoomByRect Parameter

Type: sx_bool.

If parameter is true the viewer changes zoom to fit rectagle area.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

PageToCanvas Method

The method transform coordinates from coordinate system of viewed page to canvas coordinate system. For transformation in the reverse order, use **CanvasToPage** method. See also **GetPageMatrix** method.

Index Parameter

Type: sx_uint.

The index of a page.

PtPage Parameter

Type: sx_point.

The coordinates on the specified page (input parameter).

PtCanvas Parameter

Type: sx_point.

The coordinates on the canvas (output parameter).

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

ResetMarking Method

The method sets ot cleares marking in the viewer.

Marking Parameter

Type: Graphics::IRegion.

The region of marking. Can be null.

Index Parameter

Type: sx_uint.

The index of a page where the marking located.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SnapZoom Method

The method changes zoom discretely. It returns true if zoom was changed.

Index Parameter

Type: sx_bool.

Decrease or increase the zoom.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IViewerConfig

This interface provides viewer configuration options and settings.

IViewableDocument

The interface represents viewable document. For the documents implemented in the PDF Xpansion SDK, you can get this interface using GetViewableDocument method of IBaseDocument interface. Please note that you may not attach one instance of this interface in two or more viewers simultaneously. If you want display one document in the several viewers, you must request the necessary number of instances (one for every viewer).

Windows: This interface is derived from IRefObject at this platform. We recommend you use an instance of an ObjPtr template class where you can use an IBaseDocument pointer in a safe manner.
Important!
- You may not use this interface for drawing of pages directly in your application, please use appropriate methods of native page interfaces.
- You may implement this interface for your own document format, than you can load you documents in the viewer. The actual viewer uses Direct2D technology for rendering of content, see DrawPage method for detailed information.

GetUPI Method

The method retrieves “units per inch” property of the document. Is a measure of units in the document coordinate system, in particular the number of units that can be placed within the span of 1 inch.

GetPageCount Method

The method retrieves number of pages in the document.

GetPageSize Method

The method retrieves size of the requested page, the size is in document units.

DrawPage Method

Windows: The method declared on this platform only, you must implement rendering of specified page with specified transformation to the provided graphic context.

The actual viewer implementation uses Direct2D technology for rendering of content, therefore this method must draw content to the ID2D1RenderTarget context.

IViewerEvents Interface / Delegate

Using power and effective event mechanism provided by the viewer, your application can extend standard functionality of viewer and perform the appropriate application logic for visual processing of PDF or other documents.

Many useful tools of the viewer (see **ToolMode** viewer property) can be used together with event mechanism only, because the application must control these tools using own event handler. Please separate viewer tool events using IViewerEvent::**Type** property value **viewer_event_tool_activity**.

Windows: At this platform you must implement IViewerEvents interface in your application, if you want to process different viewer events and perform your tasks. Call **SetEventHandler** method of IWinViewer or IViewer interfaces to specify an instance of your IViewerEvents implementation as an event handler.

DotNET: At this platform the IViewerEvents delegate is declared instead of callback interface. Please define an event handler method with the same signature as the IViewerEvents delegate and add an instance to the **OnEvent** event of Forms or WPF control.

UWP / WinRT / WinPhone: At these platforms the IViewerEvents delegate is declared instead of callback interface. Please define an event handler method with the same signature as the IViewerEvents delegate and add an instance to the **OnEvent** event of WinStore or WinPhone control.

OnEvent Method / Event

The viewer calls this application defined handler to inform it about different events occurred in the document viewer or produced by end user activity. The handler must return false value.

Event Parameter

Type: IViewerEvent.

The event specific data.

IViewerEvent Interface

This interface describes event raised by document viewer and provides event specific data.

Type Property

Type: viewer_event. Access: readonly.

The property provides the type of event. Using this property you can type cast the **IViewerEvent** interface to the derived interfaces:

- IViewerEvent_PDF (for viewer_event_pdf)
- IViewerEvent_XPS_Link (for viewer_event_xps_link)
- IViewerEvent_PopupActivity (for viewer_event_popup)
- IViewerEvent_SetActivePage (for viewer_event_set_page)
- IViewerEvent_ScrollCanvas (for viewer_event_scroll_canvas)
- IViewerEvent_MouseActivity (for viewer_event_mouse_activity)
- IViewerEvent_KeyboardActivity (for viewer_event_keyboard_activity)
- IViewerEvent_ToolActivity (for viewer_event_tool_activity)

Handled Property

Type: sx_bool.

The property indicating whether the event was handled. True to bypass the viewer's default handling; otherwise, false to also pass the event along to the default viewer handler.

IViewerEvent_PDF Interface

This interface provides the events described by PDF ISO standard and raised by document viewer. See also derived interface IViewerEvent_PDF_Annot for the PDF annotation events.

PDFType Property

Type: PDF::pdf_event. Access: readonly.

The property retrieves type of PDF event.

IViewerEvent_PDF_Annot Interface

This interface provides the annotation events described by PDF ISO standard and raised by document

viewer. It is derived from IViewerEvent_PDF.

Annot Property

Type: PDF::IAnnot. Access: readonly.

The property retrieves PDF annotation object.

IViewerEvent_XPS_Link Interface

This interface provides the navigation event for XPS documents.

Activity Property

Type: hotspot_flags. Access: readonly.

The property retrieves the type of end-user activity relative to hyperlink area (pointer status: enter / down / up / exit).

Link Property

Type: XPS::ILink. Access: readonly.

The property retrieves hyperlink object.

IViewerEvent_PopupActivity Interface

This interface provides the popup events in the viewer.

Activity Property

Type: popup_activity. Access: readonly.

The property retrieves the operation with popup:

- popup_activity_visibility - show / hide popup
- popup_activity_activate - popup gets input focus
- popup_activity_rect - popup area on the page was changed
- popup_activity_text - popup text was changed

Visible Property

Type: sx_bool. Access: readonly.

The visibilty status of popup.

Active Property

Type: sx_bool. Access: readonly.

The popup has input focus now.

Popup Property

Type: PDF::IPopup. Access: readonly.

The reference to PDF popup annotation. Can be null (for non-PDF popups).

GetViewportRect Method

The method retrieves the popup area relative to the viewport coordinate system.

Rect Parameter

Type: sx_rect.

The popup rectangle area.

IViewerCancelableEvent Interface

The interface provides base functionality for all cancelable event types. It is derived from IViewerEvent.

Canceled Property

Type: sx_bool. Access: readonly.

The property value indicating whether the event should be canceled.

Cancel Method

The method declares that the event should be canceled.

IViewerEvent_SetActivePage

The interface provides event which informs about intention to change active page. It is derived from IViewerCancelableEvent.

Because this is cancelable event, you can restrict changing of active page.

Page Property

Type: sx_uint. Access: readonly.

The property value retrieves an index of new page.

IViewerEvent_ScrollCanvas Interface

The interface provides event which informs about intention to scroll a document canvas relative to the viewport. It is derived from IViewerCancelableEvent.

Because this is cancelable event, you can restrict scrolling.

VerticalOffset Property

Type: sx_uint. Access: readonly.

The property value retrieves new vertical offset.

HorizontalOffset Property

Type: sx_uint. Access: readonly.

The property value retrieves new horizontal offset.

IViewerEvent_MouseActivity Interface

The interface provides the mouse, touch, stylus input interactions. It is derived from IViewerCancelableEvent. Because this is cancelable event, you can skip an input action.

Activity Property

Type: mouse_flags. Access: readonly.

The property retrieves the type of input interaction

X Property

Type: sx_uint. Access: readonly.

The property value retrieves x-coodinate of pointer in the viewport coordinate system.

Y Property

Type: sx_uint. Access: readonly.

The property value retrieves y-coodinate of pointer in the viewport coordinate system.

Delta Property

Type: sx_int. Access: readonly.

The property value that retrieves the amount that the mouse wheel has changed.

Ctrl Property

Type: sx_bool. Access: readonly.

The property that indicates the CTRL key is down.

Shft Property

Type: sx_bool. Access: readonly.

The property that indicates the SHIFT key is down.

IViewerEvent_KeyboardActivity Interface

The interface provides the keyboard input. It is derived from IViewerCancelableEvent. Because this is cancelable event, you can skip an input action.

Key Property

Type: sx_byte. Access: readonly.

The property value retrieves the key code.

Important! The key codes are mapped according to the layout of specified platform (see IViewerConfig:: **KeyboardLayout** property).

Pressed Property

Type: sx_bool. Access: readonly.

The property that indicates the specified key is pressed (or released) now.

RepeatCount Property

Type: sx_uint. Access: readonly.

The property value is the number of times the keystroke is autorepeated as a result of the user holding down the key.

Shft Property

Type: sx_bool. Access: readonly.

The property that indicates the SHIFT key is pressed now.

Ctrl Property

Type: sx_bool. Access: readonly.

The property that indicates the CTRL key is pressed now.

Alt Property

Type: sx_bool. Access: readonly.

The property that indicates the ALT key is pressed now.

RightCtrl Property

Type: sx_bool. Access: readonly.

The property that indicates the right CTRL key is pressed now.

RightAlt Property

Type: sx_bool. Access: readonly.

The property that indicates the right ALT key is pressed now.

IViewerEvent_ToolActivity Interface

The interface provides the events of the viewer tools. You must implement processing of this event because the viewer tools can't work without handling of its events. The interface is derived from IViewerCancelableEvent. Because this is cancelable event, you can skip many tool actions according to your criteria.

State Property

Type: IToolState. Access: readonly.

The property retrieves the interface of active tool.

Important! You may cache this interface (active tool) and use it between the tool events, but not later than the tool mode (see ToolMode property of the viewer) will be changed.

IToolState Interface

The interface represents active viewer tool and actual tool state.

Important! This interface represents a tool object within the viewer until the other tool will be activated or other document will be loaded to the viewer. Therefore you may cache this interface and use it between the tool events, but not later than the tool mode will be changed (see **sx_tool_change** description below).

ToolMode Property

Type: tool_mode. Access: readonly.

The property retrieves the type of actual tool, it duplicates the similar property of viewer. Using this property you can type cast the **IToolState** interface to the derived interfaces:

- IToolTracker (for tool_mode_tracker)
- IToolRectMark (for tool_mode_rect_mark)
- IToolSnapshot (for tool_mode_snapshot)
- IToolLens (for tool_mode_lens)
- IToolTextSelect (for tool_mode_select_text)
- IToolStylus (for tool_mode_stylus)
- IToolAnnot (for tool_mode_annot)
- IToolFreeText (for tool_mode_free_text)
- IToolInk (for tool_mode_ink)
- IToolEraser (for tool_mode_eraser)
- IToolShape (for tool_mode_shape)

Activity Property

Type: sx_tool_act. Access: readonly.

The property provides the status of tool activity.

- **sx_tool_change** - this status notifies about activating new tool in the viewer, under this event you can setup new tool and/or cache tool interface in your application. If tool type is "tool_mode_none", than you may cache tool interface because in is temporary object.
- **sx_tool_start** - the tool workflow is started
- **sx_tool_apply** - the tool workflow is successfully finished
- **sx_tool_cancel** - the tool workflow is caneled

Page Property

Type: sx_uint. Access: readonly.

The property value retrieves an index of page where tool is loacted now. It can be "-1" if tool workflow is not started yet.

Reset Method

The method breaks the tool workflow if it started already.

Enumerations

Viewer_flags Enumeration

This enumeration defines optional styles of the viewer window.

viewer_flags_scrollbar_always

The window scrollbars must be visible always.

viewer_flags_scrollbar_autohide

The window has the scrollbars and hide its if no scroll is possible.

viewer_flags_border_flat

The window has the flat border.

viewer_flags_border_3D

The window has the 3D style border.

Tool_mode Enumeration

This enumeration defines tool modes of the document viewer. Every tool mode implements own logics of reaction to end-user activities in the viewer. For example, you need “tool_mode_select_text” if you want allow to select text on the pages interactively.

tool_mode_none

The default mode of the viewer, it supports optionally the link navigation, PDF form filling and PDF defined actions. Please note, other tool modes don’t support the link navigation, PDF form filling and PDF defined actions.

tool_mode_custom

This mode provides the possibility to implement completelly own logics for processing of end-user activity in the viewer.

tool_mode_tracker

This mode provides the possibility to specify place or rectangle area on the page in the viewer.

This mode can be used for adding sticky notes, stamps, watermarks and similar PDF annotations.

tool_mode_rect_mark

This mode provides the possibility to mark rectangle zones on the pages of viewed document. The tool supports single or multiple zones on the page, adding, moving and resize of existing zones.

tool_mode_snapshot

This mode provides the snapshot tool.

tool_mode_lens

This mode provides the lens tool.

tool_mode_measure

This mode provides the measure tool, it is on construction yet.

tool_mode_select_text

This mode provides the possibility to select text on single page in the viewer. This tool can be used for copying in the clipboard, marking keyword placement and text markup in PDF documents.

tool_mode_annot

This mode provides the possibility to manage different types of PDF annotations. The tool supports single or multiple selection, moving and resize of annotations. Using this tool you can implement removing annoations, view or edit annotation properties (in own window or form).

tool_mode_free_text

This mode provides the possibility to create or edit FreeText type of PDF annotations (textbox and callout).

tool_mode_ink

This mode provides the possibility to create or edit Ink type of PDF annotations.

tool_mode_eraser

This mode provides the possibility to edit Ink type of PDF annotations.

tool_mode_shape

This mode provides the possibility to create or edit shape types of PDF annotations (line, rectangle, circle, polyline, polygone).

Viewer_event Enumeration

This enumeration defines types of events used by the document viewer.

viewer_event_attach_doc

New document was loaded in the viewer.

viewer_event_detach_doc

The viewed document was unloaded from the viewer.

viewer_event_invalidate

The viewport was invalidated and will be redrawed.

viewer_event_change_act_page

The active page is changed.

viewer_event_change_page_visibility

The visible page(s) was changed.

viewer_event_change_layout

The document was relayouted.

viewer_event_change_scroll

The document canvas was scrolled.

viewer_event_change_config

The viewer configuration was changed.

viewer_event_change_marking

The marking was changed or cleared.

viewer_event_change_thumb_marking

The page(s) was selected or unselected.

viewer_event_pdf

The PDF specific event.

viewer_event_xps_link

The XPS navigation event.

viewer_event_popup_activity

The event of the viewer popup.

viewer_event_set_page

The cancelable event - try to navigate.

viewer_event_scroll

The cancelable event - try to scroll.

viewer_event_mouse_activity

The cancelable event - mouse / touch / stylus activity of end-user.

viewer_event_keyboard_activity

The cancelable event - keyboard input of end-user.

viewer_event_tool_activity

The cancelable event of the viewer tool.

SX::PDF Namespace

Coordinate System and Unit of Measure

PDF defines own coordinate system that is called user space. The positive x axis extends horizontally to the right and the positive y axis vertically upward, as in standard mathematical practice. In most cases, the origin of the coordinate system (0,0) coincides with the lower left corner of the page, but this is not mandatory. As a rule, the unit of measure is the point (1/72 of an inch). But every page may have optional property - a multiplicative factor that shall give the size of page user space units, in multiples of point (1/72 of an inch), see IPage::UserUnit. Please note, **y-member** of sx_rect structure contains lowest coordinate of specified area, it is bottom side of rectangle within PDF coordinate system.

IDocument Interface

The interfaces provides interface of PDF document object implemented by SDK. This interface is derived from IBaseDocument. You can create new instance of PDF document object using IAppFactory::CreatePDFDocument method.

Windows: We highly recommend you use an instance of an ObjPtr template class where you can use an PDF::IDocument pointer in a safe manner.

FilePath Property

Type: sx_str. Access: readonly.

The property provides file path of the document if it opened from file or saved to file, otherwise the property is null.

Properties Property

Type: IDocProperties. Access: readonly.

The property provides the document properties.

Pages Property

Type: IPages. Access: readonly.

The property provides the pages collection.

Outline Property

Type: IOutlineItem. Access: readonly.

The property provides root item of the document outline. It produces an exception ErrorAuthFailed if your license does not permit to use outline elements.

NamedDests Property

Type: INamedDests. Access: readonly.

The property provides a collection of named destinations in the document. It produces an exception ErrorAuthFailed if your license does not permit to use named destinations.

FormFields Property

Type: IFormFields. Access: readonly.

The property provides an interactive form of the document. It produces an exception ErrorAuthFailed if your license does not permit to use form fields.

Layers Property

Type: ILayers. Access: readonly.

The property provides a collection of layers in the document. It produces an exception ErrorAuthFailed if your license does not permit to use layers.

Files Property

Type: IEmbeddedFiles. Access: readonly.

The property provides a collection of files embedded in the document. It produces an exception ErrorAuthFailed if your license does not permit to use embedded files.

Resources Property

Type: IDocResources. Access: readonly.

The property provides the document resources.

OpenStream Method

The method loads the PDF document from stream.

Stream Parameter

Type: ISequentialStream.

The document data stream. Please reset stream before load.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

OpenFile Method

The method loads the document from file.

FilePath Parameter

Type: sx_str.

Fully qualified path of the file to open. The file must have read access.

MaxCache Parameter

Type: sx_uint.

Max size of memory cache. If file size is less than this parameter, the file will be completelly cached in memory and immediately closed after calling this method. In other case, the file is still open and locked until the document object will be destroed or document saved.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SelectRevision

The method selects specified revision (version) of document. We recommend you use this method immediately after calling OpenStream or OpenFile because method works until the document still in nonmodified state, otherwise it is failed. See also IDocProperties::RevisionsCount and IDocProperties::CurrentRevision properties.

RevIndex Parameter

Type: sx_uint.

Zero-based index of revision. Must be less then IDocProperties::RevisionsCount.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SaveAsStream

The method saves the PDF document to the stream.

Stream Parameter

Type: ISequentialStream.

The stream for save. Please reset stream before save.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SaveAsFile

The method saves the document to the file.

FilePath Parameter

Type: sx_str.

Fully qualified path of the file to save. Method creates a new file, always. If the specified file exists it must have write access.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SaveRevisionAsStream

The method appends new revision to the PDF document (any changes or digital signature as an incremental update). It fails if document is non-modified or wasn't loaded from the stream or file. **Please note:** the method writes data of complete document to the stream, not an incremental update data only.

Stream Parameter

Type: ISequentialStream.

The stream for save. Please reset stream before save.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SaveRevisionAsFile

The method appends new revision to the PDF document (any changes or digital signature as an incremental update). It fails if document is non-modified or wasn't loaded from the stream or file. **Please note:** the method writes data of complete document to the file, not an incremental update data only.

FilePath Parameter

Type: sx_str.

Fully qualified path of the file to save. Method creates a new file, always. If the specified file exists it must have write access.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

StartCombine

The method checks the possibility to combine with other document (copy pages and other elements from specified document). It prepares combining operation also. You can use CombinePage or CombineAllPages methods to specify pages to be combined. You must call the EndCombine method to comply combine operation.

Please note! Using the combine methods you can not only combine different types of documents (PDF, XPS, etc.) but also convert PDF to XPS and XPS to PDF.

SourceDocument Parameter

Type: IBaseDocument.

The pages (and other elements as links or annotations) of SourceDocument will be copied in current document.

CombineOptions Parameter

Type: sx_flags, see also combine enumeration.

The parameter specifies combine options.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

CombinePage

The method copies one page from SourceDocument (see StartCombine method above) to the current document, optionally with PDF annotations. The method can't be called twice for the same page. Also it cannot be called together with CombineAllPages method.

From Parameter

Type: sx_uint.

The page index in SourceDocument. Can be greater than or equal to 0 and less than page count.

To Parameter

Type: sx_uint.

The position in the current document where a copied page will be inserted. Can be greater than or equal to 0 and less than or equal page count.

CombineOptions Parameter

Type: sx_flags, see also annot_type enumeration.

The parameter specifies the types of PDF annotations to be copied. You can combine the enumeration values by using the bitwise OR operator.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

CombineAllPages

The method copies all pages of SourceDocument (see StartCombine method above) to the current document, optionally with PDF annotations. It cannot be called together with CombinePage method.

To Parameter

Type: sx_uint.

The position in the current document where the copied pages will be inserted. Can be greater than or equal to 0 and less than or equal page count.

CombineOptions Parameter

Type: sx_flags, see also annot_type enumeration.

The parameter specifies the types of PDF annotations to be copied. You can combine the enumeration values by using the bitwise OR operator.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

EndCombine

The method performs necessary actions to combine two documents after calling StartCombine and CombinePage/CombineAllPages methods.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

ConformToPDFA

The method converts the PDF document to make it conform to specified PDF/A format.

Please note that

- some correct PDF documents can't be correctly converted to PDF/A format and method fails, the detailed reason you get using callback messaging (see information about third parameter below)
- any changes in the document after calling this method, can make document non conform to PDF/A, so we highly recommend you to save conform document immediately after successful calling of this method
- you must disable document encryption if available, before call this method

Version Parameter

Type: pdfa.

The PDF/A version to which a document must be conform.

Opt Parameter

Type: sx_flags.

The reserved parameter, must be 0.

Status Parameter

Type: IStatusPDFA.

The callback interface, the parameter is optional, but strongly recommended. You need to implement it to get detailed information about converting procedure.

UWP / WinRT / WinPhone: At these platforms the OnMessagePDFA event of Document class is declared instead of third parameter. Please define an event handler method and add an instance to this event.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IDocProperties Interface

The interface represents the properties of PDF document.

ID1 Property

Type: sx_str.

The property provides the permanent identifier of the document.

ID2 Property

Type: sx_str.

The property provides the changing identifier of the document.

FormatVersion Property

Type: sx_str.

The property provides the PDF format version, the valid version values are “1.0”, “1.1”, “1.2”, “1.3”, “1.4”, “1.5”, “1.6”, “1.7”, “2.0”.

Linearized Property

Type: sx_bool. Access: readonly.

The property is true, if document was loaded from file or stream optimized for delivery from servers to web browsers over the Internet.

Security Property

Type: IDocSecurity. Access: readonly.

The property provides security properties of the document.

IDocSecurity Interface

This interface manages the security properties of PDF document.

Security Property

Type: doc_security. Access: readonly.

The property determine whether security is enabled and type of PDF security. The values can be:

- doc_security_none - the document is not encrypted
- doc_security_undefined - the document is encrypted, encryption method is unknown
- doc_security_password - the document is encrypted, standard password based security
- doc_security_pki - the document is encrypted, standard public-key based security

StandardSecurity Property

Type: IStandardSecurity. Access: readonly.

The property provides the interface of two standard types of document security.

You may use this property if **Security** property equals doc_security_password (cast IStandardSecurity to IPasswordSecurity) or it equals doc_security_pki (cast IStandardSecurity to IPKISecurity).

HandlerName Property

Type: sx_str.

The property provides the name of used security method.

HandlerFormat Property

Type: sx_str.

The property provides the format description of used security method.

ChangeSecurity Method

The method sets or removes the document security.

Sec Parameter

Type: doc_security.

The new security type. You can't set doc_security_undefined value here.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IStandardSecurity Interface

The interface represents the base properties of two standard types of document security. So two interfaces: IPasswordSecurity and IPKISecurity inherit it.

FullControl Property

Type: sx_bool. Access: readonly.

If property is true, unlimited access to the document is granted. This unlimited access includes the ability to change the document's security and access permission. In case of password based security, you must load document using owner password.

Important! The IDocSecurity::ChangeSecurity method allows you completelly reset security without this permission.

EncryptionAlgorithm Property

Type: encryption_algorithm.

The property determines encryption algorithm. The values can be:

- encryption_algorithm_rc4 - RC4 encryption algorithm
- encryption_algorithm_aes - AES encryption algorithm

EncryptionKeyLength Property

Type: encryption_key_length.

The property determines length of key used for encryption. The values can be:

- encryption_key_length_40
- encryption_key_length_128
- encryption_key_length_256
- encryption_key_length_384
- encryption_key_length_512

Permissions Property

Type: doc_permissions.

The property provides the document permissions. The permissions are:

- doc_permissions_full_control

- doc_permissions_normal_print
- doc_permissions_modify_content
- doc_permissions_copy_content
- doc_permissions_modify_annots
- doc_permissions_fill_form
- doc_permissions_accessibility
- doc_permissions_assemble
- doc_permissions_quality_print

The multiple permissions can be combined by using the bitwise OR operator.

IPasswordSecurity Interface

The interface inherits IStandardSecurity interface in the case of use of doc_security_password security.

SetOwnerPassword Method

The method sets owner password for encryption or decryption of secured PDF document. It returns boolean status of call.

You must call this method successfully for opening of encrypted document and getting full control over the document, otherwise it is enough to call SetUserPassword method successfully.

You must call this method successfully before save encrypted document.

Password Parameter

Type: sx_str.

The property specifies the owner password.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SetUserPassword Method

The method sets user password for encryption or decryption of secured PDF document. It returns boolean status of call.

You need call this method successfully for opening of encrypted document and getting access to document according to security permissions, if you need full acces to document you must call SetOwnerPassword method successfully.

You must call this method successfully before save encrypted document.

Password Parameter

Type: sx_str.

The property specifies the owner password.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IPages Interface

The interface represents a pages collection of PDF document.

Count Property

Type: sx_uint. Access: readonly.

The property retrieves number of pages in the document.

Item Method

The method retrieves the specified page. It returns IPage interface.

Index Parameter

Type: sx_uint.

The index of a page. Can be greater than or equal to 0 and less than page count.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Add Method

The method adds new empty page to the end of document. It returns IPage interface. **Note**: you must set media box of a page after adding.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Insert Method

The method inserts new empty page to the specified position. It returns IPage interface. **Note**: you must set media box of a page after inserting.

To Parameter

Type: sx_uint.

The position to insert a page. Can be greater than or equal to 0 and less than or equal to page count.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Copy

The method duplicates a page in this document or copies a page from other document (PDF or XPS) and inserts copied page to the specified position. It returns IPage interface.

Page Parameter

Type: IPage or XPS::IPage.

The page to be copied.

To Parameter

Type: sx_uint.

The position to insert a page. Can be greater than or equal to 0 and less than or equal to page count.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Move

The method moves a page to the specified position.

From Parameter

Type: sx_uint.

The index of a page to move. Can be greater than or equal to 0 and less than page count.

To Parameter

Type: sx_uint.

The position to insert a page. Can be greater than or equal to 0 and less than or equal to page count.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Remove

The method removes a specified page from the document.

Index Parameter

Type: sx_uint.

The index of a page. Can be greater than or equal to 0 and less than page count.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IndexOf

The method searches for the specified page object and returns the zero-based index of a page within the entire pages collection.

Page Parameter

Type: IPage.

The page object.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Compact

The method compacts memory used by page content and resources.

CompactObjects Parameter

Type: sx_flags, see also PDF::compact_objects.

The options of compact operation.

IPage Interface

The interface represents a page of PDF document.

Width Property

Type: sx_uint. Access: readonly.

The width of a page, in points. This property don't considers values of UserUnit and Rotation properties. See also GetMediaBox method. Use SetMediaBox or SetCropBox methods to change page width.

Height Property

Type: sx_uint. Access: readonly.

The height of a page, in points. This property don't considers values of UserUnit and Rotation properties. See also GetMediaBox method. Use SetMediaBox or SetCropBox methods to change page height.

Rotation Property

Type: sx_rotation.

The number of degrees (multiple of 90) by which the page shall be rotated clockwise when displayed or printed.

UserUnit Property

Type: sx_double.

A positive number that shall give the size of user space units, in multiples of point.

Annots Property

Type: IAnnots. Access: readonly.

The property provides the annotations collection of the page.

Label Property

Type: sx_str. Access: readonly.

The property provides the page label. See IPages methods to change labeling of the pages.

Thumbnail Property

Type: IRasterImage. Access: readonly.

The property provides the page thumbnail.

GetMediaBox Method

The method retrieves a rectangle in page user space, that shall define the boundaries of the page. The visible region of the page can be clipped, see GetCropBox method below.

Box Parameter (or return parameter)

Type: sx_rect.

The boundaries of the page.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SetMediaBox Method

The method sets a rectangle in page user space, that shall define the boundaries of the page. If the page has crop box and it is greater than new media box, the library inflates media box up to crop box, see GetCropBox method below.

Box Parameter

Type: sx_rect.

The boundaries of the page.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetCropBox Method

The method retrieves a rectangle in page user space, that shall define the visible region of the page. If crop box isn't defined, the method retrieves media box of the page, see GetMediaBox method above.

Box Parameter (or return parameter)

Type: sx_rect.

The visible region of the page.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SetCropBox Method

The method sets a rectangle in page user space, that shall define the visible region of the page. If the new crop box is greater than media box of the page, the library deflates crop box up to media box, see GetMediaBox method above.

Box Parameter

Type: sx_rect.

The visible region of the page.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SetThumbnail Method

The method renders new thumbnail of a page and saves it inside the document. Call the method with zero parameters to remove existing thumbnail. The method fails if you try to render a big thumbnail (with DPI > 18) without "Draw Page" permission in your license.

MaxW / MaxH Parameters

The parameters specify the max size of new thumbnail in pixels.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

CalcMatrix Method

The method calculates a transformation matrix using properties of the page and specified parameters of external coordinate system (for example, unified page coordinate system of viewer). Using this matrix you can transform any coordinates from the page space to the external coordinate system and vice versa (using inverted matrix).

M Parameter

Type: sx_matrix.

The transformation matrix, it is output parameter.

Zoom Parameter

Type: sx_double.

The scale value, where 1.0 is original size.

Angle Parameter

Type: sx_rotation.

The number of degrees (multiple of 90) by which the page shall be rotated.

Dx Parameter

Type: sx_double.

The horisontal position of the page in external coordinate system.

Dy Parameter

Type: sx_double.

The vertical position of the page in external coordinate system.

DPI Parameter

Type: sx_uint.

The DPI property external coordinate system.

GetRichContent Method

The method provides direct access to the page content as a hierarchical structure of content objects. It retrieves IRichContent interface.

Reserved Parameter

The parameter reserved for future use.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IAnnots Interface

The interface represents an annotations collection of the page. Using this interface you can enumerate annotations, create or remove its, change order (Z-order on the page) of the annotations.

Count Property

Type: sx_uint. Access: readonly.

The property retrieves number of annotations on the document.

Item Method

The method retrieves the specified annotation. It returns IAnnot interface.

Index Parameter

Type: sx_uint.

The index of an annotation. Can be greater than or equal to 0 and less than page count.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

FindItem Method

Method retrieves an annotation located at the coordinates of x and y. Most often used with pointer operations in the viewer to determine an annotation under the pointer. It returns IAnnot interface.

Important! The method enumerates annotations from last to first (according to visibility of overlapped annotations).

P Parameter

Type: sx_point.

The coordinates to find. They must be in page user space.

Mask Parameter

Type: sx_flags, see also annot_type enumeration.

The type(s) of annotations to be found. You can combine the enumeration values by using the bitwise OR operator.

FromAnnot Parameter

Type: IAnnot.

The annotation returned by a previous call to this method, this parameter allow you to find all overlapped annoatations. For first call it must be null.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

InsertNew Method

The method creates the annotation of specified type. It returns IAnnot interface. You must set necessary properties of annotation, at least the rectangle area.

Type Parameter

Type: sx_flags, see also annot_type enumeration.

The type of new annotation.

Before Parameter

Type: sx_uint.

The new annotation will be inserted into the collection before this annotation. Can be greater than or equal to 0 and equal to annotation count.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

InsertNewWidget Method

The method creates the widget annotation. It returns IWidget interface. You must set necessary properties of annotation, at least the rectangle area.

Field Parameter

Type: IFormField.

The new widget represents the specified form field.

Before Parameter

Type: sx_uint.

The new annotation will be inserted into the collection before this annotation. Can be greater than or equal to 0 and equal to annotation count.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Insert Method

The method inserts existing annotation on the page. The annotation should be from this document and shouldn't be on this page already.

Annot Parameter

Type: IAnnot.

The existing annotation.

Before Parameter

Type: sx_uint.

The new annotation will be inserted into the collection before this annotation. Can be greater than or equal to 0 and equal to annotation count.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Move Method

The method moves annotation in the collection order.

Annot Parameter

Type: IAnnot.

The annotation to be moved.

Before Parameter

Type: sx_uint.

The annotation will be moved into the collection before this annotation. Can be greater than or equal to 0 and equal to annotation count.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Remove Method

The method removes annotation from the page.

Annot Parameter

Type: IAnnot.

The annotation to be removed.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

ResetPopup Method

The method creates or removes popup annotation of markup annotation.

MarkupAnnot Parameter

Type: IAnnot.

The markup annotation.

Clear Parameter

Type: sx_bool.

If parameter is false, new popup annotation will be created, otherwise the popup will be removed.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IAnnot Interface

The interface represents a page annotation in PDF document.

Type Property

Type: annot_type. Access: readonly.

The property provides the type of annotation. Using this property you can type cast the **IAnnot** interface to the derived interfaces.

IStatusPDFA Interface

The callback interface, you need to implement it for using in ConformToPDFA method for monitoring of converting process.

OnMessage

The method is called everytime when ConformToPDFA procedure needs to inform your application about error, warning or other event in process of converting. After first error message, ConformToPDFA procedure throws an exception and breaks converting. The method must return **true** to continue converting or **false** to break ConformToPDFA procedure.

Message Parameter

Type: IMessagePDFA.

The converting message.

IMessagePDFA Interface

The interface represents a converting message for procedure of PDF/A converting. The message provides category of message (error, warning, information) and coded description of a message.

Type Property

Type: sx_flags. Access: readonly.

The property provides the category and description of message. Both values are "packed" in this property, as set of bit flags.

Message Category

Bits 24 through 31 of Type property contain the message category. Use bitwise AND operation between Type property and **PDF::pdfa_mess_cat** mask to get category of message, it could be:

- pdfa_mess_info - this message informs your application about performing of converting
- pdfa_mess_warn - this message informs your application about modifications in the document to make

it PDF/A conform

- pdfa_mess_error - this message informs your application about finding non-conform and non-convertable element or missing mandatory element in the processed document, after this message the converting procedure will be breaked with exception.

Message Description

Bits 0 through 23 of Type property contain the message description. Use bitwise AND operation between Type property and **PDF::pdfa_mess_type** mask to get coded description of message, it could be:

- pdfa_mess_begin - information, the procedure started
- pdfa_mess_end - information, the procedure successfully performed
- pdfa_mess_security - error, the document must be non-encrypted
- pdfa_mess_version - warning, format version was adjusted
- pdfa_mess_doc_id - warning, document ID was generated
- pdfa_mess_doc_actions - warning, document actions was removed
- pdfa_mess_optional_content - warning, optional content groups was removed, the optional content transformed to normal content
- pdfa_mess_need_appearances - warning, the form property "NeedAppearances" was cleared
- pdfa_mess_field_actions - warning, form field actions was removed
- pdfa_mess_annot_appearance - warning, the annotation apperance was adjusted
- pdfa_mess_portfolio - error, can't convert the portfolio to PDF/A
- pdfa_mess_emb_files - warning, the embedded file(s) was removed
- pdfa_mess_xfa - error (dynamic XFA isn't allowed) or warning ("static" XFA removed)
- pdfa_mess_mark_info - warning, the document mark info was adjusted
- pdfa_mess_java_script - warning, the document JavaScript was removed
- pdfa_mess_lang - warning, the document language identifier was adjusted
- pdfa_mess_struct_tree - warning, the logical structure tree of a document was adjusted
- pdfa_mess_output_intents - error (can't find standard color profile) or warning (standard color profile was embedded)
- pdfa_mess_annot_type - warning, annotation of non-allowed type was removed

- pdfa_mess_action_type - warning, event action of non-allowed type was removed
- pdfa_mess_widget_actions - warning, widget action was removed
- pdfa_mess_annot_opacity - warning, annotation opacity was cleared
- pdfa_mess_annot_flags - warning, annotation flags was adjusted
- pdfa_mess_page_actions - warning, page action was removed
- pdfa_mess_image_alternates - warning, alternate images was removed
- pdfa_mess_image_opi - warning, image OPI entry was removed
- pdfa_mess_image_interpolate - warning, image interpolation property was cleared
- pdfa_mess_rendering_intent - warning, image rendering intent was cleared
- pdfa_mess_font_embedding - error (non-embedded font can't be embedded) or warning (nonembedded font was embedded)
- pdfa_mess_xobject_opi - warning, composite image OPI entry was removed
- pdfa_mess_xobject_ps - warning, PostScript image was removed
- pdfa_mess_xobject_ref - warning, reference image was removed
- pdfa_mess_smask - warning, soft mask (mask image) was removed
- pdfa_mess_transparency - warning, transparency property was cleared
- pdfa_mess_jpx_compression - warning, JPEG2000 compression was replaced with JPEG
- pdfa_mess_lzw_compression - warning, LZW compression was replaced with Flate
- pdfa_mess_crypt_filter - error, crypto filter isn't allowed
- pdfa_mess_external_stream - warning, reference to an external stream was removed
- pdfa_mess_perms - warning, document permissions was adjusted
- pdfa_mess_blend_mode - warning,blend mode was removed
- pdfa_mess_glyph_widths - warning, glyph width was adjusted
- pdfa_mess_font_cidset - warning, CID set was adjusted or removed
- pdfa_mess_font_tounicode - error, missed a mandatory unicode map of a font
- pdfa_mess_page_userunit - error or warning, page user unit was cleared
- pdfa_mess_missing_glyph - error, missed a mandatory glyph of a character

- pdfa_mess_struct_type - warning, type of the document logical structure was adjusted
- pdfa_mess_struct_lang - warning, language ID of the document logical structure was adjusted
- pdfa_mess_embedded_files - warning, embedded files was removed
- pdfa_mess_renditions - warning, renditions was removed
- pdfa_mess_alternate_presentations - warning, alternate presentations was removed
- pdfa_mess_page_pressteps - warning, sub-page navigation was removed
- pdfa_mess_file_association - warning, embedded files was adjuested

Enumerations

Compact_objects Enumeration

The members of this enumeration are the options for compact operation of the page. You can combine the enumeration values by using the bitwise OR operator.

compact_objects_content

The page content must be compacted.

compact_objects_composite_images

The composite images of the page must be compacted.

compact_objects_raster_images

The raster images of the page must be compacted.

compact_objects_fonts

The fonts used by the page must be compacted.

Annot_type Enumeration

The members of the enumeration are the possible types of PDF annotations.

annot_type_undefined

annot_type_text

annot_type_link

annot_type_freetext

annot_type_line

annot_type_square

annot_type_circle
annot_type_polygon
annot_type_polyline
annot_type_highligh
annot_type_underline
annot_type_squiggly
annot_type_strikeout
annot_type_caret
annot_type_stamp
annot_type_ink
annot_type_popup
annot_type_attachment
annot_type_sound
annot_type_movie
annot_type_screen
annot_type_widget
annot_type_printermark
annot_type_trapnet
annot_type_watermark
annot_type_3D
annot_type_redact
annot_type_projectio
annot_type_richmedia
annot_type_reply_to

SX::XPS Namespace

IPackage Interface

The interfaces provides interface of XPS package object implemented by SDK. This interface is derived from IBaseDocument. You can create new instance of XPS package object using IAppFactory::CreateXPSPackage method.

SX::Content Namespace

This namespace covers the Rich Content API (RC) that provides a close access to page content of PDF or XPS documents. Using RC interfaces and methods you can create content of new pages (fill the blank pages), edit content of pages created by any other software. Also you can explore content of pages, for such tasks as: recognize placement content objects (text, images, vector elements) on the page, format of text, etc.

The RC API represents page content as a hierarchical structure of content objects. The PDF and XPS documents store the page content in different own formats and both formats are not suitable for direct using (creating / editing / exploring) of content object. The library translates the internal structures of page content in the hierarchical structure of content objects independently from native format of content. If the RC structure (or objects) was modified, the library makes backward translation (from RC to native format) automatically.

The key interface of RC API is the IRichContent interface, see more details below.

Coordinate System and Unit of Measure

The unified coordinate system of RC is traditional for the Windows. The unit of measure is the point (1/72 of an inch). The x-coordinates increase to the right; y-coordinates increase from top to bottom.

The content owners (page or composite image) in the PDF or XPS documents have own native coordinate systems, these systems are different and don't coincide with this unified coordinate system. You can transform the coordinates between RC and native systems using GetMatrix method of IRichContent interface.

IRichContent Interface

The interface is starting point of Rich Content API it provides a hierarchical structure of content objects (for a page or self-contained composite image in the PDF/XPS documents). Also it is object factory of the page content.

Page content to RC

You can get it using the GetRichContent method of IPage interface within PDF / XPS namespaces. When this method is called first time, there library exports the page content (if present) to a hierarchical structure of content objects automatically.

RC to page content

If content was modified (or created) it will be imported to the page (and replaces old page content) automatically when you release (dispose) all references to this interface.

Windows: We highly recommend you use an instance of an ObjPtr template class where you can use an **Content::IRichContent** pointer in a safe manner.

Objects Property

Type: IRichCollection. Access: readonly.

The property provides first level collection of the content objects.

Modified Property

Type: sx_bool.

The property indicates that the content has been modified.

GetUPI Method

The method retrieves "units per inch" property. Is a measure of units in the native coordinate system of content owner (page or self-contained composite image), in particular the number of units that can be placed within the span of 1 inch.

GetMatrix Method

The method retrieves matrix of difference between coordinate system of content owner (page or self-contained composite image) and unified coordinate system of RC. Using this matrix you can transform any coordinates from the page space to the RC space and vice versa (using inverted matrix). It works as the PDF::IPage::CalcMatrix method.

In other words inversion of this this matrix gives you transformation to the "coordinate system" of first level collection (see Objects property above) which is exactly native page space.

M Parameter

Type: sx_matrix.

The transformation matrix, it is output parameter.

Clear Method

The method removes all content objects in first level collection.

CreateBrush Method

The method creates a new brush object. It returns IBrush interface.

Important: you may not use this object with the content objects which are not childs of this IRichContent.

Type Parameter

Type: brush_type.

The type of the brush.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

CreateSolidBrush Method

The method creates a new solid brush object. It returns ISolidBrush interface.

Important: you may not use this object with the content objects which are not childs of this IRichContent.

Color Parameter

Type: sx_color.

The color of the brush.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

CreateFont Method

The method creates a new font object based at the system font. It returns IFont interface.

Important: you may not use this object with the content objects which are not childs of this IRichContent.

Family Parameter

Type: sx_str.

The font family name.

Weight Parameter

Type: sx_uint.

The density of a typeface, use 400 for regular density and 700 for bold density.

Italic Parameter

Type: sx_bool.

The style of a font, italic or normal.

Stretch Parameter

Type: sx_uint.

The degree to which a font has been stretched compared to a font's normal aspect ratio, use 5 for normal, 3 for condensed and 7 for expanded styles of a font.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

CreatePath Method

The method creates a new geometry path object. It returns IPath interface.

Important: you may not use this object with the content objects which are not childs of this IRichContent.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

CreatePen Method

The method creates a new pen object. It returns IPen interface.

Important: you may not use this object with the content objects which are not childs of this IRichContent.

Width Parameter

Type: sx_double.

The width of pen, in points.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

CreateTextStyle Method

The method creates a new text style object. It returns ITextStyle interface.

Important: you may not use this object with the content objects which are not childs of this IRichContent.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IRichCollection Interface

This interface represents collection of content objects in a hierarchical structure on a “horizontal” level. The order of objects in a collections explicitly defines the z-order of the order. When objects overlap, z-order determines which one covers the other. An object with a larger index within a collection generally covers an object with a lower one.

The coordinate system for the first level collection is native page space (see Content::IRichContent::GetMatrix method), for the second and higher levels it is "coordinate system" of group content object (see Content::IRichObject::GetTransformation method).

Root Property

Type: IRichContent. Access: readonly.

The property retrieves interface which represents current hierarchical structure of content objects.

Group Property

Type: IRichGroup. Access: readonly.

The property retrieves an immediate parent group object for the collections of second or higher levels. For the first level collection this property is null.

Count Property

Type: sx_uint. Access: readonly.

The property retrieves number of objects in the collection.

Item Method

The method retrieves the specified object. It returns IRichObject interface.

Index Parameter

Type: sx_uint.

The index of an object. Can be greater than or equal to 0 and less than number of objects.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IndexOf Method

The method searches for the specified object and returns the zero-based index of an object within the collection.

Page Parameter

Type: IRichObject.

The content object.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Insert Method

The method creates new content object of specified type and inserts it to the specified position. It returns IRichObject interface.

To Parameter

Type: sx_uint.

The position to insert an object. Can be greater than or equal to 0 and less than or equal to number of objects.

Type Parameter

Type: rich_type.

The type of content object.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

InsertCopy Method

The method clones a specified object and inserts new copy to the specified position. It returns IRichObject interface. If a copied object uses shareable resource objects, they will be used by reference. For example if you copy one text object to another and than you change text style object of new object, the original text object will use a changed style also. The child content objects of group object are copied with parent object.

Important. The object to be copied must be located in the any collection within a hierarchical structure where current collection is located, in other words both collections must reference to one IRichContent object (see Root property).

To Parameter

Type: sx_uint.

The position to insert an object. Can be greater than or equal to 0 and less than or equal to number of objects.

Object Parameter

Type: IRichObject.

The content object to be copied.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Move Method

The method moves an content object to the specified position within collection. In other words this method changes z-order of objects.

From Parameter

Type: sx_uint.

The index of an object to move. Can be greater than or equal to 0 and less than number of objects.

To Parameter

Type: sx_uint.

The position to insert an object. Can be greater than or equal to 0 and less than or equal to number of objects.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Remove Method

The method removes a specified content object from the collection and a hierarchical structure.

Index Parameter

Type: sx_uint.

The index of an object. Can be greater than or equal to 0 and less than number of objects.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IRichObject Interface

The interfaces declares common properties (and functionality) for all types of content objects supported by Rich Content API.

Type Property

Type: rich_type. Access: readonly.

The property retrieves type of content object. Using this property you can type cast the **IRichObject** interface to the derived interfaces.

Collection Property

Type: IRichCollection. Access: readonly.

The property retrieves an owner collection of this object. Use this collection to get IRichContent or IRichGroup of for this object (see IRichCollection::Root and IRichCollection::Group properties).

GetMatrix Method

The method retrieves the matrix which defines transformation of object within the coordinate system of owner collection. If you need the matrix of transformation within unified coordinate system of RC you must use **GetTransformation** method.

M Parameter

Type: sx_matrix.

The transformation matrix, it is output parameter.

SetMatrix Method

The method defines transformation of object within the coordinate system of owner collection. If you want define the transformation within unified coordinate system of RC you must use **ApplyTransformation** method.

M Parameter

Type: sx_matrix.

The transformation matrix.

GetTransformation Method

The method retrieves the matrix which defines transformation of object within unified coordinate system of RC. If you need the individual matrix of object (transformation within the coordinate system of owner collection) you must use **GetMatrix** method.

M Parameter

Type: sx_matrix.

The transformation matrix, it is output parameter.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

ApplyTransformation Method

The method defines transformation of object within unified coordinate system of RC. If you want define the individual matrix of object (transformation within the coordinate system of owner collection) you must use **SetMatrix** method.

M Parameter

Type: sx_matrix.

The transformation matrix.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetBBox Method

The method retrieves the bounding box of the content object within the coordinate system specified by transformation matrix (see first parameter).

M Parameter

Type: sx_matrix.

The transformation matrix specifies the coordinate system where you want to get bounding box of object. Use identity matrix to get original dimenshions of object (position is irrelevant in this case) or use matrix returned by **GetTransformation** method to get boundinx box within unified coordinate system of RC.

Rc Parameter

Type: sx_rect.

The bounding box of content object, it is output parameter.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SX::XMP Namespace

Metadata is data that describes the characteristics or properties of a document. It can be distinguished from the main contents of a document. For example, for a word processing document, the contents include the actual text data and formatting information, while the metadata might include such properties as author, modification date, or copyright status. Metadata in PDF documents may be specified for the document itself of for individual components of a PDF document (pages, fonts, images).

Metadata of PDF document is in XML based XMP format (Extensible Metadata Platform). This format standardizes the definition, creation, and processing of metadata. XMP metadata may include properties from one or more of the schemas. An XMP schema is a set of top level property names in a common XML namespace, along with data type and descriptive information. Typically, an XMP schema contains properties that are relevant for particular types of documents or for certain stages of a workflow. PDF and PDF/A formats define some schemas for use in a PDF document.

PDF Xpansion SDK profides XMP API for processing metadata within predefined schemas used in PDF documents. The starting point of XMP API is XMP::IDocument interface, see it below.

IDocument Interface

The interfaces represents a metadata object implemented by SDK. You can create new instance of document object using IAppFactory interface.

Windows: This interface is derived from IRefObject at this platform.

OpenStream Method

The method loads the metadata from XML stream. It returns validation status of metadata. The metadata can be used if status of loading is **xmp_status_xmp**.

Stream Parameter

Type: ISequentialStream.

The XML formatted metadata stream. Please reset stream before load.

Profile Parameter

Type: xmp_profile.

The type of validation profile.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

OpenFromString Method

The method loads the metadata from XML string.. It returns validation status of metadata. The metadata can be used if status of loading is **xmp_status_xmp**.

XMP Parameter

Type: sx_str.

The string which contains the XML formatted metadata. You can get this string from PDF document using PDF::IDocProperties::GetMetadata method.

Profile Parameter

Type: xmp_profile.

The type of validation profile.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SaveAsStream Method

The method saves the XML formatted metadata to the stream.

Stream Parameter

Type: ISequentialStream.

The stream for save. Please reset stream before save.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SaveToString Method

The method saves the XML formatted metadata to the string. It returns IStr object. You can use this string for changing of metadata in PDF document, see PDF::IDocProperties::SetMetadata method.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetScheme Method

The method retrieves interface of specified scheme. It returns base interface IScheme, you can type cast this interface to the derived interface according to requested scheme.

Scheme Parameter

Type: xmp_scheme.

The type of requested scheme.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Reset Method

The method resets the metadata document - remove all schemas (if present) and creates root structure of metadata.

Enumerations

xmp_profile enumeration

The enumeration declares some validation profiles. The profile defines set of schemas which can be used in the metadata.

xmp_status enumeration

The enumeration declares possible results of metadata validation.

SX::ZUGFeRD Namespace

The ZUGFeRD standard describes a document and data format for the exchange of electronic invoices. It is based on a hybrid approach. This means that a PDF/A-3 compliant document is used for the visual representation of the invoice information, while a XML file carries the invoice data, and this file is embedded into the PDF/A-3 document.

ZUGFeRD part of PDF Xpansion SDK provides all necessary functionality for processing ZUGFeRD invoices:

- comprehensive validation of incoming ZUGFeRD files
- read structured und unstructured invoice data
- display and / or print PDF part of invoice in viewer
- create outgoing invoices and save ZUGFeRD conform files, including converting of visual part of invoice to PDF/A-3 format and invoice data to embedded XML attachment

All ZUGFeRD profiles (BASIC, COMFORT and EXTENDED) are supported for reading and writing.

PDF part of invoice can be created:

- from existing PDF file
- from scanned paper pages (image files)
- from existing XPS / OXPS file
- from text (file or stream)
- from the dynamically generated pages (using Rich Content API)

The primary way of use ZUGFeRD API is using of IInvoiceDocument interface which represents a PDF document (file or stream) with embedded XML invoice data and is fully conform to ZUGFeRD standard. This document (PDF part of document) can be displayed in the viewer or printed to the printer.

The secondary way allows processing of XML formatted part of ZUGFeRD invoice (after extracting from PDF/A part or before embedding of new invoice in a PDF/A document). This way supported by IInvoiceStream interface.

IInvoiceDocument Interface

The interfaces provides interface of ZUGFeRD document object (based on PDF file or stream) implemented by SDK. This interface is derived from IBaseDocument. You can create new instance of PDF document object using IAppFactory::CreateInvoiceDocument method.

Windows: We highly recommend you use an instance of an ObjPtr template class where you can use an ZUGFeRD::IInvoiceDocument pointer in a safe manner.

Invoice Property

Type: IInvoice. Access: readonly.

The property retrieves interface which provides access to invoice data.

Title Property

Type: IStr. Access: full.

The property specifies the title of document (within PDF part and metadata).

Subject Property

Type: IStr. Access: full.

The property specifies the subject of document (within PDF part and metadata).

Author Property

Type: IStr. Access: full.

The property specifies the author (person or company) of document (within PDF part and metadata).

Creator Property

Type: IStr. Access: full.

The property specifies the creator (software) of document (within PDF part and metadata).

OpenStream Method

The method loads the ZUGFeRD document from stream. It returns validation status of PDF part including validation of metadata and ZUGFeRD conform embedding of XML invoice. The invoice data can be used if status of loading is **zf_status_valid**. The format and data of XML invoice can be validated using IInvoice::Validate method, we recommend to do it immediately after loading of invoice.

Stream Parameter

Type: ISequentialStream.

The document data stream. Please reset stream before load.

ValidatePDFA Parameter

Type: sx_bool.

If this parameter is false (recommended value) than SDK checks PDF/A conformity within the document metadata only. If this parameter is true, than SDK validates the structure and objects of PDF part for conformity with PDF/A-3. Please note that such validation requires much more resources and time. Also it checks technical characteristics of PDF part and doesn't checks content of PDF part or anything within of XML part of invoice document. Therefore PDF/A validation can reject practically correct ZUGFeRD on formal, rather than substantive reasons.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

OpenFile Method

The method loads the ZUGFeRD document from file. It returns validation status of PDF part including validation of metadata and ZUGFeRD conform embedding of XML invoice. The invoice data can be used if status of loading is **zf_status_valid**. The format and data of XML invoice can be validated using IInvoice::Validate method, we recommend to do it immediately after loading of invoice.

FilePath Parameter

Type: sx_str.

Fully qualified path of the file to open. The file must have read access.

ValidatePDFA Parameter

Type: sx_bool.

If this parameter is false (recommended value) than SDK checks PDF/A conformity within the document metadata only. If this parameter is true, than SDK validates the structure and objects of PDF part for conformity with PDF/A-3. Please note that such validation requires much more resources and time. Also it checks technical characteristics of PDF part and doesn't checks content of PDF part or anything within of XML part of invoice document. Therefore PDF/A validation can reject practically correct ZUGFeRD on formal, rather than substantive reasons.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

CreateFromPDF Method

The method resets the PDF part of ZUGFeRD document, use this method for creating of invoice. The invoice data completely removed (if exists) within this method.

Stream Parameter

Type: ISequentialStream.

The PDF document data stream. Please reset stream before load.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

AddPageFromImage Method

The method resets the PDF part of ZUGFeRD document and inserts first page if PDF part hasn't the pages yet or add new pages otherwise. Use this method for creating of invoice. The invoice data completely removed (if exists) when method adds first page to PDF part.

Image Parameter

Type: Graphics::IBitmapData.

The image object, for example a scanned page.

Width Parameter

Type: sx_double.

The desired width of a page, in points. It can be 0, in this case SDK use width of image.

Height Parameter

Type: sx_double.

The desired height of a page, in points. It can be 0, in this case SDK use height of image.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SaveAsStream Method

The method saves the ZUGFeRD document to the stream. We recommend to validate the format and data of XML invoice using IInvoice::Validate method before save.

Stream Parameter

Type: ISequentialStream.

The stream for save. Please reset stream before save.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

SaveAsFile Method

The method saves the ZUGFeRD document to the file. We recommend to validate the format and data of XML invoice using IInvoice::Validate method before save.

FilePath Parameter

Type: sx_str.

Fully qualified path of the file to save. Method creates a new file, always. If the specified file exists it must have write access.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IInvoiceStream Interface

The interfaces provides interface of ZUGFeRD invoice object (based on XML file or stream) implemented by SDK.

Windows: This interface is derived from IRefObject at this platform. We recommend you use an instance of an ObjPtr template class where you can use an ZUGFeRD::IInvoiceStream pointer in a safe manner.

Invoice Property

Type: IInvoice. Access: readonly.

The property retrieves interface which provides access to invoice data.

Open Method

The method loads the ZUGFeRD invoice from stream. The format and data of invoice can be validated using IInvoice::Validate method, we recommend to do it immediately after loading of invoice.

Stream Parameter

Type: ISequentialStream.

The XML formatted data stream. Please reset stream before load.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Save Method

The method saves the ZUGFeRD invoice in XML format to the stream. We recommend to validate the format and data of XML invoice using IInvoice::Validate method before save.

Stream Parameter

Type: ISequentialStream.

The stream for save. Please reset stream before save.

Errors

The method produces the exceptions if failed, see ErrorCode property of exception for details.

Reset Method

The method resets ZUGFeRD invoice in the initial state – remove any data if exist, than creates header and transaction objects.

IInvoice Interface

The interface represents ZUGFeRD invoice. The structure of properties, methods and all referenced interfaces correspond with technical specification of ZUGFeRD standard, please use it as additional and detailed reference for using SDK API.

Test Property

Type: sx_bool. Access: full.

The property marks the invoice as a "test" (if is true) and thus not leading to VAT issues.

Validate Method

The method provides many validation possibilities of XML format and invoice data. We recommend to call this method immediately after loading of invoice and/or before saving of invoice. It returns true if no critical errors were found.

Options Parameter

Type: sx_flags, see also zf_val_option enumeration.

The parameter specifies the options of validation. You can combine the enumeration values by using the bitwise OR operator.

Recommended value of this parameter:

- for incoming invoices = zf_val_option_schema | zf_val_option_rules | zf_val_option_codes
- for created and outgoing invoices = zf_val_option_rules

Status Parameter

Type: IValidationStatus.

The optional parameter sets a callback handler of validation process. The IValidationStatus interface must implemented by application to process warning or error events.

UWP / WinRT / WinPhone: At these platforms the IValidationStatus delegate is declared instead of callback interface. Please define an event handler method with the same signature as the **IValidationStatus** delegate and add an instance to the **OnMessage** event of the invoice object.

GetProfile Method

The method retrieves the profile of invoice. It returns **IProfileElement** or nullptr if element is not defined yet. Set "create" parameter to true if you want create not existing element.

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetHeader Method

The method retrieves the block of properties regarding the whole invoice. It returns **IHeader** element or nullptr if element is not defined yet. Set "create" parameter to true if you want create not existing element.

The method produces the exceptions if failed, see ErrorCode property of exception for details.

GetTransaction Method

The method retrieves the information of trade transaction covered by this invoice. It returns **ITradeTransaction** element or nullptr if element is not defined yet. Set "create" parameter to true if you want create not existing element.

The method produces the exceptions if failed, see ErrorCode property of exception for details.

IValidationStatus Interface / Delegate

The callback interface, you need to implement it for using in IInvoice::Validate method for monitoring of validation process.

Windows: At this platform you must implement **IValidationStatus** interface in your application than you can process error or warning messages.

DotNET: At this platform you must define an event handler method with the same signature as the **IValidationStatus** delegate than you can process error or warning messages.

UWP / WinRT / WinPhone: At this platform you must define an event handler method with the same signature as the **IValidationStatus** delegate than you can process error or warning messages.

OnMessage Method / Event

The method is called everytime when IInvoice::Validate procedure needs to inform your application about error or warning in process of validation. The method must return **true** to continue converting or **false** to break validation procedure.

Message Parameter

Type: IValidationMessage.

The properties of validation message.

IValidationMessage Interface

The interface provides properties of validation message.

CriticalError Property

Type: sx_bool. Access: readonly.

The property separates validation results as error and warning messages.

Code Property

Type: zf_val_code. Access: readonly.

The property retrieves coded type of problem.

ExtendedInfo Property

Type: sx_str. Access: readonly.

The property retrieves string with extended info for message, as rule it is XML node where a proble was found. It could null in some cases.

Enumerations

zf_status

The enumeration defines validation status of PDF part of ZUGFeRD invoice.

zf_status_not_pdf

The file (or stream) is not recognized as PDF/A-3 conform file, it is not PDF exactly.

zf_status_enc_pdf

The file (or stream) is not recognized as PDF/A-3 conform file because it is encrypted PDF.

zf_status_no_meta

The file (or stream) hasn't metadata or metadata is corrupted.

zf_status_err_meta

The metadata of file (or stream) isn't conform to ZUGFeRD requirements, possible reasons:

- PDF/A identification schema isn't present or has invalid version

- PDF/A extension schema isn't present or don't defines ZUGFeRD properly

- ZUGFeRD schema isn't present or contains invalid values

zf_status_not_pdfa

The PDF/A validation of PDF structure and objects is failed.

zf_status_err_emb

The XML invoice isn't embedded or embedded incorrectly.

zf_status_not_zf

The embedded XML can't be recognized as ZUGFeRD invoice.

zf_status_dif_profile

The profile of invoice is not equal to the invoice profile specified in metadata.

zf_status_valid

The valid ZUGFeRD invoice.

zf_val_option

The enumeration defines options for validation of ZUGFeRD invoice.

zf_val_option_namespace

The minimal level of validation, it is non-optional.

zf_val_option_schema

Validate the XML schema (common XML structure) according to ZUGFeRD format must be validated. This option is recommended for incoming invoices only.

zf_val_option_rules

Validate the required elements and attributes of ZUGFeRD invoice. This strongly recommended for validation of any invoices (incoming and outgoing).

zf_val_option_codes

Validation of code (and ID) values. This option is recommended for incoming invoices only.

zf_val_option_profile

Validation of accordance with specified profile. Usually it should not be used.

zf_val_code

The enumeration defines error or warning codes for validation procedure.

zf_val_code_schema

The XML schema does not conform to ZUGFeRD.

zf_val_code_attr_notused

The specified XML attribute can not be used in this context (see extended info).

zf_val_code_elem_notused

The specified XML element can not be used in this context (see extended info).

zf_val_code_attr_req

The specified XML attribute required but not present (see extended info).

zf_val_code_elem_req

The specified XML element required but not present (see extended info).

zf_val_code_elem_one_req

The specified XML element must occur exactly 1 times (see extended info).

zf_val_code_elem_one_opt

The specified XML element may occur at maximum 1 times (see extended info).

zf_val_code_value_req

The specified XML element must have a value (see extended info).

zf_val_code_unknown_code

The specified XML element has wrong value - unknown code or ID (see extended info).

zf_val_code_out_of_profile

The specified XML element or attribute is out of invoice profile (see extended info).

TECHNICAL SUPPORT

E-Mail: pdf@soft-xpansion.com