scConverter

SDK

Documentation

Version 9

Software Companions

http://www.softwarecompanions.com

IMPORTANT

Contact Software Companions:

sales@softwarecompanions.com

www.softwarecompanions.com

Trademarks

Acrobat	Registered trademark of Adobe Systems Inc.
AutoCAD	Registered trademark of Autodesk, Inc.
Microsoft Windows	Registered trademark of Microsoft Corp.
Microsoft Office	Registered trademark of Microsoft Corp.
Microsoft Excel	Registered trademark of Microsoft Corp.
Microsoft PowerPoint	Registered trademark of Microsoft Corp.
Microsoft Word	Registered trademark of Microsoft Corp.

All other brands and product names used in this document are trademarks of their respective companies.

Table of Contents

1 General Information

1.1 Welcome to scConverter

scConverter is a powerful component for converting plot files (PLT, HPGL, HPGL/2 and Calcomp), PDF, CGM, DWF, TIFF, PNG, JPEG, CALS and BMP files to PDF, DXF, DWF, TIF and many other formats.

You may use scConverter in any development environment that can use COM components, for example Visual Basic, Visual C++, C#, ASP, ASPX, NET, Delphi and much more.
The component may be used in environments not supporting COM as the DLL does export functions that can be called directly. This makes it easy to use the DLL even in a Java application. If you choose to use the exported functions directly you do not have to register the DLL.
This document describes each method, property and exported function available in this component.

1.2 Build your application with scConverter

scConverter contains more than 200 methods, properties and exported functions that may be used by your application. The component includes methods for merging, splitting and encrypting PDF files.

The control includes a feature named PDF to CAD which can be used to create a CAD file from a PDF file. With this feature you may convert an Adobe PDF file into, for example, a fully editable Autodesk DXF file.
When you convert your PDF file to DXF, all text in the PDF file will be converted into DXF MTEXT entities.
Any layer defined in the PDF file will be retained when converting to both DXF and DWF formats.

Use the powerful markup XML format to add text, stamps, watermarks, barcodes and much more to your converted files.

Included with the SDK you will find complete applications written in C# and C++/MFC with full source code available.

2 The scConverter Interface

2.1 scConverter Methods

2.1.1 AddGerberLayer

Add a Gerber, HPGL/2 or Excellon file as a new layer to use for conversion. You may use this method to convert multiple layers to a single file. When all required layers are added you may use either the ConvertGerberLayersToImage or the ConvertGerberLayersToImageEx function to create an image file combining all layers. You can create a vector file (e.g.: PDF, SVG or DXF) by using the ConvertGerberLayersToCAD function.

<table>
<tr><td>Syntax</td><td colspan="2">HRESULT AddGerberLayer(BSTR InputFile, OLE_COLOR Color, long Flags);</td></tr>
<tr><td>Parameters</td><td>InputFile</td><td>Name of the file to add as a layer. The file can either be local (UNC) or URL (http:// or https://).</td></tr>
<tr><td></td><td>Color</td><td>Color value to use for this Gerber layer during conversion. Color values are given as 0x00BBGGRR.</td></tr>
<tr><td></td><td>Flags</td><td>The following values are supported:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Set layer as transparent. Color values are in this case given as 0xAABBGGRR. Valid opacity range is from 0 to 255, where 255 is full opacity. Sample color value : 0x800000FF. This will set the color to full red (255) and an opacity value of 128 (0x80).</td></tr>
</table>
Values can be combined.</td></tr>
<tr><td>Returns</td><td>HRESULT</td><td>Returns S_OK if file is added successfully. Any other value indicates an error, see GetLastError for more information.</td></tr>
</table>

2.1.2 AddMarkupEx

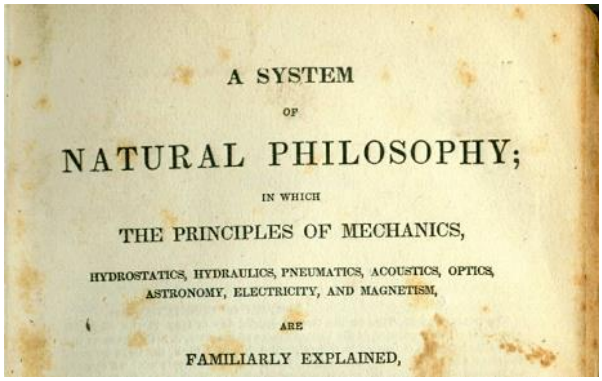
Add markup data to a file already opened by the OpenFileEx method.

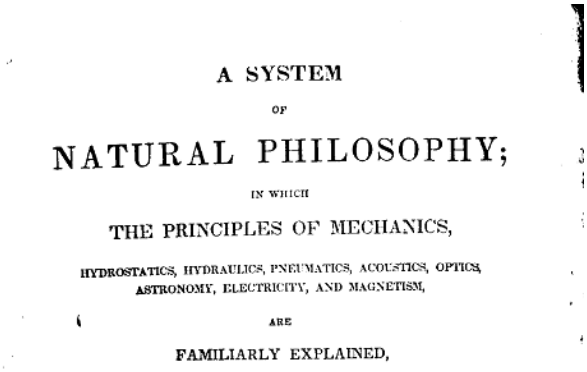
<table>
<tr><td>Syntax</td><td colspan="2">HRESULT AddMarkupEx(long Handle, BSTR Markup);</td></tr>
<tr><td>Parameters</td><td>Handle</td><td>Handle to file returned by OpenFileEx.</td></tr>
<tr><td></td><td>Markup</td><td>This may either be a filename that contains markup data, or it may be string containing markup XML data. See section 3.6 for a description of the markup XML format.</td></tr>
<tr><td>Returns</td><td>HRESULT</td><td>Returns S_OK if file is added successfully. Any other value indicates an error, see GetLastError for more information.</td></tr>
</table>

2.1.3 BinarizeImage

The binarize filter can convert a color image to a black and white image (1 bits per pixel). Like the DefoxImage method described below, it can also be used to

remove stain from old, scanned documents. The pictures below show an image before and after running binarize filter (Threshold value used is 0.5). The sample file used is also included in the SDK. This method can be used for raster image and PDF file formats.





Syntax	HRESULT BinarizeImage(BSTR InputFile, BSTR OutputFile, BSTR Format, double Threshold);	
Parameters	InputFile	Input file to process using the dexofing algorithm.
	OutputFile	Name of processed output file.
	Format	File format to use for output file. This string have to be one of the following: • AVIF • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Threshold	Threshold value to use for the binarization. Threshold values must be between 0 to 1.
Returns	HRESULT	Returns S_OK if successful. Any other value indicates an error, see GetLastError for more information.

2.1.4 BinarizeImageEx

The binarize filter can convert a color image to black and white image (1 bits per pixel). Please see the BinarizeImage method description above for more details.

Syntax	RESULT BinarizeImageEx(LONG Handle, LONG Page, BSTR OutputFile, BSTR Format, double Threshold);	
Parameters	Handle	Handle to a previously opened file using the OpenFilexEx method.
	Page	Page to process in input file. Pages start at index 0.
	OutputFile	Name of processed output file.
	Format	File format to use for output file. This string have to be one of the following: • AVIF

		• BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Threshold	Threshold value to use for the binarization. Threshold values must be between 0 to 1.
Returns	HRESULT	Returns S_OK if successful. Any other value indicates an error, see GetLastError for more information.

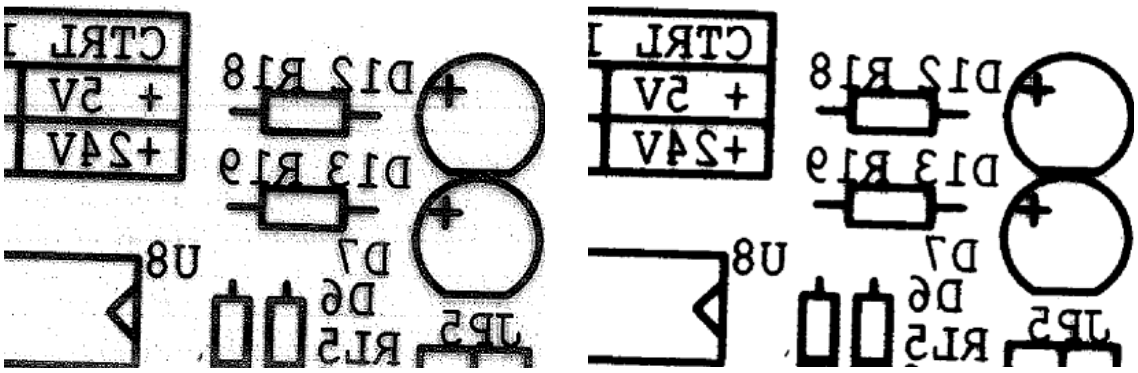
2.1.5 CheckFile

Check if the given file can be opened by the converter.

Syntax	HRESULT CheckFile(BSTR InputFile, long *Format);	
Parameters	InputFile	Name of the file to check.
	Format	One of the following values is returned: 0. Unknown file format 1. HPGL/2 2. TIFF 3. CALS 4. PNG 5. JPEG 6. Windows BMP 7. CGM (Computer Graphics Metafile) 8. Calcomp Plot Format 9. Autodesk DWF 10. Gerber Plot Format 11. Adobe PDF 12. WEBP Google Image Format 13. GIF Image Format 14. EDMICS Raster Format 15. Intergraph Raster Format 16. JPEG2000 17. Text Format 18. Word Format 19. Excel Format 20. Powerpoint Format 21. LibreOffice Format 22. Excellon Drill format 23. HEIC Image Format 24. PSD Image Format (Adobe Photoshop) 25. JPEG-XL Image Format 26. AVIF Image Format
Returns	HRESULT	Returns S_OK if this file can be opened by the converter.

2.1.6 CleanupImage

Remove noise from an image. This method is supported for raster image and PDF file formats. Below are two pictures that show an image before and after CleanupImage has been applied:



Syntax	HRESULT CleanupImage(BSTR InputFile, BSTR OutputFile, BSTR Format);	
Parameters	InputFile	Input image file that will be cleaned.
	OutputFile	Name of cleaned output file.
	Format	File format to use for output file. This string have to be one of the following: • AVIF • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.7 CleanupImageEx

Remove noise from an image. Please see the CleanupImage description above for more information.

Syntax	HRESULT CleanupImageEx(LONG Handle, LONG Page, BSTR OutputFile, BSTR Format);	
Parameters	Handle	Handle to a file previously opened using the OpenFileEx method.
	Page	Page number to process. Pages start at index 0.
	OutputFile	Name of cleaned output file.
	Format	File format to use for output file. This string have to be one of the following: • AVIF

		• BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.8 ClearGerberLayers

Unload all added layers added using the AddGerberLayer method from memory.

Syntax	HRESULT ClearGerberLayers();	
Parameters	None	
Returns	HRESULT	Returns S_OK if successful. Any other value indicates an error, see GetLastError for more information.

2.1.9 CloseFileEx

Close a file previously opened by OpenFileEx.

Syntax	HRESULT CloseFileEx(long Handle);	
Parameters	Handle	Handle for the file to close.
Returns	HRESULT	This function returns S_OK if the file was successfully closed.

2.1.10 CompressFile

Compress a file using the given compression method.

Syntax	HRESULT CompressFile(BSTR InputFile, BSTR OutputFile, LONG CompressionMethod);	
Parameters	InputFile	Full path to input file to compress.
	Outputfile	Full path to compressed output file.
	CompressionMethod	Compression method to use: 0 : GZIP compression method. 1 : Brotli compression method. 2 : BZIP2 compression method. 3 : ZSTD compression method.
Returns	HRESULT	This function returns S_OK if the file was successfully compressed.

2.1.11 Convert

Convert the input file to a new file using the selected format.

Syntax	HRESULT Convert(BSTR InputFileName, BSTR OutputFileName, BSTR Format, double Scale, long BitsPerPixel, long DPI);	
Parameters	InputFileName	Name of file to convert.
	OutputFileName	Name of destination file.
	Format	Select output format to use. The following formats are supported: • BMP (Windows Bitmap) • CALS (CALS Type 1 CCITT-G4 Format) • CGM (Computer Graphics Metafile) • DWF (Drawing Web Format) • DXF (AutoDesk Drawing Exchange Format) • EMF (Windows Enhanced Metafile) • GBR (Gerber RS-274X) • HEIC (High Efficiency Image Format – HEIF) • HPGL (HPGL/2) • HPRTL(HP-RTL) • JPEG (JFIF Compliant) • JPEG2K (JPEG 2000) • JXL (JPEG-XL) • PCX (Paintbrush Format) • PDF (Acrobat PDF) • PDFRASTER (Acrobat PDF raster). • PNG (Portable Network Graphics) • PS (Adobe Postscript) • SVG (Scalable Vector Format) • TIFF (Tagged Image File Format) • WEBP (Google WebP Image Format) • WMF (Windows Metafile)
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for raster format conversion, for example TIFF or PNG. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color For TIFF, HEIC, WEBP and PNG file formats you can use 32 bits to create images with an alpha channel.
	DPI	Resolution given in pixels per inch. This parameter is used only for raster format conversion, for example TIFF.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.12 Convert2

Convert the input file to a new file using the selected format. This is a special variant that allows rotation and offsets for the converted file. Please note that offsets will only work if the selected output format is a vector format (for example PLT and DXF).

Syntax	HRESULT Convert2(BSTR InputFileName, BSTR OutputFileName, BSTR Format, double Scale, double Rotation, double OffsetX, double OffsetY);	
Parameters	InputFileName	Name of file to convert.
	OutputFileName	Name of destination file.
	Format	Please see the Convert method description above for a list of supported formats.
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	Rotation	Rotation factor in degrees.
	OffsetX	Add an x offset value to the output file in millimeters.
	OffsetY	Add an y offset value to the output file in millimeters.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.13 Convert3

Convert the input file to a new file using the selected format. This is a special variant that let you set rotation for converted file.

Syntax	HRESULT Convert2(BSTR InputFileName, BSTR OutputFileName, BSTR Format, double Scale, double Rotation, double OffsetX, double OffsetY);	
Parameters	InputFileName	Name of file to convert.
	OutputFileName	Name of destination file.
	Format	Please see the Convert method description above for a list of supported formats.
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	Rotation	Rotation factor in degrees.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for raster format conversion, for example TIFF or PNG. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color For TIFF, HEIC, WEBP and PNG file formats you can also use 32 bits to create images with an

		alpha channel.
	DPI	Resolution given in pixels per inch. This parameter is used only for raster format conversion, for example TIFF.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.14 ConvertFileEx

Convert a single page, or all pages, from an already opened file. into a new file using the selected output format.

Syntax	HRESULT ConvertFileEx(long Handle, BSTR OutputFileName, BSTR Format, long Page, double Scale, long BitsPerPixel, long DPI);	
Parameters	Handle	Handle to file returned by OpenFileEx.
	OutputFileName	Name of destination file.
	Format	Select the file format to use for the output file. Please see the Convert method above for a list of available formats.
	Page	Index of the page to extract from the input file, and convert to a new output file. First page number is 0 (zero indexed). Set to -1 to convert all pages in the document..
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	BitxPerPixel	Bits Per Pixel. This parameter is used only for raster format conversion, for example TIFF. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color For TIFF, HEIC, WEBP and PNG file formats you can use 32 bits to create images with an alpha channel.
	DPI	Resolution given in pixels per inch. This parameter is used ony for raster format conversion, for example TIFF.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.15 ConvertFileEx2

Convert a single page, or all pages, from an already opened file. into a new file using the selected output format. This method is equal to ConvertFileEx but also allows you to rotate the output file.

Syntax	HRESULT ConvertFileEx(long Handle, BSTR OutputFileName, BSTR Format, long Page, double Scale, double Rotation, long BitsPerPixel, long DPI);	
Parameters	Handle	Handle to file returned by OpenFileEx.
	OutputFileName	Name of destination file.
	Format	Select the file format to use for the output file. Please see the Convert method above for a list of available formats.
	Page	Index of the page to extract from the input file, and convert to a new output file. First page number is 0 (zero indexed). Set to -1 to convert all pages in the document..
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	Rotation	Rotation of output file given in degrees. Please note that only 0, 90, 180 and 270 degrees is currently supported.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for raster format conversion, for example TIFF. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color For TIFF, HEIC, WEBP and PNG file formats you can use 32 bits to create images with an alpha channel.
	DPI	Resolution given in pixels per inch. This parameter is used ony for raster format conversion, for example TIFF or PNG.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.16 ConvertGerberLayersToCAD

Convert all loaded Gerber and Excellon layers to a CAD formatted output file.
To be able to use this method one or more Gerber or Excellon files must have previously been added as layers using the AddGerberLayer method.

Syntax	HRESULT ConvertGerberLayersToCAD(BSTR OutputFile, BSTR Format);	
Parameters	OutputFile	Name of the destination file.
	Format	Select the output format to use. The following formats are supported by this method: • CGM (Computer Graphics Metafile) • DWF (Drawing Web Format) • DXF (AutoDesk Drawing Exchange Format) • GBR (Gerber RS274X) • HPGL(HPGL/2) • PDF (Acrobat PDF)

		• PS (Adobe Postscript) • SVG (Scalable Vector Format)
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.17 ConvertGerberLayersToMP

Convert all loaded Gerber and Excellon layers to a multipage file, one layer per page. To be able to use this method one or more Gerber or Excellon files must have previously been added as layers using the AddGerberLayer method.

Syntax	HRESULT ConvertGerberLayersToMP(BSTR OutputFile, BSTR Format, LONG BitsPerPixel, LONG DPI);	
Parameters	OutputFile	Name of the destination file.
	Format	Select the output format to use. The following formats are supported by this method: • DWF (Drawing Web Format) • HPGL(HPGL/2) • PDF (Acrobat PDF) • TIFF (Tagged Image File Format)
	BitsPerPixel	Number of colors to used for TIFF output. Supported values: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color
	DPI	Resolution to use for TIFF output given in pixels per inch.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.18 ConvertGerberLayersToImage

Convert all loaded Gerber layers to a new image file of the given file format.
To be able to use this method one or more Gerber files must have previously been added as layers using the AddGerberLayer method. Use the BackgroundColor property to change the image background color.
The image width and height will automatically be calculated based on the input file and given scale factor.

Syntax	HRESULT ConvertGerberLayersToImage(BSTR OutputFile, BSTR Format, double ScaleFactor, long BitsPerPixel, long DPI, long Flags, double *ImageW, double *ImageH);	
Parameters	OutputFile	Name of the destination file.
	Format	Select the output format to use. The following formats are supported by this method: • BMP (Windows Bitmap) • CALS (CALS Type 1 CCITT-G4 Raster Format) • HEIC

		• JPEG (JFIF Compliant) • JXL (JPEG-XL) • PCX (Paintbrush Format) • PDF (Acrobat PDF) • PDFRASTER (Acrobat PDF raster). • PNG (Portable Network Graphics) • TIFF (Tagged Image File Format) • WEBP
	ScaleFactor	Scalefactor to use for conversion, 1.0 is original size.
	BitsPerPixel	Bits Per Pixel. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color
	DPI	Resolution given in pixels per inch.
	Flags	The following values are supported: **Value** | **Description** 1 | Add margins to image. The margins can be defined by using SetMargins function. Values can be combined.
	ImageW	Returns the width of the created image in millimeters.
	ImageH	Returns the height of the created image in millimeters.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.19 ConvertGerberLayersToImageEx

Convert all loaded Gerber layers to an image. To be able to use this method you will need to add one or more Gerber layers using the AddGerberLayer method. Use the BackgroundColor property to change the image background color. The image width and height will automatically be calculated based on the input files and given scale factor.

Syntax	HRESULT ConvertGerberLayersToImageEx(double ScaleFactor, long BitsPerPixel, long DPI, long Flags, IPictureDisp** Picture);	
	ScaleFactor	Scalefactor to use for conversion, 1.0 is original size.
	BitsPerPixel	Bits Per Pixel. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color
	DPI	Resolution given in pixels per inch.
	Flags	The following values are supported: **Value** | **Description** 1 | Add margins to image. The margins can be defined by using SetMargins function. Values can be combined.
	Picture	The returned image (IDispatch)
Returns	HRESULT	Returns S_OK if image was created. Any other value indicates an error, see GetLastError for more information.

2.1.20 ConvertMarkup

Convert the input file together with markup data to a new file using the given format.

Syntax	HRESULT ConvertMarkup(BSTR InputFileName, BSTR Markup, BSTR OutputFileName, BSTR Format, double Scale, long BitsPerPixel, long DPI);	
Parameters	InputFileName	Name of file to convert.
	Markup	This may either be a filename that contains markup data, or it may be string containing markup XML data. See section 3.6 for description of the Markup XML format.
	OutputFileName	Name of destination file.
	Format	Select the file format to use for the output file. Please see the Convert method above for a list of available formats.
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for raster format conversion, for example TIFF or PNG. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color
	DPI	Resolution given in pixels per inch. This parameter is used ony for raster format conversion, for example TIFF or PNG.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.21 ConvertPage

Convert a single page from the input file into a new file using the given format.

Syntax	HRESULT ConvertPage(BSTR InputFileName, BSTR OutputFileName, BSTR Format, long Page, double Scale, long BitsPerPixel, long DPI);	
Parameters	InputFileName	Name of file to convert.
	OutputFileName	Name of destination file.
	Format	Select the file format to use for the output file. Please see the Convert method above for a list of available formats.
	Page	Index of the page to extract from the input file, and convert to a new output file. First page number is 0 (zero indexed).
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for

		raster format conversion, for example TIFF. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color
	DPI	Resolution given in pixels per Inch. This parameter is used ony for raster format conversion, for example TIFF or PNG.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.22 ConvertPageToImage

Convert a single page from the input file into a memory image (IPicture).

Syntax	HRESULT ConvertPageToImage(BSTR InputFileName, long Page, double Scale, long BitsPerPixel, long DPI, IPictureDisp** Picture);	
Parameters	InputFileName	Name of file to convert.
	Page	Index of the page to convert into an image. First page number is 0 (zero indexed).
	ScaleFactor	Scalefactor to use for conversion, 1.0 is original size.
	BitsPerPixel	Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color
	DPI	Resolution given in pixels per Inch.
	Picture	The returned image (IDispatch)
Returns	HRESULT	Returns S_OK if image is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.23 ConvertPageToImageEx

Convert a single page from a previously opened file into a memory image (IPicture).

Syntax	HRESULT ConvertPageToImage(long Handle, long Page, double Scale, long BitsPerPixel, long DPI, IPictureDisp** Picture);	
Parameters	Handle	Handle to file returned by OpenFileEx.
	Scale	Scale factor to apply, use 1.0 for original size.
	Page	Index of the page to convert into an image. First page number is 0 (zero indexed).
	ScaleFactor	Scalefactor to use for conversion, 1.0 is original size.
	BitsPerPixel	Supported values for BitsPerPixel:

		1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color
	DPI	Resolution given in pixels per Inch.
	Picture	The returned image (IDispatch)
Returns	HRESULT	Returns S_OK if image is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.24 ConvertSVGToImage

Convert the input SVG file to an image that will be returned as an IPicture.
.

Syntax	HRESULT ConvertSVGToImage(BSTR InputFile, LONG MaxSize, IPictureDisp** Picture);	
Parameters	InputFile	Name of input file to convert.
	MaxSize	Maximum width or height of the image. Aspect ratio will be maintained. Set this parameter to 0 if you want to use the original SVG dimensions.
	Picture	The returned image (IDispatch).
Returns	HRESULT	Returns S_OK if image was created. Any other value indicates an error, see GetLastError for more information.

2.1.25 ConvertSVGToImageFile

Convert the input SVG file to a new image file of the given file format.

Syntax	HRESULT ConvertSVGToImageFile(BSTR InputFile, BSTR OutputFile, BSTR Format, LONG MaxSize, LONG DPI);	
Parameters	InputFile	Name of input file to convert.
	OutputFile	Name of destination file.
	Format	The supported output formats are: • AVIF (AV1 Image File Format) • BMP (Windows Bitmap) • HEIC (High Efficiency Image File Format) • JPEG (JFIF Compliant) • JXL (JPEG-XL) • PDF (Acrobat PDF) • PNG (Portable Network Graphics) • TIFF (Tagged Image File Format) • WEBP (Google Image Format)
	MaxSize	Maximum width or height of the image. Aspect ratio will be maintained. Set this parameter to 0 if you want to use the original SVG dimensions.
	DPI	Resolution given in pixels per Inch.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.26 ConvertToImageFile

Convert the input file to a new image file of the given file format.
This method provides better control of the created image than the other conversion methods, which may be very useful for Gerber to image conversion.

Syntax	HRESULT ConvertToImageFile(BSTR InputFile, BSTR OutputFile, BSTR Format, long Page, double ScaleFactor, long ImageWidth, long ImageHeight, long BitsPerPixel, long DPI, long Flags, double OffsetX, double OffsetY);	
Parameters	InputFile	Name of input file to convert.
	OutputFile	Name of destination file.
	Format	The supported output formats are: • AVIF (AV1 Image File Format) • BMP (Windows Bitmap) • CALS (CALS Type 1 CCITT-G4 Raster Format) • HEIC (High Efficiency Image File Format) • JPEG (JFIF Compliant) • JXL (JPEG-XL) • PCX (Paintbrush Format) • PDF (Acrobat PDF) • PDFRASTER (Acrobat PDF raster). • PNG (Portable Network Graphics) • TIFF (Tagged Image File Format) • WEBP (Google Image Format)
	ScaleFactor	Scalefactor to use for conversion, 1.0 is original size.
	ImageWidth	Width of the destination image file in pixels.
	ImageHeight	Height of the destination image file in pixels.
	BitsPerPixel	Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color
	DPI	Resolution given in pixels per Inch.
	Flags	Following flag bits are supported: 1. The original Gerber offsets will be used during conversion. 2. The output file will be scaled to fit the given output size.
	OffsetX	Left offset in pixels.
	OffsetY	Top offset in pixels.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.27 ConvertToPaperSize

Convert the input file to a new file using the given format and selected paper size.

Syntax	ConvertToPaperSize(BSTR InputFile, BSTR OutputFile, BSTR Format, Long PaperSizeIndex, long KeepPaper, long Page, long BitsPerPixel, long DPI);	
Parameters	InputFile	Name of input file to convert.
	OutputFile	Name of destination file.
	Format	Select the file format to use for the output file. Please see the Convert method above for a list of available formats.
	PaperSizeIndex	The index of the predefined paper size to use for conversion. The output file will be scaled to fit the selected paper size. Please see appendix D for a list of supported paper formats.
	KeepPaper	If enabled margins will be added if necessary, to make sure the output use the exact paper size.
	Page	Index of the page to extract from the input file, and convert to a new output file. First page number is 0 (zero indexed). Set to -1 to convert all pages in the document..
	BitsPerPixel	Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color This parameter is only used for raster format conversion.
	DPI	Resolution given in pixels per Inch. This parameter is used ony for raster format conversion.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.28 ConvertToPaperSizeEx

Convert an already opened file to a new file using the given format and selected paper size.

Syntax	ConvertToPaperSizeEx(long Handle, BSTR OutputFile, BSTR Format, Long PaperSizeIndex, long KeepPaper, long Page, long BitsPerPixel, long DPI);	
Parameters	Handle	Handle returned by OpenFileEx.
	OutputFile	Name of destination file.
	Format	Select the file format to use for the output file. Please see the Convert method above for a list of available formats.
	PaperSizeIndex	The index of the predefined paper size to use for conversion. The output file will be scaled to fit the selected paper size. Please see appendix D for a list of supported paper

		formats.
	KeepPaper	If enabled margins will be added if necessary, to make sure the output use the exact paper size.
	Page	Index of the page to extract from the input file, and convert to a new output file. First page number is 0 (zero indexed). Set to -1 to convert all pages in the document..
	BitsPerPixel	Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color This parameter is used ony for raster format conversion.
	DPI	Resolution given in pixels per Inch. This parameter is used ony for raster format conversion.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.29 ConvertToPDF

Convert the input file into a PDF file. Accepts all supported input formats.

Syntax	HRESULT ConvertToPDF(BSTR InputFileName, BSTR OutputFileName)	
Parameters	InputFileName	Name of file to convert.
	OutputFileName	Name of destination PDF file.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.30 ConvertToStream

Convert the input file to a new file using the given format and write it to the provided IStream.

Syntax	HRESULT Convert(BSTR InputFileName, IStream *OutputStream, BSTR Format, double Scale, long BitsPerPixel, long DPI);	
Parameters	InputFileName	Name of the file to convert.
	OutputStream	IStream interface pointer to use for output.
	Format	Select the file format to use for the output stream. Please see the Convert method above for a list of available formats.
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for raster format conversion, for example TIFF or PNG. Supported values for BitsPerPixel: 1. Monochrome, black & white

		4. 16 Colors 8. 256 Colors 24. True Color
	DPI	Resolution given in pixels per inch. This parameter is used ony for raster format conversion.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error (GetLastError).

2.1.31 CreateGUID

Create and return a GUID (globally unique identifier) string.

Syntax	HRESULT CreateGUID(BSTR *GUID);	
Parameters	GUID	Returned GUID.
Returns	HRESULT	Returns S_OK if successful.

2.1.32 CreateSearchablePDF

Create a searchable PDF file from a raster file, for example a scanned PDF file. You can create a searchable PDF file from all supported raster formats which include PDF, TIFF, PNG, JPEG and more. This method is only available if the Tesseract OCR engine is installed on the system. See appendix H for more information about Tesseract. You can use the **OCRAvailable** property to check if Tesseract is installed on the system.

Syntax	HRESULT CreateSearchablePDF(BSTR InputFileName, BSTR OutputFileName);	
Parameters	InputFileName	Name of input file
	OutputFileName	Name of destination PDF file with searchable text.
Returns	HRESULT	Returns S_OK if file is created successfully. Any other value indicates an error, see GetLastError for more information.

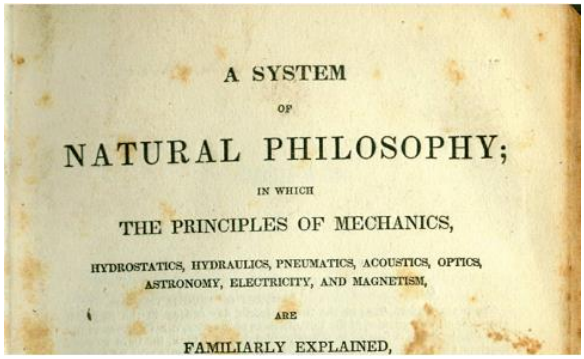
2.1.33 DetectQREx

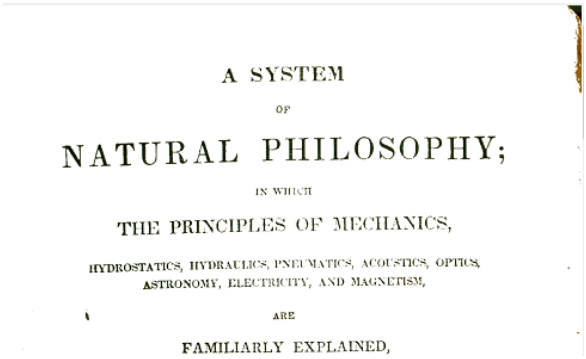
Search for QR codes in a file opened by OpenFileEx. You can search for QR images in a single page or all pages in the opened file. Use the GetQRTextEx method to return the decoded text for each found QR code.

Syntax	HRESULT DetectQREx(LONG Handle, LONG Page, LONG* QRCount);	
Parameters	Handle	Handle to a file previously opened by OpenFileEx.
	Page	Search for QR's in the given Page. Page indexes start at zero. Set this value to -1 to search all pages in the file.
	QRCount	Number of QR codes found.
Returns	HRESULT	Returns S_OK if successful.

2.1.34 DefoxImage

The defox image filter can be used to remove stain from old, scanned documents. A stained image is displayed to the left below and an image that has been defoxed to the right. For this example, a threshold value of 0.45 was used. The sample image file is included in the SDK. This method is supported for raster image and PDF file formats.





Syntax	HRESULT DefoxImage(BSTR InputFile, BSTR OutputFile, BSTR Format, double Threshold);	
Parameters	InputFile	Input file to process using the dexofing algorithm.
	OutputFile	Name of processed output file.
	Format	File format to use for output file. This string have to be one of the following: • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Threshold	Threshold value to use for the destaining. Threshold values must be between 0 to 1.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.35 DefoxImageEx

Remove stain from old, scanned documents. See the DefoxImage description above for more information.

Syntax	HRESULT DefoxImageEx(LONG Handle, LONG Page, BSTR OutputFile, BSTR Format, double Threshold);	
Parameters	Handle	Handle to a file previously opened using the OpenFileEx method.
	Page	Page number to process. Pages start at index 0.
	OutputFile	Name of processed output file.
	Format	File format to use for output file. This string have to be one of the following:

		• BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Threshold	Threshold value to use for the destaining. Threshold values must be between 0 to 1.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.36 DeskewImage

Deskew, or straighten, a skewed image by the given angle. If you pass an angle value of 0.0, the component will calculate the optimal angle before processing the image. This method is supported for raster image and PDF file formats.

Syntax	HRESULT DeskewImage(BSTR InputFile, BSTR OutputFile, BSTR Format, double Angle);	
Parameters	InputFile	Input file that will be deskewed.
	OutputFile	Name of processed output file.
	Format	File format to use for output file. This string have to be one of the following: • AVIF • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Angle	Deskew angle in degrees.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.37 DeskewImageEx

Deskew, or straighten, a skewed image by the given angle. If you pass an angle value of 0.0, the component will calculate the optimal angle before processing the image. This method is only supported for raster image file formats.

Syntax	HRESULT DeskewImageEx(LONG Handle, LONG Page, BSTR OutputFile, BSTR Format, double Angle);	
Parameters	Handle	Handle to a file previously opened using the OpenFileEx method.
	Page	Page number to process. Pages start at index 0.

	OutputFile	Name of processed output file.
	Format	File format to use for output file. This string have to be one of the following: • AVIF • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Angle	Deskew angle in degrees.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.38 DrillFormatSettings

Change the settings needed to load Excellon drill files correctly. Excellon files usually do not include any information about the internal format used. To be sure that such files are loaded correctly you may need to change the format settings before loading the file.

Syntax	HRESULT DrillFormatSettings(VARIANT_BOOL Incremental, long Preceding, long Succeeding, long Units, VARIANT_BOOL Leading, VARIANT_BOOL Trailing);	
Parameters	Incremental	If true the drill files contains incremental coordinates. Set to false for absolute coordinates.
	Preceding	Number of digits before decimal point.
	Succeeding	Number of digits after decimal point.
	Units	Units used in drill file. Set to 0 for inch and 1 for milimetres.
	Leading	Leading zeros suppressed if set to true.
	Trailing	Trailing zeros suppressed if set to true.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.39 GerberFormatSettings

Configure the settings that are needed to load RS-274D Gerber files correctly.

Syntax	HRESULT GerberFormatSettings(VARIANT_BOOL Incremental, long Preceding, long Succeeding, long Units, VARIANT_BOOL Leading, VARIANT_BOOL Trailing);	
Parameters	Incremental	If true the Gerber files contains incremental coordinates. Set to false for absolute coordinates.
	Preceding	Number of digits before decimal point.
	Succeeding	Number of digits after decimal point.

	Units	Units used for coordinates. Set to 0 for inch and 1 for milimetres.
	Leading	Leading zeros are suppressed if set to true.
	Trailing	Trailing zeros are suppressed if set to true.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.40 GetDeskewAngleEx

Return the calculated deskew angle for the given file and page. This method is only supported for raster image file formats.

Syntax	HRESULT GetDeskewAngleEx(LONG Handle, LONG Page, DOUBLE *Angle);	
Parameters	Handle	Handle returned by OpenFileEx.
	Page	Index of the page to query. Pages start at index 0.
	Angle	Returned calculated deskew angle in degrees.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.41 GetFileDimensions

Return the dimensions of the given page in the given file. All values are in millimeters.

Syntax	HRESULT GetFileDimensions(BSTR InputFile, long Page, double *OffsetX, double *OffsetY, double *Width, double *Height);	
Parameters	InputFile	Full path to the file to inspect.
	Page	Index of the page to query. First page number is 0 (zero indexed).
	OffsetX	Returned original file x offset of the selected page.
	OffsetY	Returned original file y offset of the selected page.
	Width	Returned width of the selected page.
	Height	Returned height of the selected page.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.42 GetFileDimensionsEx

Return dimensions of a page in an already opened file. All values are in millimeters.

Syntax	HRESULT GetFileDimensionsEx(long Handle, long Page, double *OffsetX, double *OffsetY, double *Width, double *Height);	
Parameters	Handle	Handle returned by OpenFileEx.
	Page	Index of the page to query. First page number is 0 (zero indexed).

	OffsetX	Returned original file x offset of the selected page.
	OffsetY	Returned original file y offset of the selected page.
	Width	Returned width of the selected page.
	Height	Returned height of the selected page.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.43 GetFormatEx

Return file format for a file previously opened with OpenFileEx.

Syntax	HRESULT GetFormatEx(long Handle, long *Format);	
Parameters	Handle	Handle returned by OpenFileEx.
	Format	Returned file format identifier. One of the following values is returned: 1. HPGL/2 2. TIFF 3. CALS 4. PNG 5. JPEG 6. Windows BMP 7. CGM (Computer Graphics Metafile) 8. Calcomp Plot Format 9. Autodesk DWF 10. Gerber Plot Format 11. Adobe PDF 12. WEBP Google Image Format 13. GIF Image Format 14. EDMICS Raster Format 15. Intergraph Raster Format 16. JPEG2000 17. Text Format 18. Word Format 19. Excel Format 20. Powerpoint Format 21. LibreOffice Format 22. Excellon Drill format 23. HEIC Image Format 24. PSD Image Format (Adobe Photoshop) 25. JPEG-XL Image Format 26. AVIF Image Format
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.44 GetNumFilePages

Return the number of pages in the given file.

Syntax	HRESULT GetNumFilePages (BSTR InputFile, long *Pages);	
Parameters	InputFile	Full path to the file to inspect.
	Pages	Returned number of pages in the file.

Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.45 GetNumFilePagesEx

Return the number of pages in an already opened file.

Syntax	HRESULT GetFileDimensionsEx(long Handle, long *Pages);	
Parameters	InputFile	Handle returned by OpenFileEx.
	Pages	Returned number of pages in the file.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.46 GetPageThumbnailEx

Returns a thumbnail image for a page in an already opened file.
You can also look at the C# scPDFMerge sample too see how this method may be used in an application.

Syntax	LONG GetPageThumbnailEx(long Handle, long Page, long Width, ' Long Height, long BitsPerPixel, IPictureDisp** Thumbnail);	
Parameters	Handle	Handle returned by OpenFileEx.
	Page	Page number to create a thumbnail of (first page is index 0)
	Width	Width of thumbnail image in pixels.
	Height	Height of thumbnail image in pixels.
	BitsPerPixel	The bits per pixel controls number of colors to use in the thumbnail image. Set this value to 1 for a monochrome (black & white) image, or 24 for a true color image.
	Thumbnail	The returned image (IDispatch).
Returns	HRESULT	Returns S_OK if image is created successfully. Any other value indicates an error, see GetLastError for more information.

2.1.47 GetPenTableEntry

Return width, color and style for the given pen table entry.
Please see the SetPenTableEntry method for description of the returned values.

Syntax	HRESULT GetPenTableEntry(long Pen, double Width, long Color, long LineStyle, long EndStyle);	
Parameters	Pen	Pen number.
	Width	Returned pen width.
	Color	Returned pen color.
	LineStyle	Returned pen linestyle.
	EndStyle	Returned pen line ending style.

Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.48 GetQRImage

Create and return a QR barcode image based on input settings.

Syntax	HRESULT GetQRImage(BSTR Text, LONG Size, LONG Margin, LONG ErrorLevel, IPictureDisp** Picture);	
Parameters	Text	Text string to encode as QR.
	Size	Set to 1 for default size. Setting size to 2 will create an image double as large.
	Margin	Optional white margin around QR image in pixels.
	ErrorLevel	Error level to use for QR generation.Default is 0.
	Picture	The image will returned as a IPicture dispatch pointer.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.49 GetQRTextEx

Return the decoded text for a QR code previously found using the DetectQREx method.

Syntax	HRESULT GetQRTextEx(LONG Handle, LONG Index, BSTR* Text);	
Parameters	Handle	Handle to file opened using OpenFileEx.
	Index	Index of QR to return text for. QR indexes start at 0 (zero).
	Text	Returned text for given QR code.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.50 GetRasterFileInfoEx

Return dimensions, bits per pixel and resolution of selected page in an already opened file.

Syntax	HRESULT GetFileRasterInfoEx(LONG Handle, LONG Page, LONG *Width, LONG *Height, LONG *BitsPerPixel, LONG *DPI);	
Parameters	Handle	Handle returned by OpenFileEx.
	Page	Index of the page to query. First page number is 0 (zero indexed).
	Width	Width of raster page in pixels.
	Height	Height of raster page in pixels.
	BitsPerPixel	Returned bits per pixel used for selected page.
	DPI	Returned resolution for selected page.
Returns	HRESULT	Returns S_OK if success. Any other value indicates

		an error, see GetLastError for more information.

2.1.51 LoadPenTable

Load a ViewCompanion compatible pen table file.

Syntax	HRESULT LoadPenTable(BSTR FileName);	
Parameters	FileName	Name of pentable file to load.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.52 OpenFileEx

Open a file and keep it open for as long as needed.
This method will return a handle that may be uses by other methods to perform different tasks. When you've done with the file it should be closed by using the CloseFileEx method.

Syntax	HRESULT OpenFileEx(BSTR FileName, LONG* Handle);	
Parameters	FileName	Full path name of file to open
	Handle	Returned file handle.
Returns	HRESULT	Returns S_OK if file was successfully opened. Any other value indicates an error, see GetLastError for more information.

2.1.53 PDFConform

Conform, or convert, an existing PDF file to PDF/A standard.

Syntax	HRESULT PDFComform(BSTR InputFile, SC_PDF_CONFORM ConformType, BSTR OutputFile, BSTR *Errors);	
Parameters	InputFile	The PDF file you want to conform into PDF/A.
	ConformType	The conformation standard to use: **Value** / **Standard** 0 Normalize 1 PDF/A-1b 2 PDF/A-2b 3 PDF/A-3b 4 PDF/A-2u 5 PDF/A-3u 6 PDF/A-4 7 PDF/A-4e
	OutputFile	Conformed output file.
	Errors	Will contain error message(s) if the conformation fail.
Returns	HRESULT	Returns S_OK if output file was successfully conformed. Any other value indicates an error, see GetLastError for more information.

If the parameter ConformType is set to Normalize the function checks the PDF file for errors, rebuilds all embedded fonts, optionally embeds non-embedded fonts, and repairs potential file errors if possible. The resulting PDF file is easy to read and should not produce any printing error. Normalization is useful in print workflows to detect potential errors before the file will be printed.

2.1.54 PDFEncrypt

Encrypt a PDF file using password(s) and optional restriction settings.

<table>
<tr><td>Syntax</td><td colspan="2">HRESULT PDFEncrypt(BSTR OriginalFile, BSTR EncryptedFile, BSTR OpenPassword, BSTR OwnerPassword, LONG Restrictions);</td></tr>
<tr><td>Parameters</td><td>OriginalFile</td><td>The PDF file you want to create an encrypted copy of.</td></tr>
<tr><td></td><td>EncryptedFile</td><td>The encrypted PDF file. This will be an exact copy of the orignal file but encrypted.</td></tr>
<tr><td></td><td>OpenPassword</td><td>Optional password required to open the file.</td></tr>
<tr><td></td><td>OwnerPassword</td><td>Optional owner password.</td></tr>
<tr><td></td><td>Restrictions</td><td>Set optional user restrictions for the encrypted PDF file. The following values are available:
<table>
<tr><th>Value</th><th>Restriction</th></tr>
<tr><td>0</td><td>No restrictions.</td></tr>
<tr><td>4</td><td>Deny printing.</td></tr>
<tr><td>8</td><td>Deny modification of contents.</td></tr>
<tr><td>16</td><td>Deny copying of contents.</td></tr>
<tr><td>32</td><td>Deny commenting.</td></tr>
<tr><td>3900</td><td>Deny all.</td></tr>
</table>
The flags can be combined, for example you may set restrictions to 20 (4+16) to deny both printing and copying.</td></tr>
<tr><td>Returns</td><td>HRESULT</td><td>Returns S_OK if output file was successfully enctypted. Any other value indicates an error, see GetLastError for more information.</td></tr>
</table>

2.1.55 PDFExtractPageEx

Extract a single page from a PDF to a new PDF file. By using this method all the original PDF content will be maintained.

Syntax	HRESULT LONG Handle, LONG Page, BSTR OutputFile);	
Parameters	Handle	Handle returned by OpenFileEx.
	Page	Page to extract, first page is index 0.
	OutputFile	Name of output PDF file.
Returns	HRESULT	Returns S_OK if output file was successfully enctypted. Any other value indicates an error, see GetLastError for more information.

2.1.56 PDFFlattenAnnotations

Flatten either all or some annotations and create a new PDF file.

<table>
<tr><td>Syntax</td><td colspan="2">HRESULT PDFFlattenAnnotations (BSTR InputFile, BSTR OutputFile, LONG Flags);</td></tr>
<tr><td>Parameters</td><td>InputFile</td><td>The original PDF file with annotations that you want to create a flattened copy of.</td></tr>
<tr><td></td><td>OutputFile</td><td>The new flattened PDF file.</td></tr>
<tr><td></td><td>Flags</td><td>Control how annotations should be flattened.
The following flattening flag values are available:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>All annotations which have an appearance stream and which have the print flag set are flattened.</td></tr>
<tr><td>1</td><td>Only visible annotations will be flattened.</td></tr>
<tr><td>2</td><td>Only markup annotations are flattened. Link, Sound, or FileAttach annotations are not markup annotations and such types will be left intact.</td></tr>
<tr><td>4</td><td>Flatten all annotations which are not supported in PDF/A 1.</td></tr>
<tr><td>8</td><td>Flatten all annotations which are not supported in PDF/A 2 or 3.</td></tr>
<tr><td>16</td><td>Form fields will be flattened.</td></tr>
<tr><td>4096</td><td>Link annotations will be kept.</td></tr>
<tr><td>8192</td><td>File attachment annotations will be kept.</td></tr>
<tr><td>16384</td><td>Text annotations will be kept.</td></tr>
</table>
The flags may be combined.</td></tr>
<tr><td>Returns</td><td>HRESULT</td><td>Returns S_OK if output file was successfully flattened. Any other value indicates an error, see GetLastError for more information.</td></tr>
</table>

2.1.57 PDFForceOrientation

Create a copy of the input PDF file where all pages are forced (if required) to the given orientation. Pages that already are using the given orientation will not be modified. The selected pages, defined by the Pages parameter, will be rotated using the given rotation factor.

<table>
<tr><td>Syntax</td><td colspan="2">HRESULT PDFForceOrientation(BSTR InputFile, BSTR OutputFile, BSTR Pages, LONG Orientation);</td></tr>
<tr><td>Parameters</td><td>InputFile</td><td>Full path name for input PDF file that will be modified and copied to the output file.</td></tr>
<tr><td></td><td>OutputFile</td><td>Full path name for rotated output file.</td></tr>
<tr><td></td><td>Pages</td><td>String describing the pages that are to be modified. Set string to empty or use "-1" for all pages. Specify page indices as comma-separated values or ranges to process (e.g. "1, 2,3,7" or "1, 2, 3-7").</td></tr>
</table>

		For this function the first-page index is 1.
	Orientation	The selected pages will be be forced to the given orientation. The orientation is set to either Portrait or Landscape.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.58 PDFMergeAddFile

Add a file to the currently merged PDF file. If you pass a file that is not a PDF file, but using another of the supported formats, it will be converted to PDF before being merged.
All pages from the given PDF file will be added to the merged output file. PDFMergeInit must be called before any PDF file is added.

Syntax	HRESULT PDFMergeAddFile(BSTR PDFFileName);	
Parameters	PDFFileName	Full path name to the PDF file to add to the merged PDF.
Returns	HRESULT	Returns S_OK if file was successfully added. Any other value indicates an error, see GetLastError for more information.

2.1.59 PDFMergeAddFileEx

Add a file to the currently merged PDF file. If you pass a file that is not a PDF file, but using another of the supported formats, it will be converted to PDF before being merged.
The PageInformation parameter controls which pages from the given PDF file that will be added to the merged output file. PDFMergeInit must be called before any PDF file is added.

Syntax	HRESULT PDFMergeAddFileEx(BSTR PDFFileName, BSTR PageInformation);	
Parameters	PDFFileName	Full path name to the PDF file to add to the merged PDF.
	PageInformation	Control which pages that should be imported from the given PDF file. Use ";" to separate pages, e.g.: setting PageInformation to "1,2,10,11" will import pages 1,2,10 and 11 and add them to the merged PDF file.
Returns	HRESULT	Returns S_OK if file was successfully added. Any other value indicates an error, see GetLastError for more information.

2.1.60 PDFMergeClose

Close the currently merged PDF file and output all pages to given file name.

Syntax	HRESULT PDFMergeClose(BSTR PDFOutputName);

Parameters	PDFOutputName	Full path name of the new PDF file to create.
Returns	HRESULT	Returns S_OK if the merged file was successfully created. Any other value indicates an error, see GetLastError for more information.

2.1.61 PDFMergeInit

Start a new empty PDF file prepared for merging.
Use PDFMergeAddFile or PDFMergeAddFileEx to add files, and finally call PDFMergeClose to complete the file merge.

Syntax	HRESULT PDFMergeInit();	
Parameters	None	
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.62 PDFOptimize

Optimize a PDF file for faster loading and rendering. The optimization process will rescale images and rebuild the structure of the input file. The function rebuilds the content streams of all pages, templates and annotations.
Useless operators as well as errors in content streams will be removed. The resulting content streams are error free and usually smaller.
How much optimization takes effect depends on the quality of the original content streams.

<table>
<tr><td>Syntax</td><td colspan="2">HRESULT PDFOptimize(BSTR InputFile, BSTR OutputFile,
LONG MonoDPI, LONG ColorDPI, LONG MonoCompression,
LONG lColorCompression);</td></tr>
<tr><td>Parameters</td><td>InputFile</td><td>The source PDF file you want to optimize.</td></tr>
<tr><td></td><td>OutputFile</td><td>Optimized output file.</td></tr>
<tr><td></td><td>MonoDPI</td><td>Resolution to use for monochrome images in output file. Any image larger than the set resolution will be rescaled to fit this setting.</td></tr>
<tr><td></td><td>ColorDPI</td><td>Resolution to use for color images in output file. Any image larger than the set resolution will be rescaled to fit this setting.</td></tr>
<tr><td></td><td>MonoCompression</td><td>Set compression method to use for monochrome images:
<table><tr><th>Value</th><th>Compression method</th></tr><tr><td>0</td><td>Flate Compression (ZIP)</td></tr><tr><td>2</td><td>CCITT Group 3 Fax</td></tr><tr><td>3</td><td>CCITT Group 4 Fax</td></tr><tr><td>8</td><td>JBIG2</td></tr></table></td></tr>
<tr><td></td><td>ColorCompression</td><td>Set compression method to use for color images:
<table><tr><th>Value</th><th>Compression</th></tr></table></td></tr>
</table>

		<table><tr><th></th><th>**method**</th></tr><tr><td>0</td><td>Flate Compression (ZIP)</td></tr><tr><td>1</td><td>JPEG</td></tr><tr><td>4</td><td>LZW</td></tr><tr><td>7</td><td>JPEG2000</td></tr></table>
Returns	HRESULT	Returns S_OK if output file was successfully enctypted. Any other value indicates an error, see GetLastError for more information.

2.1.63 PDFRotatePages

Create a rotated copy of the input PDF file. The selected pages, defined by the Pages parameter, will be rotated using the given rotation factor.

Syntax	HRESULT PDFRotatePages(BSTR InputFile, BSTR OutputFile, BSTR Pages, LONG Rotation);	
Parameters	InputFile	Full path name for input PDF file that will be rotated and copied to the output file.
	OutputFile	Full path name for rotated output file.
	Pages	String describing the pages that are to be rotated. Set string to empty or use "-1" for all pages. Specify page indices as comma-separated values or ranges to process (e.g. "1, 2,3,7" or "1, 2, 3-7"). For this function the first-page index is 1.
	Rotation	The selected pages will be rotated using this rotation factor. Allowed values are 0, 180 or 270.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.64 PDFScale

Create a scaled copy of the input PDF file. Page width will be scaled using ScaleX and height will be scaled using ScaleY.

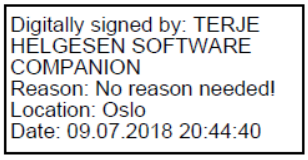
Syntax	HRESULT PDFScale(BSTR InputFile, BSTR OutputFile, DOUBLE ScaleX, DOUBLE ScaleY);	
Parameters	InputFile	Full path name for input PDF file that will be scaled and copied to the output file.
	OutputFile	Full path name for scaled output file.
	ScaleX	Scale factor to use for X direction.
	ScaleY	Scale factor to use for Y direction.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.65 PDFSignFile

Digitally sign a PDF file and add a signature image. To be able to sign a PDF file with scConverter you will need a certificate stored as a PKCS#12 file (PFX).

Syntax	HRESULT PDFSignFile(BSTR OriginalFile, BSTR SignedFile, LONG Position,BSTR ImageFile, BSTR CertFile, BSTR CertPassword, BSTR Reason);	
Parameters	OriginalFile	The PDF file you want to create a signed copy of. If you set this parameter to an empty file name ("") the currently loaded file will be used as original.
	SignedFile	The signed PDF file. This will be an exact copy of the orignal file but digitally signed.
	Position	The location of the signature image, either given as file or automatically generated. Following options available: 0. Top left 1. Top center 2. Top right 3. Bottom left 4. Bottom center 5. Bottom right
	ImageFile	Name of the image file to use. This can be a scanned signature or any other image. Supported formats include PNG, JPEG and TIFF. If no image file is provided the contro will create an automatic image which contain information about the signer and more.
	CertFile	Full path and name for the certificate file (PFX).
	CertPassword	The password to use for the certificate file.
	Reason	A string describing the reason for signing this file (optional).
	Location	A string describing the location (optional).
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

Here is an example on how an automatic created signature image may look:



2.1.66 PDFSplit

Split a multi-page PDF file into smaller files based on the parameters.

Syntax	HRESULT PDFSplit(BSTR InputFile, BSTR OutputFolder, BSTR FileLabel, long PagesPerFile);	
Parameters	InputFile	The multi-page PDF file that will be split into smaller files.
	OutputFolder	A valid folder name for the output files.

	FileLabel	Optional label to add to output filenames. By default the output files will be named "inputfilename_1.pdf", "inputfilename_2.pdf" and so on. By adding "part" as label, the output files will be named "inputfilename__part_1.pdf", "inputfilename__part_2.pdf" and so on.
	PagesPerFile	Number of pages per output file. If the input document contains 20 pages and you set page per file to 1, the control will create 20 files. If you set pages per file to e.g. 2, the control will output 10 files with 2 pages each.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.67 PDFToCAD

Convert a PDF file to an editable vector format file, for example Autodesk DXF.

Syntax	HRESULT PDFToCAD(BSTR InputFileName, BSTR OutputFileName, BSTR Format, long PageNo);	
Parameters	InputFileName	Name of the PDF file to convert.
	OutputFileName	Name of the exported vector file.
	Format	Export format to use for conversion. Supported formats: • CGM (Computer Graphics Metafile) • DWF (Drawing Web Format) • DXF (Autodesk Drawing Exchange Format) • GBR (Gerber RS274X) • PLT (HPGL/2) • SVG (Scalable Vector Format)
	Page	PDF page that should converted to vector format. Page numbering start at 0.
Returns	HRESULT	Returns S_OK if file was successfully converted. Any other value indicates an error, see GetLastError for more information.

2.1.68 ResizeImage

Resize an image file using one of two available high quality scaling algorithms. The width and height parameters are used to determine the new size, but the aspect ratio will be maintained. This method can be used for raster image and PDF file formats.

Syntax	HRESULT ResizeImage(BSTR InputFile, BSTR OutputFile, BSTR Format, enumResizeMethod Method, LONG Width, LONG Height);	
Parameters	InputFile	Input image file to resize.
	OutputFile	Name of resized output file.
	Format	File format to use for output file. This string need to be one of the following:

		• AVIF • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Method	Scaling algorithm to use. The following values are supported: 0. Avir 1. Lanczos
	Width	Maxmium width of resized image.
	Height	Maxmium height of resized image.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

More information regarding the available scaling algorithms:
a) Avir information: https://github.com/avaneev/avir
b) Lanczos information: https://en.wikipedia.org/wiki/Lanczos_resampling

2.1.69 ResizeImageEx

Resize an image file using one of two available high quality scaling algorithms.
See the ResizeImage description above for more details.

Syntax	HRESULT ResizeImage(LONG Handle, LONG Page, InputFile, BSTR OutputFile, BSTR Format, enumResizeMethod Method, LONG Width, LONG Height);	
Parameters	Handle	Handle of a file previously opened using the OpenFileEx method.
	Page	Page in input file to resize. Page index start at 0.
	OutputFile	Name of resized output file.
	Format	File format to use for output file. This string need to be one of the following: • AVIF • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Method	Rescale algorithm selection. Following values are supported: 0. Avir 1. Lanczos
	Width	Maxmium width of resized image.
	Height	Maxmium height of resized image.
Returns	HRESULT	Returns S_OK if success. Any other value

		indicates an error, see GetLastError for more information.

2.1.70 SaveCompareResult

Compare two files and save the result as an image file. If you set BackColor to red and FrontColor to blue all information that is equal in both files will be displayed as black. Information that has been deleted will be displayed using red, and new additions are displayed using blue color.

Syntax	HRESULT SaveCompareResult(BSTR BackFile, BSTR FrontFile, BSTR OutputFile, BSTR Format, OLE_COLOR BackColor, OLE_COLOR FrontColor, LONG ExportDPI);	
Parameters	BackFile	Name of the background file to use in the comparison.
	FrontFile	Name of the foreground file to use in the comparison.
	OutputFile	Name of the output file.
	Format	The file format to use for the saved image file. The following format identifiers are supported: • JPEG • PDF • PNG • TIFF
	BackColor	Color to use for the background file.
	FrontColor	Color to use for the foreground file.
	ExportDPI	Resolution to use for the saved image file. Higher values give better result but consumes more memory and the file size increases.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.71 SaveCompareResultEx

Compare two files and save the result as an image file. If you set BackColor to red and FrontColor to blue all information that is equal in both files will be displayed using EqualColor. Information that has been deleted will be displayed using red, and new additions are displayed using blue color.

Syntax	HRESULT SaveCompareResultEx(BSTR BackFile, BSTR FrontFile, BSTR OutputFile, BSTR Format, OLE_COLOR BgColor, OLE_COLOR FgColor, OLE_COLOR EqualColor, LONG ExportDPI);	
Parameters	BackFile	Name of the background file to use in the comparison.
	FrontFile	Name of the foreground file to use in the comparison.
	OutputFile	Name of the output file.
	Format	The file format to use for the saved image file. The following format identifiers are supported: • JPEG

		• PDF • PNG • TIFF
	BackColor	Color to use for the background file.
	FrontColor	Color to use for the foreground file.
	EqualColor	Color to use for information that are equal in both files. If you set this color to white, only the differences between the files will be visible.
	ExportDPI	Resolution to use for the saved image file. Higher values give better result but consumes more memory and the file size increases.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.72 SaveThumbnailEx

Save a page in the currently open file as a thumbnail image file.

Syntax	HRESULT SaveThumbnailEx(LONG Handle, LONG Page, LONG Width, LONG Height, LONG BitsPerPixel, BSTR Format, BSTR OutputFile);	
Parameters	Handle	Handle for file returned by OpenFileEx.
	Page	Page number to create a thumbnail of (first page is index 0)
	Width	Width of thumbnail image in pixels
	Height	Height of thumbnail image in pixels
	BitsPerPixel	The bits per pixel controls number of colors to use in the thumbnail image. Set this value to 1 for a monochrome (black & white) image, or 24 for a true color image.
	Format	The file format to use for the thumbnail image file. The following format identifiers are supported: • JPEG • PDF • PNG • TIFF
	OutputFile	Name of the created thumbnail image file.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.73 SaveQRImage

Create a QR barcode image based on input settings and save it to a PNG file.

Syntax	HRESULT SaveQRImage(BSTR TEXT, LONG Size, LONG Margin, LONG ErrorLevel, BSTR ImageFile);	
Parameters	Text	Text string to encode as QR.
	Size	Set to 1 for default size. Setting size to 2 will create an image double as large.
	Margin	Optional white margin around QR image in pixels.

	ErrorLevel	Error level to use for QR generation.Default is 0.
	ImageFile	Full path and name of the output image file (PNG).
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.74 SetConfigValue

This method can be used to customize the behavior for different functions and operations.

Syntax	HRESULT SetConfigValue(long ID, long Value);	
Parameters	ID	Identifier for the setting you want to change.
	Value	Value to set for the given setting.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

Available configuration settings:

ID	**Description**
0	Enable or disable fill and outline for all rectangles added during PE mode. A zero value will disable this behaviour and rectangles will only be filled (default mode). Set this value to non-zero value to enable both fill and outline.
1	Control how the header should be written in converted HPGL/2 files. Following values are supported: 0. Add standard header information 1. Add header using a custom paper size. Paper size have to be set using the SetPLTPaperSize method. 2. Add no HPGL/2 header information at all (no BP, IN, PS).
2	Set default bits per pixel to use for raster images in HPGL/2 files. Some files are missing the bits per pixel setting and you may need to make sure the correct value are set for the file to be loaded correctly. Supported values include 1,4,8 and 24. The DLL uses 24 bit as default bits per pixel.
3	Set absolute pattern length for line styles (LT) in exported HPGL/2 files. The length is set in 1/100 milimeters. The default value is 0.
4	Enable TrueType text for exported PDF files. Default is enabled.
5	Enable creation of a user profile when LibreOffice is used to open Office file formats. This may be required if you're using scConverter in for example a service running under system user account.
6	Set HEIC quality level. Must be between 0 and 100. Default value is 50.
7	Set JPEG quality level. Must be between 0 (highest compresion/lowest quality) and 100 (lowest compression/highest quality). Default value is 75.
8	Set WebP lossless mode where 1 is lossless and 0 is lossy. Default value is 1.
9	Set WebP quality level. Must be set to a value between 0 (highest compresion/lowest quality) and 100 (lowest compression/highest quality). Default value is 75.
10	Set WebP compression method. This setting controls quality/speed trade-off. Supported values between 0 and 6, where 0=fast and 6=slower, but higher compression.

11	Set Brotli compression quality flag. The value must be between 0 and 11, where 0 is the fastest and 11 is the highest compression. Defailt value is 11.
12	Set resolution to use when a PDF file is used for OCR detection. This setting will also be used if you are using any of the image processing methods on a PDF file (for example Deskew, Defox and more). The default value is 200.
13	Set number of bits per pixel to use when a PDF file is used for OCR detection. Default value is 24.
14	Set resolution (DPI) to use when searching for QR codes in a PDF file.
15	Set resolution to use when rendering arcs for raster output. The default value is 360 line segments for a full arc.
16	Enable or disable creation of progressive JPEG files. Set to 0 to create non progressive files, and any other value to create progressive files. The default value is 0.
17	Enable to disable transparency when converting multiple Gerber files to a CAD format (for example PDF or SVG) using the ConvertGerberLayersToCAD method. The default value is 0 (disabled).
18	Set JPEG XL quality level. Must be between 0 and 100. Default value is 90.
19	Enable or disable JPEG XL lossless mode. Default value is 0 (disabled).
20	Set AVIF quality level. Valid values are between 0 and 100. The default value is 60.
21	Set AVIF alpha channel quality level. Valid values are between 0 and 100. The default value is 100 (lossless).
22	Set AVIF compression speed. Valid values are between 0 (slowest) and 10 (fastest). The default value is 6.

2.1.75 SetGerberLayerVisible

Toggle the visibility for a layer previously added using AddGerberLayer.

Syntax	HRESULT SetGerberLayerVisible(LONG Index, VARIANT_BOOL Visible);	
Parameters	Index	Index to a layer previously added using the AddGerberLayer method.
	Visible	Enable or disable visibility for given layer.
Returns	HRESULT	Returns S_OK if success.

2.1.76 SetLoadPassword

Set password to use for loading encrypted (password protected) files. Password is currently supported for encrypted DWF, DWFX and PDF files.

Syntax	HRESULT SetLoadPassword(BSTR Password);	
Parameters	Password	Set password to use when loading password protected DWF, DWFX and PDF files.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.77 SetMargins

Add margins to converted files.

Syntax	HRESULT SetMargins(VARIANT_BOOL AddMargins, DOUBLE Left, DOUBLE Top, DOUBLE Right, DOUBLE Bottom);	
Parameters	AddMargins	If TRUE margins will be added to the converted file. Set to FALSE to disable margins.
	Left	Set left margin in millimeters.
	Top	Set top margin in millimeters.
	Right	Set right margin in millimeters.
	Bottom	Set bottom margin in millimeters.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.78 SetPDFCustomProperty

Add a custom PDF property to the next converted PDF file. Both the key and the value are user defined.

Syntax	HRESULT SetPDFProperty(BSTR Key, BSTR Value);	
Parameters	Key	A custom key name.
	Value	A custom value be attached to the given key name.
Returns	HRESULT	Returns S_OK if success.

2.1.79 SetPDFProperty

Set one of the predefined PDF properties to given value (strings only).

Syntax	HRESULT SetPDFProperty(LONG Key, BSTR Value);	
Parameters	Key	Following values are supported: 0. Author 1. Creator 2. Keywords 3. Producer 4. Subject 5. Title 6. Company
	Value	The string to set for given key.
Returns	HRESULT	Returns S_OK if success.

2.1.80 SetPenTableEntry

Set width, color and styles for the given pen table entry. Pen tables can be used to make for example thin lines thicker by setting a pen entry to a specific width. Pen tables are support for most CAD formats, which includes DWF, CGM and PLT.

Syntax	HRESULT SetPenTableEntry(long Pen, double Width, long Color,

	long LineStyle, long EndStyle);	
Parameters	Pen	Pen number to modify.
	Width	Pen width in millimeters. Set width to 0.0 if you want to keep the line width as defined in the file.
	Color	Pen Color (Windows COLORREF). When specifying an RGB color, the color value has the following hexadecimal form: 0x00bbggrr. Set a color 0xFFFFFFFF to use the color as defined in the file.
	Style	Windows GDI pen style. The following styles are supported: #define PS_SOLID 0 #define PS_DASH 1 /* ------- */ #define PS_DOT 2 /* ..... */ #define PS_DASHDOT 3 /* _._._ */ #define PS_DASHDOTDOT 4 /* _.._.._ */ Set to -1 to use the style defined in the file.
	Endstyle	The following line end styles are supported: 0. Round style (same as GDI PS_ENDCAP_ROUND) 1. Butt style (same as GDI PS_ENDCAP_FLAT) 2. Square style (same as GDI PS_ENDCAP_SQUARE) Set to -1 to use the style defined in the file
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

If you only want to change the width for all pens you could do the following:

```
int numpens = Scconverter.NumPens();
for (int pen=0; pen<numpens; pen++)
{
  Scconverter.SetPenTableEntry( pen, 0.3, 0xFFFFFFFF, -1, -1 );
}
```

This will change all pens to 0.3mm width, but keep any color and style already defined in the file.

2.1.81 SetPLTPaperSize

Define custom paper size to use for converted HPGL/2 files. To use custom paper size you will need to call SetConfigValue(1,1) before the conversion.

Syntax	HRESULT SetPLTPaperSize(LONG Unit, double Width, double Length)	
Parameters	Unit	Setup units to use for the paper size (width/length): 0. Millimeters 1. Inch 2. HPGL/2 units (1016 DPI)
	Width	Width of paper.
	Width	Length, or height, of paper.
Returns	HRESULT	Returns S_OK if success.

2.1.82 SetPropertyString

Set a property string with the given ID to the given value.

Syntax	HRESULT SetPropertyString(LONG ID, BSTR Value);	
Parameters	ID	Value for the the property to read, the following values are available: 0. **Word converter name.** Set the filename of the Word to PDF converter to use. The default value is "scWordToPDF.exe". 1. **Excel converter name**. Set the filename of the Excel to PDF converter to use. The default value is scExcelToPDF.exe". 2. **Powerpoint converter name**. Set the filename of the Powerpoint to PDF converter to use. The default value is "scPPTtoPDF.exe". 3. **Chinese PDF font.** Set the name of the font to use when creating searchable PDF files with Chinese text. Default value is "SimSun".
	Value	String value to set for given ID.
Returns	HRESULT	Returns S_OK if success.

2.1.83 SetSerialNumber

Use this function to unlock the component to use all its functionality.
You will receive a serial number when you purchase a developer license.

Syntax	HRESULT SetSerialNumber(BSTR SerialNumber);	
Parameters	SerialNumber	Serial number to unlock the component.
Returns	HRESULT	Returns S_OK if success. Any other value indicates that there is a problem with the serial number.

2.1.84 SetTIFFCompression

Set compression types to use for exported TIFF files.

Syntax	HRESULT SetTIFFCompression(LONG Monochrome, LONG Color, LONG TrueColor);	
Parameters	Monochrome	Set compression type to use for monochrome, 1-bit, files. Supported values:

Value	Compression method
1	No compression
4	CCITT Group 4 Fax
5	LZW
8	Deflate Compression (ZIP)
32773	Packbits
32946	Deflate (same as 8)

	Color	Set compression type to use for color files, 4-bit or 8-bit, files. Supported values: **Value** / **Compression method** 1 – No compression 5 – LZW 8 – Deflate Compression (ZIP) 32773 – Packbits 32946 – Deflate (same as 8)
	TrueColor	Set compression type to use for true color, 24-bit, files. Supported values: **Value** / **Compression method** 1 – No compression 5 – LZW 7 – JPEG 8 – Deflate Compression (ZIP) 32773 – Packbits 32946 – Deflate (same as 8)
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.85 SetTIFFTag

Override available TIFF tags to given value (strings only). These tags will be added to all TIFF files created by the component.

Syntax	HRESULT SetTIFFTag (LONG Index, BSTR Value);	
Parameters	Index	The following values are supported by this method: 0. Software 1. Artist 2. Copyright
	Value	The string to set for the given key.
Returns	HRESULT	Returns S_OK if success.

2.1.86 TextExtractEx

Extract text from an already open document.

Syntax	HRESULT TextExtractEx(LONG Handle, LONG Page, BSTR *Text);	
Parameters	Handle	Handle returned by OpenFileEx.
	Page	Page number to extract text from. First page is 0 (zero index).
	Text	Extracted text.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.87 TIFFSplit

Split a multi-page TIFF file into multiple single paged files (one per page).

Syntax	HRESULT TIFFSplit(BSTR InputFile, BSTR OutputFolder, BSTR FileLabel);	
Parameters	InputFile	The multi-page TIFF file that will be split into smaller files.
	OutputFolder	A valid folder name for the output files.
	FileLabel	Optional label to add to output filenames. By default the output files will be named "inputfilename_1.tif", "inputfilename_2.tif" and so on. By adding "part" as label, the output files will be named "inputfilename__part_1.tif", "inputfilename__part_2.tif" and so on.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.88 VBSClose

Close the file previously opened using the VBSOpen method.

Syntax	HRESULT VBSClose();	
Parameters	None	
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.89 VBSConvert

Convert a file previously opened using the VBSOpen method into a new file using the selected file format and other parameters.

Syntax	HRESULT VBSConvert(BSTR OutputFile, BSTR Format, LONG Page, DOUBLE ScaleFactor, DOUBLE Rotation, LONG BitsPerPixel, LONG DPI);	
Parameters	OutputFileName	Name of destination file.
	Format	Please see the Convert method description above for a list of supported formats.
	Page	Page of the opened document to output. Set this parameter to -1 to include all pages. To output only first page you can set this parameter to 0.
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	Rotation	Rotation factor in degrees.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for raster format conversion, for example TIFF or PNG. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors

		8. 256 Colors 24. True Color For TIFF and PNG formats you can also use 32 bits to create images with an alpha channel.
	DPI	Resolution given in pixels per inch. This parameter is used only for raster format conversion, for example TIFF.
Parameters	None	
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.90 VBSCreateGUID

Create and return a GUID (globally unique identifier) string as a variant COM type. This function can be used from VB Script.

Syntax	HRESULT VBSCreateGUID (VARIANT *GUID);	
Parameters	GUID	Returned GUID.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.91 VBSOpen

Open a file and keep it open until it is closed by calling VBSClose. It is only possible to keep one file open at a time using this method, and if a file is already opened it will be closed before the given file is opened. You can call VBSConvert, VBSResize and any of the VBSXXX properties if the file is open.
The VBSXXX methods were added to support development environments that cannot use the OpenFileEx based methods (for example VBScript).

Syntax	HRESULT VBSOpen(BSTR FileName)	
Parameters	FileName	Name of file to open.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

2.1.92 VBSResize

Create a resized copy of an image file previously opened using the VBSOpen method. The image will be resized using one of two available high quality scaling algorithms. The width and height parameters are used to determine the new size, but the aspect ratio will be maintained. This method can be used for raster image and PDF file formats.

Syntax	HRESULT VBSResize(BSTR OutputFile, BSTR Format, LONG Page, LONG Method, LONG Width, LONG Height);	
Parameters	OutputFile	Name of resized output file.
	Page	Page of the opened file to output. Page numbering starts at index 0.

	Format	File format to use for output file. This string need to be one of the following text strings: • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Method	Scaling algorithm to use. The following values are supported: 0. Avir 1. Lanczos
	Width	Maxmium width of resized image.
	Height	Maxmium height of resized image.
Returns	HRESULT	Returns S_OK if success. Any other value indicates an error, see GetLastError for more information.

Notes regarding scaling algorithms:

a) Avir information: https://github.com/avaneev/avir
b) Lanczos information: https://en.wikipedia.org/wiki/Lanczos_resampling

2.2 scConverter Properties

2.2.1 BackgroundColor

Set the background color to use for the different conversion methods. This property accepts a Windows color value.	
Type	OLE_COLOR
Access	Read and write
Default value	WHITE (0x00FFFFFF)

2.2.2 DWFExporterDPI

Set resolution to use for exported DWF files.	
Type	LONG
Access	Read and write
Default value	1000

2.2.3 DWFLoadHiddenText

If set to 1 (true) hidden text in DWF files will be loaded and included when converting to PDF. Set to 0 to ignore hidden text in DWF files.	
Type	LONG
Access	Read and write
Default value	0 (FALSE)

2.2.4 DWFLoadMarkup

If set to 1 (true) markup created by Autodesk Design Review will be loaded and included in the conversion. Set this option to 0 to ignore markup created by Design Review.	
Type	LONG
Access	Read and write
Default value	1 (TRUE)

2.2.5 DWFUsePaperSize

If set to 1 (true) all DWF files will be loaded using paper size if this is available, else the file will be loaded with drawing extents.	
Type	LONG
Access	Read and write
Default value	1 (TRUE)

2.2.6 DXFExtractImages

If set to 1 all images in the original file will be extracted to PNG files located in the same folder as the exported DXF file.	
Type	LONG
Access	Read and write
Default value	0 (FALSE)

2.2.7 DXFForceLineLW

If set to 1 all line elements will be written as LWPolyline instead of Line. This will increase file size but makes it possible to write lines with width. Line elements can only have a width of 1 pixel.	
Type	LONG
Access	Read and write
Default value	1 (TRUE)

2.2.8 DXFIgnoreWhiteAreas

If set to non-zero, all filled white areas will be ignored when the DXF files is written. This may be useful if you're converting PDF files to DXF because white areas may cause problems in AutoCAD.	
Type	LONG
Access	Read and write
Default value	0 (FALSE)

2.2.9 DXFWriteMM

If enabled all DXF files be written using coordinates in millimeter. If disabled DXF files will use Inch coordinates.	
Type	LONG
Access	Read and write
Default value	It depends on system regional settings.

2.2.10 GerberApertureFileName

Set filename for a Gerber aperture table to be used when loading RS-274D Gerber data.	
Type	BSTR
Access	Read and write
Default value	none

2.2.11 GetLastError

Retrieves the component's last-error code value. Possible error codes: 0. No error. 1. Unknown format. 2. Can't create the output file. Possible reasons could be security settings or out of disk space. 3. No license available. No serial number given, and the evaluation time (30 days) has expired. 4. Out of memory. Not enough memory is available to complete the conversion. 5. File read error. Could not load the file. 6. Unsupported file format. Could not load the file. 7. Error during export of pen table. Probably write access problem. 8. No pens defined. 9. Too many files are opened. This error is only returned by OpenFileEx. 10. Unknown error. 11. PDF Conformation failed to conform the file into PDF/A. 12. The PDF file is password protected. 13. Read access error. 14. Tesseract OCR engine is not installed on the system. 15. The file you tried to open is password protected. You need to set a valid password using the SetLoadPassword function before attempting to open this file again. 16. The selected output format is not supported by this method.	
Type	LONG
Access	Read only
Default value	none

2.2.12 GrayscaleMode

Enable or disable grayscale conversion. If set to non-zero, the converted files will only contain graphics using shades of gray. Note that enabling GrayscaleMode will disable MonochromeMode if it is active. Only one of the Monochrome and Grayscale modes may be active at a time.	
Type	LONG
Access	Read and write
Default value	0 (FALSE)

2.2.13 ImportersPath

Set full path to where the additional file importers are located.	
Type	BSTR
Access	Read and write
Default value	

2.2.14 MeasurementViewports

If this property is enabled (non-zero value) all DWF viewports with measurement settings will be included when PDF files are created. If set to zero, all viewports will be ignored.	
Type	LONG
Access	Read and write
Default value	1 (TRUE)

2.2.15 MonochromeMode

Enable or disable monochrome conversion. If set to non-zero, the converted files will only contain black and white graphics. Note that enabling MonochromeMode will disable GrayscaleMode if it is active. Only one of the Monochrome and Grayscale modes may be active at a time.	
Type	LONG
Access	Read and write
Default value	0 (FALSE)

2.2.16 NamedViewsAsBookmarks

Enable or disable creation of a bookmark for each named view in the original file (for example DWF).	
Type	LONG
Access	Read and write
Default value	0 (FALSE)

2.2.17 NumPens

Returns maximum number of pens that can be set using SetPenTableEntry.	
Type	LONG
Access	Read
Default value	256

2.2.18 OCRAvailable

Returns a non-zero value if the Tesseract OCR engine is installed on the system. If Tesseract is installed you can use the **CreateSearchablePDF** method to create searchable PDF files from scanned PDF, TIFF, PNG and other file formats. See appendix H for more information about Tesseract.	
Type	BOOL
Access	Read
Default value	TRUE if Tesseract is installed and FALSE if it is not installed.

2.2.19 PDFHighPrecision

Enable or disable creation of high-resolution PDF files. If set to non-zero, all coordinates will be written using 4 decimals to give higher accuracy. If this option is disabled 2 decimals will be used for coordinates. High resolution will increase file size.	
Type	LONG
Access	Read and write
Default value	1 (TRUE)

2.2.20 PDFLargeFormat

Enables or disables creation of large format PDF files. Large format files are files with width or height beyond 5x5m.	
Type	LONG
Access	Read and write
Default value	1 (TRUE)

2.2.21 PDFLayers

Enable or disable creation of layered PDF files. If set to non-zero, layers and pens in input files will be converted to individual PDF layers.	
Type	LONG
Access	Read and write
Default value	1 (TRUE)

2.2.22 PDFMonoRasterCompression

Select the compression method to use for monochrome (1-bit) raster images in exported PDF files. The following compression methods are available: 0. Deflate. 1. CCITT Group 4. 2. JBIG2.	
Type	LONG
Access	Read and write
Default value	0 - Deflate

2.2.23 PDFSearchable

Enable or disable creation of searchable PDF files. If this option is enabled and the original file contains text, the converted PDF file will be searchable.	
Type	LONG
Access	Read and write
Default value	0

2.2.24 PDFTransparency

Enable or disable use of transparency in exported PDF files. Some printer drivers don't support transparency, and turning off transparency for exported PDF files may make your files print faster.	
Type	LONG
Access	Read and write
Default value	1

2.2.25 PDFTrueColorRasterCompression

Select the compression method to use for true color (24 and 32 bit) raster images in exported PDF files. The following compression methods are available: 0. Deflate 1. JPEG 2. JPEG2000	
Type	LONG
Access	Read and write
Default value	0 - Deflate

2.2.26 PDFWriterFormat

Select the PDF standard to use for exported PDF files. The following options are available: 0. PDF 1.7 (editable) 1. PDF/A-1B 2. PDF/A-2B 3. PDF/A-3B 4. PDF/A-2U 5. PDF/A-3U 6. PDF/A-4 7. PDF/A-4e	
Type	LONG
Access	Read and write
Default value	2 (PDF/A-2B).

2.2.27 PLTEncoded

Enable or disable encoded polylines (PE instruction) in exported PLT (HPGL/2) files. Files with encoded polylines are usually smaller (compressed).	
Type	LONG
Access	Read and write
Default value	1 (Use encoded polygons)

2.2.28 TIFFSingleStrip

Enable or disable creation of single-strip TIFF files. By default, the component will create multi-strip files. Single-strip files are usually smaller (in terms of file size) than multi-strip files. A multi-strip file may however have some advantages: • Less memory can be used to process the image because an application can process one strip at a time. • Random access to image data is less complicated. Set to 1 (true) to create single-strip TIFF files	
Type	LONG
Access	Read and write
Default value	0

2.2.29 UseROInstruction

HPGL-2 plotter files may contain a RO (Rotation) instruction. Enabled this option to enable scConverter to recognize this instruction when converting the file.	
Type	LONG
Access	Read and write
Default value	0

2.2.30 UsePenTable

Enable or disable pen table settings. If set to non-zero, the converted files be converted using the current pen table settings. Pen tables are supported for PLT (HPGL, HPGL/2 and Calcomp), CGM and DWF file formats.	
Type	LONG
Access	Read and write
Default value	0

2.2.31 VBSDPI

Return the resolution (DPI) for the file opened using the VBSOpen method.	
Type	LONG
Access	Read only
Value	Resolution in dots per inch, example 200.

2.2.32 VBSFormat

Return the file format identifier for the file opened using the VBSOpen method.	
Type	LONG
Access	Read only
Value	One of the following values is returned:

	1. HPGL/2 2. TIFF 3. CALS 4. PNG 5. JPEG 6. Windows BMP 7. CGM (Computer Graphics Metafile) 8. Calcomp Plot Format 9. Autodesk DWF 10. Gerber Plot Format 11. Adobe PDF 12. WEBP Google Image Format 13. GIF Image Format 14. EDMICS Raster Format 15. Intergraph Raster Format 16. JPEG2000 17. Text Format 18. Word Format 19. Excel Format 20. Powerpoint Format 21. LibreOffice Format 22. Excellon Drill format 23. HEIC Image Format 24. PSD Image Format (Adobe Photoshop) 25. JPEG-XL Image Format 26. AVIF

2.2.33 VBSHeight

Return the height for current page in a file opened using the VBSOpen method.	
Type	DOUBLE
Access	Read only
Value	Height in millimeters.

2.2.34 VBSPage

Set or return the current page in a file opened using the VBSOpen method.	
Type	LONG
Access	Read and write
Value	The currently active page, first page is 0.

2.2.35 VBSPages

Return the number of pages in a file opened using the VBSOpen method.	
Type	LONG
Access	Read only
Value	Number of pages, for example 1 (single page file).

2.2.36 VBSPixHeight

Return the height in pixels for the current page in a file opened using the VBSOpen method.	
Type	LONG
Access	Read only
Value	Height in number of pixels.

2.2.37 VBSPixWidth

Return the width in pixels for the current page in a file opened using the VBSOpen method.	
Type	LONG
Access	Read only
Value	Width in number of pixels.

2.2.38 VBSWidth

Return the width for current page in a file opened using the VBSOpen method.	
Type	DOUBLE
Access	Read only
Value	Width in millimeters.

2.2.39 Version

Return a string with the version of the currently installed scConverter DLL.	
Type	BSTR
Access	Read only
Value	For example "7.20.0.264"

2.3 scConverter Exported Functions

The DLL exports several functions that may be called directly without any use of COM. Most languages can use these functions, for example C++, VB.Net, C#, Java and more. In this section you will find a description of all the exported functions that are available.

Most of the functions is available as both Unicode and multi-byte (ANSI) variants, in this case syntax for both are shown. If there is an ANSI variant it will have the same name but with a capital A added to the function name.

All ANSI functions will automatically detect and properly decode UTF-8 encoded files and folder names.

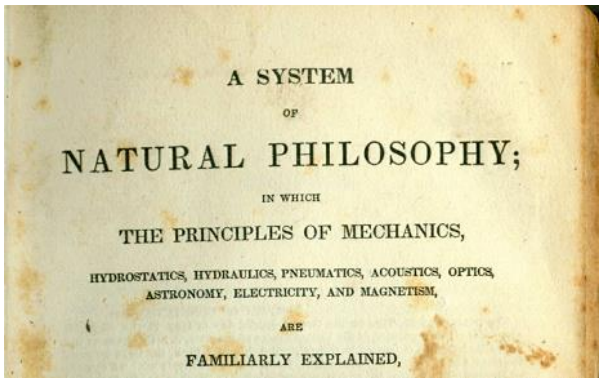
2.3.1 scAddPCBLayer

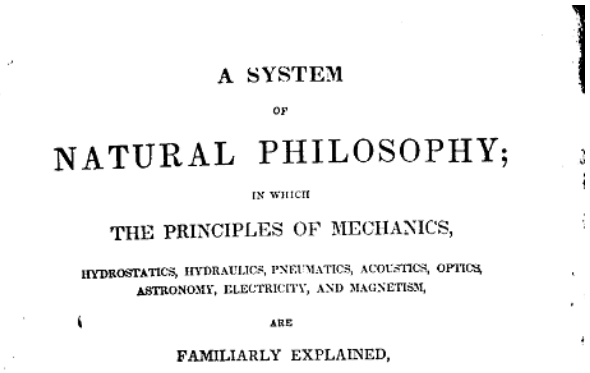
Add a Gerber, HPGL/2 or Excellon file as a new layer to use for later conversion. When all required layers are added you may create a new vector file (e.g.: PDF, SVG, DXF) by using the scConvertPCBLayersToCAD function. You will need to initialize a new PCB with scStartPCB before using this function.

<table>
<tr><td>Syntax</td><td colspan="2">int AddPCBLayer(WCHAR *InputFile, long Color, long Flags);</td></tr>
<tr><td>Parameters</td><td>InputFile</td><td>Name of file to add as a layer. The file can either be local (UNC) or URL (http:// or https://).</td></tr>
<tr><td></td><td>Color</td><td>Color value to use for this Gerber layer during conversion. Color values are given as 0x00BBGGRR.</td></tr>
<tr><td></td><td>Flags</td><td>The following values are supported:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Set layer as transparent. Color values are in this case given as 0xAABBGGRR. Valid opacity range is from 0 to 255, where 255 is full opacity. Sample color value : 0x800000FF. This will set the color to full red (255) and an opacity value of 128 (0x80).</td></tr>
</table>
Values can be combined.</td></tr>
<tr><td>Returns</td><td>int</td><td>Returns zero if file was successfully compressed. Please see GetLastError for a description of possible error codes.</td></tr>
</table>

2.3.2 scBinarizeImage

The binarize filter can convert a color image to black and white image (1 bits per pixel). Like the DefoxImage method described below, it can also be used to remove stains from old, scanned documents. The pictures below show an image before and after running the binarize filter (Threshold value used is 0.5). The sample file used is also included in the SDK.


A SYSTEM
OF
NATURAL PHILOSOPHY;
IN WHICH
THE PRINCIPLES OF MECHANICS,
HYDROSTATICS, HYDRAULICS, PNEUMATICS, ACOUSTICS, OPTICS,
ASTRONOMY, ELECTRICITY, AND MAGNETISM,
ARE
FAMILIARLY EXPLAINED,


A SYSTEM
OF
NATURAL PHILOSOPHY;
IN WHICH
THE PRINCIPLES OF MECHANICS,
HYDROSTATICS, HYDRAULICS, PNEUMATICS, ACOUSTICS, OPTICS,
ASTRONOMY, ELECTRICITY, AND MAGNETISM,
ARE
FAMILIARLY EXPLAINED,

ANSI	int scBinarizeImageA(const char* SerialNumber, const char* InputFile, const char* OutputFile, const char* Format, double Threshold);	
UNICODE	int scBinarizeImage(const WCHAR* SerialNumber, const WCHAR* InputFile, const WCHAR* OutputFile, const WCHAR* Format, double Threshold);	
Parameters	SerialNumber	Your serial number.
	InputFile	Input file to process using the dexofing algorithm.
	OutputFile	Name of processed output file.
	Format	File format to use for output file. This string have to be one of the following: • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Threshold	Threshold value to use for the binarization. Threshold values must be between 0 to 1.
Returns	int	Returns zero if file was successfully compressed. Please see GetLastError for a description of possible error codes.

2.3.3 scCheckFormat

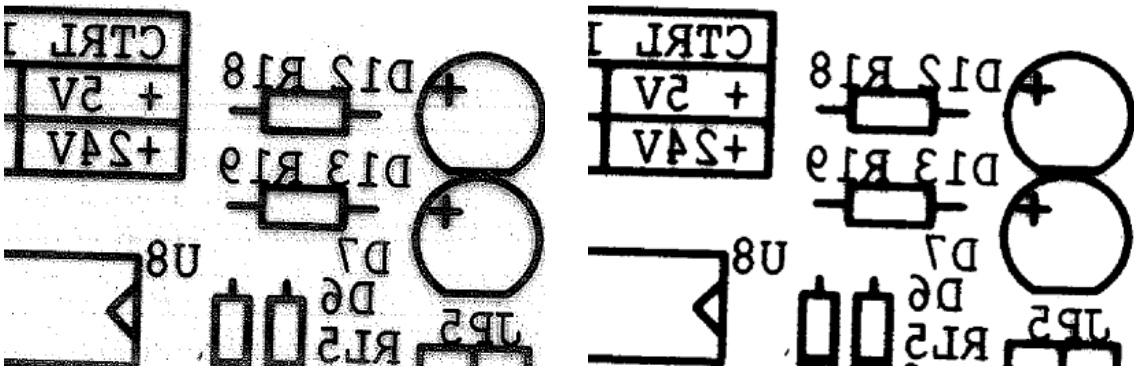
Check if the given file can be opened by the converter.

ANSI	int scCheckFormatA(char *InputFile, long *Format);	
UNICODE	int scCheckFormat(WCHAR *InputFile, long *Format);	
Parameters	InputFile	Name of the file to check.

	Format	One of the following values is returned: 0. Unknown file format 1. HPGL/2 2. TIFF 3. CALS 4. PNG 5. JPEG 6. Windows BMP 7. CGM (Computer Graphics Metafile) 8. Calcomp Plot Format 9. Autodesk DWF 10. Gerber Plot Format 11. Adobe PDF 12. WEBP Google Image Format 13. GIF Image Format 14. EDMICS Raster Format 15. Intergraph Raster Format 16. JPEG2000 17. Text Format 18. Word Format 19. Excel Format 20. Powerpoint Format 21. LibreOffice Format 22. Excellon Drill Format 23. HEIC Image Format 24. PSD Image Format (Adobe Photoshop) 25. JPEG-XL Image Format 26. AVIF Image Format
Returns	int	Error code, zero if success. If successfully checked the file format identifier will be returned using the Format parameter. Please see GetLastError for a description of possible error codes.

2.3.4 scCleanupImage

Remove noise from an image file. This method is only supported for raster image file formats. Below are two pictures that show a sample image before and after CleanupImage has been applied:



ANSI	int scCleanupImageA(const char* SerialNumber, const char* InputFile, const char* OutputFile, const char* Format);

UNICODE	int scCleanupImage(const WCHAR* SerialNumber, const WCHAR* InputFile, const WCHAR* OutputFile, const WCHAR* Format);	
Parameters	SerialNumber	Your serial number.
	InputFile	Input image file that will be cleaned.
	OutputFile	Name of cleaned output file.
	Format	File format to use for output file. This string have to be one of the following: • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
Returns	int	Returns zero if file was successfully processed. Please see GetLastError for a description of possible error codes.

2.3.5 scCompressFile

Compress a file using the given compression method.

ANSI	Int scCompressFileA(char *Serialnumber, char *InputFile, char *OutputFile, long CompressionMethod);	
UNICODE	Int scCompressFile(WCHAR *Serialnumber, WCHAR *InputFile, WCHAR *OutputFile, long CompressionMethod);	
Parameters	Serialnumber	Your serial number.
	InputFile	Full path to input file to compress.
	Outputfile	Full path to compressed output file.
	CompressionMethod	Compression method to use: 0 : GZIP compression method. 1 : Brotli compression method. 2 : BZIP2 compression method. 3 : ZSTD compression method.
Returns	int	Returns zero if file was successfully compressed. Please see GetLastError for a description of possible error codes.

2.3.6 scConvertFile

Convert the input file to a new file using the selected file format.

ANSI	int scConvertFileA(char *SerialNumber, char *InputFile, char *OutputFile, char *Format, double Scale, long BitsPerPixel, long DPI);
UNICODE	int scConvertFile(WCHAR *SerialNumber, WCHAR *InputFile, WCHAR *OutputFile, WCHAR *Format, double Scale, long BitsPerPixel, long DPI);

Parameters	SerialNumber	Your serial number.
	InputFile	Name of the file to convert.
	OutputFile	Name of destination file.
	Format	The following output formats are supported: • BMP (Windows Bitmap) • CALS (CALS Type 1 CCITT-G4 Raster Format) • CGM (Computer Graphics Metafile) • DWF (Drawing Web Format) • DXF (AutoDesk Drawing Exchange Format) • EMF (Windows Enhanced Metafile) • GBR (Gerber RS274X) • HPRTL(HP-RTL) • JPEG (JFIF Compliant) • JXL (JPEG-XL) • PCX (Paintbrush Format) • PDF (Acrobat PDF) • PDFRASTER (Acrobat PDF raster). • PNG (Portable Network Graphics) • PS (Adobe Postscript) • SVG (Scalable Vector Format) • TIFF (Tagged Image File Format) • WEBP (Google Image Format) • WMF (Windows Metafile)
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for raster format conversion, for example TIFF or PNG. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color For TIFF, HEIC, WEBP and PNG output file formats you can use 32 bits to create images with an alpha channel.
	DPI	Resolution given in pixels per inch. This parameter is used ony for raster format conversion, for example TIFF or PNG.
Returns	int	Error code, zero if file was successfully converted. Please see GetLastError for a description of possible error codes.

2.3.7 scConvertFileEx

Convert the input file to a new file using the selected file format. This function includes the option to rotate and mirror the output file.

ANSI	int scConvertFileExA(char *SerialNumber, char *InputFile, char *OutputFile,char *Format, double Scale, double Rotation, BOOL MirrorX, BOOL MirrorY, long BitsPerPixel, long DPI);

UNICODE	int scConvertFileEx(WCHAR *SerialNumber, WCHAR *InputFile, WCHAR *OutputFile, WCHAR *Format, double Scale, double Rotation, BOOL MirrorX, BOOL MirrorY, long BitsPerPixel, long DPI);	
Parameters	SerialNumber	Your serial number.
	InputFile	Name of the file to convert.
	OutputFile	Name of destination file.
	Format	The following output formats are supported: • BMP (Windows Bitmap) • CALS (CALS Type 1 CCITT-G4 Raster Format) • CGM (Computer Graphics Metafile) • DWF (Drawing Web Format) • DXF (AutoDesk Drawing Exchange Format) • EMF (Windows Enhanced Metafile) • GBR (Gerber RS274X) • HPRTL(HP-RTL) • JPEG (JFIF Compliant) • JXL (JPEG-XL) • PCX (Paintbrush Format) • PDF (Acrobat PDF) • PDFRASTER (Acrobat PDF raster). • PNG (Portable Network Graphics) • PS (Adobe Postscript) • SVG (Scalable Vector Format) • TIFF (Tagged Image File Format) • WEBP (Google Image Format) • WMF (Windows Metafile)
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	Rotation	Rotate the file clockwise in degrees.
	MirrorX	Mirror the output file about x-axis.
	MirrorY	Mirror the output file about y-axis.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for raster format conversion, for example TIFF or PNG. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color For TIFF, HEIC, WEBP and PNG output file formats you can use 32 bits to create images with an alpha channel.
	DPI	Resolution given in pixels per inch. This parameter is used ony for raster format conversion, for example TIFF or PNG.
Returns	int	Error code, zero if file was successfully converted. Please see GetLastError for a description of possible error codes.

2.3.8 scConvertFilePage

Convert a single page from input file to a new file using the selected file format.

ANSI	int scConvertFilePageA(char *SerialNumber, char *InputFile, char *OutputFile, char *Format, long Page, double Scale, long BitsPerPixel, long DPI);	
UNICODE	int scConvertFilePage(WCHAR *SerialNumber, WCHAR *InputFile, WCHAR *OutputFile, WCHAR *Format, long Page, double Scale, long BitsPerPixel, long DPI);	
Parameters	SerialNumber	Your serial number.
	InputFile	Name of the file to convert.
	OutputFile	Name of destination file.
	Format	Select the file format to use for the output file. Please see the scConvertFile function above for a list of available formats.
	Page	Index of the page to extract from the input file, and convert to a new output file. First page number is 0 (zero indexed).
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for raster format conversion, for example TIFF or PNG. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color For TIFF, HEIC, WEBP and PNG output file formats you can use 32 bits to create images with an alpha channel.
	DPI	Resolution given in pixels per inch. This parameter is used ony for raster format conversion, for example TIFF or PNG.
Returns	int	Error code, zero if file was successfully converted. Please see GetLastError for a description of possible error codes.

2.3.9 scConvertMarkup

Convert the input file together with markup data to a new file using the given format.

ANSI	int scConvertMarkupA(char *SerialNumber, char *InputFile, char *Markup,char *OutputFile, char *Format, double Scale, long BitsPerPixel, long DPI);	
UNICODE	int scConvertMarkup(WCHAR *SerialNumber, WCHAR *InputFile, WCHAR *Markup, WCHAR *OutputFile, WCHAR *Format, double Scale, long BitsPerPixel, long DPI);	
Parameters	SerialNumber	Your serial number.

	InputFile	Name of file to convert.
	Markup	This may either be a filename that contains markup data, or it may be string containing markup XML data. See section 3.6 for description of the markup XML format.
	OutputFile	Name of destination file.
	Format	Select the file format to use for the output file. Please see the scConvertFile function above for a list of available formats.
	Scale	Scalefactor to apply, set to 1.0 to use original scaling.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for raster format conversion, for example TIFF or PNG. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color For TIFF, HEIC, WEBP and PNG output file formats you can use 32 bits to create images with an alpha channel.
	DPI	Resolution given in pixels per inch. This parameter is used ony for raster format conversion, for example TIFF or PNG.
Returns	int	Error code, zero if file was successfully converted. Please see GetLastError for a description of possible error codes.

2.3.10 scConvertPCBLayersToCAD

Convert all loaded Gerber, Excellon and HPGL/2 layers to a CAD formatted file.
To be able to use this method one or more files must have previously been added as layers using the scAddPCBLayer method.

Syntax	int ConvertGerberLayersToCAD(WCHAR *OutputFile, WCHAR *Format);	
Parameters	OutputFile	Name of the destination file.
	Format	Select the output format to use. The following formats are supported by this method: • CGM (Computer Graphics Metafile) • DWF (Drawing Web Format) • DXF (AutoDesk Drawing Exchange Format) • GBR (Gerber RS274X) • HPGL (HPGL/2) • PDF (Acrobat PDF) • PS (Adobe Postscript) • SVG (Scalable Vector Format)
Returns	int	Error code, zero if file was successfully converted. Please see GetLastError for a description of possible error codes.

2.3.11 scConvertPCBLayersToMP

Convert all loaded Gerber, Excellon and HPGL/2 layers into a multipage file, one layer per page. To be able to use this method one or more files must have previously been added as layers using the scAddPCBLayer method.

Syntax	int scConvertPCBLayersToMP(WCHAR *OutputFile, WCHAR *Format, LONG BitsPerPixel, LONG DPI);	
Parameters	OutputFile	Name of the destination file.
	Format	Select the output format to use. The following formats are supported by this method: • DWF (Drawing Web Format) • HPGL(HPGL/2) • PDF (Acrobat PDF) • TIFF (Tagged Image File Format)
	BitsPerPixel	Number of colors to used for TIFF output. Supported values: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color
	DPI	Resolution to use for TIFF output given in pixels per inch.
Returns	int	Returned error code, zero if file was successfully converted. Please see GetLastError for a description of possible error codes.

2.3.12 scConvertSVGToImage

Convert the input SVG file to an image which will be returned as a DIB (device independent bitmap).

ANSI	int scConvertSVGToImageA(char *SerialNumber, char *InputFile, LONG MaxSize HGLOBAL *DIB);	
UNICODE	Int scConvertSVGToImage(WCHAR *SerialNumber, LONG MaxSize, HGLOBAL *DIB);	
Parameters	SerialNumber	Your serial number.
	InputFile	Name of input SVG file to convert.
	MaxSize	Maximum width or height of the image. Aspect ratio will be maintained. Set this parameter to 0 if you want to use the original SVG dimensions.
	DIB	Returned DIB data.
Returns	int	Returned error code, zero if file was successfully converted. Please see GetLastError for a description of possible error codes.

2.3.13 scConvertSVGToImageFile

Convert the input SVG file to a new image file of the given file format.

ANSI	int scConvertSVGToImageFileA(char *SerialNumber, char *InputFile, char *OutputFile, char *Format, LONG MaxSize, LONG DPI);	
UNICODE	Int scConvertSVGToImageFile(WCHAR *SerialNumber, WCHAR *InputFile, WCHAR *OutputFile, WCHAR *Format, LONG MaxSize, LONG DPI);	
Parameters	SerialNumber	Your serial number.
	InputFile	Name of input SVG file to convert.
	OutputFile	Name of destination file.
	Format	Select one of the supported output formats: • AVIF (AV1 Image File Format) • BMP (Windows Bitmap) • HEIC (High Efficiency Image File Format) • JPEG (JFIF Compliant) • JXL (JPEG-XL) • PDF (Acrobat PDF) • PNG (Portable Network Graphics) • TIFF (Tagged Image File Format) • WEBP (Google Image Format)
	MaxSize	Maximum width or height of the image. Aspect ratio will be maintained. Set this parameter to 0 if you want to use the original SVG dimensions.
	DPI	Resolution given in pixels per Inch.
Returns	int	Returned error code, zero if file was successfully converted. Please see GetLastError for a description of possible error codes.

2.3.14 scConvertToImage

Convert the input file to a new image file of the given file format. This function gives improved control of the created image.

ANSI	Int scConvertToImageFileA(char *SerialNumber, char *InputFile, char *OutputFile, char *Format, long Page, double Scale, long ImageWidth, long ImageHeight, long BitsPerPixel, long DPI, long Flags, double dOffsetX, double dOffsetY);	
UNICODE	int scConvertToImageFile(WCHAR *SerialNumber, WCHAR *InputFile, WCHAR *OutputFile, WCHAR *Format, long Page, double Scale, long ImageWidth, long ImageHeight, long BitsPerPixel, long DPI, long Flags, double dOffsetX, double dOffsetY);	
Parameters	SerialNumber	Your serial number.
	InputFile	Name of the file to convert.
	OutputFile	Name of destination file.
	Format	The following output formats are supported: • AVIF (AV1 Image File Format) • BMP (Windows Bitmap) • CALS (CALS Type 1 CCITT-G4 Raster Format) • HEIC (High Efficiency Image File Format)

		• JPEG (JFIF Compliant) • JXL (JPEG-XL) • PCX (Paintbrush Format) • PDF (Acrobat PDF) • PDFRASTER (Acrobat PDF raster). • PNG (Portable Network Graphics) • TIFF (Tagged Image File Format) • WEBP (Google WebP Image Format)
	Page	Index of the page to extract from the input file, and convert to a new output file. First page number is 0 (zero indexed).
	Scale	Scale factor to apply, set to 1.0 to use original scaling.
	ImageWidth	Width of the destination image file in pixels.
	ImageHeight	Height of the destination image file in pixels.
	BitxPerPixel	Bits Per Pixel. This parameter is used ony for raster format conversion, for example TIFF or PNG. Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color For TIFF, AVIF, JPEG-XL HEIC, WEBP and PNG output file formats you can use 32 bits to create images with an alpha channel.
	DPI	Resolution given in pixels per inch. This parameter is used ony for raster format conversion, for example TIFF or PNG.
	Flags	Following flag bits are supported: 1. The original Gerber offsets will be used during conversion. 2. The output file will be scaled to fit the given output size.
	OffsetX	Left offset in pixels.
	OffsetY	Top offset in pixels.
Returns	int	Error code, zero if file was successfully converted. Please see GetLastError for a description of possible error codes.

2.3.15 scConvertToPDF

Convert the input file into a PDF file. This function accepts all supported input formats.

ANSI	int scConvertToPDFA(char *SerialNumber, char *InputFile, char *OutputFile);	
UNICODE	int scConvertToPDF(WCHAR *SerialNumber, WCHAR *InputFile, WCHAR *OutputFile);	
Parameters	SerialNumber	Your serial number.
	InputFile	Name of file to convert.

	OutputFile	Name of destination PDF file.
Returns	int	Error code, zero if file was successfully converted. Please see GetLastError for a description of possible error codes.

2.3.16 scConvertToPaperSize

Convert the input file to a new file using the given format and selected paper size.

ANSI	int scConvertToPaperSizeA(char *SerialNumber, char *InputFile, char *OutputFile, char *Format, long PaperSizeIndex, BOOL KeepPaper, long Page, long BitsPerPixel, long DPI);	
UNICODE	int scConvertToPaperSize(WCHAR *SerialNumber, WCHAR *InputFile, WCHAR *OutputFile, WCHAR *Format, LONG PaperSizeIndex, BOOL KeepPaper, long Page, long BitsPerPixel, long DPI);	
Parameters	SerialNumber	Your serial number.
	InputFile	Name of input file to convert.
	OutputFile	Name of destination file.
	Format	Select the file format to use for the output file. Please see the scConvertFile function above for a list of available formats.
	PaperSizeIndex	The index of the predefined paper size to use for conversion. The output file will be scaled to fit the selected paper size. Please see appendix D for a list of supported paper formats.
	KeepPaper	If enabled margins will be added if necessary, to make sure the output use the exact paper size.
	Page	Index of the page to extract from the input file, and convert to a new output file. First page number is 0 (zero indexed). Set to -1 to convert all pages in the document..
	BitsPerPixel	Supported values for BitsPerPixel: 1. Monochrome, black & white 4. 16 Colors 8. 256 Colors 24. True Color This parameter is used ony for raster format conversion.
	DPI	Resolution given in pixels per Inch. This parameter is used ony for raster format conversion.
Returns	int	Error code, zero if file was successfully converted. Please see GetLastError for a description of possible error codes.

2.3.17 scCreateGUID

Create and return a GUID (globally unique identifier) string.

Syntax	int scCreateGUID(WCHAR *GUID)	
Parameters	GUID	Returned GUID
Returns	Int	Returns 0 if successful.

2.3.18 scCreateSearchablePDF

Create a searchable PDF file from a raster file, for example a scanned PDF file. You can create a searchable PDF file from all supported raster formats which include PDF, TIFF, PNG, JPEG and more. This method is only available if Tesseract OCR engine is installed on the system. See appendix H for more information about Tesseract. You can use the **scOCRAvailable** function to check if Tesseract is installed on the system.

ANSI	Int scCreateSearchablePDFA(char *InputFile, char *OutputFile);	
UNICODE	Int scCreateSearchablePDF(WCHAR *InputFile, WCHAR *OutputFile);	
Parameters	InputFileName	Name of input file
	OutputFileName	Name of destination PDF file with searchable text.
Returns	Int	Returns zero if file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.3.19 scDefoxImage

Remove the stain from old, scanned documents. See the DefoxImage description above for more information. This method is only supported for raster image file formats.

ANSI	int scDefoxImageA(const char* SerialNumber, const char* InputFile, const char* OutputFile, char* Format, double Threshold);	
UNICODE	int scDefoxImage(const WCHAR* SerialNumber, const WCHAR* InputFile, const WCHAR* OutputFile, const WCHAR* Format, double Threshold);	
Parameters	Serial Number	Your serial number.
	InputFile	Name of of input file to process.
	OutputFile	Name of processed output file.
	Format	File format to use for output file. This string have to be one of the following: • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Threshold	Threshold value to use for the destaining.

		Threshold values must be between 0 to 1.
Returns	Int	Returns zero if an output file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.3.20 scDeskewImage

Deskew, or straighten, a skewed image by the given angle. If you pass an angle value of 0.0, the method will calculate the optimal angle before processing the image. This method is only supported for raster image file formats.

ANSI	int scDeskewImageA(const char* SerialNumber, const char* InputFile, const char* OutputFile, const char* Format, double Angle);	
UNICODE	int scDeskewImage(const WCHAR* SerialNumber, const WCHAR* InputFile, const WCHAR* OutputFile, const WCHAR* Format, double Angle);	
Parameters	Serial Number	Your serial number.
	InputFile	Input file that should be deskewed.
	OutputFile	Name of processed output file.
	Format	File format to use for output file. This string have to be one of the following: • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Angle	Deskew angle in degrees.
Returns	HRESULT	Returns zero if an output file is created successfully. Any other value indicates an error, see GetLastError for more information.

2.3.21 scDrillFormatSettings

Configure the settings that may be required to load Excellon drill files correctly.

Syntax	void scSetDrillFormatSettings(BOOL Incremental, long NumDigits, long NumDecimals, LONG Units, BOOL LeadingZeros, BOOL TrailingZeros);	
Parameters	Incremental	If true incremental coordinates are used. Set to false for absolute coordinates.
	Preceding	Number of digits before decimal point.
	Succeeding	Number of digits after decimal point.
	Units	Units used for coordinates. Set to 0 for Inch and 1 for Millimeters.
	Leading	Leading zeros are suppressed if set to true.

	Trailing	Trailing zeros are suppressed if set to true.
Returns	void	

2.3.22 scEndPCB

Unload all added layers added by the scAddPCBLayer function from the memory.

Syntax	int scEndPCB();	
Parameters	None	
Returns	int	Error code, zero if file was successfully inspected. Please see GetLastError for a description of possible error codes.

2.3.23 scGetFileDimensions

Return the dimensions of the given page in the given file. All values are in millimeters.

ANSI	int scGetFileDimensionsA(char *InputFile, long Page, double *OffsetX, double *OffsetY, double *Width, double *Height, long *Rotation);	
UNICODE	int scGetFileDimensions(WCHAR *InputFile, long Page, double *OffsetX, double *OffsetY, double *Width, double *Height, long *Rotation);	
Parameters	InputFile	Full path of the file to inspect.
	Page	Index of the page to query. First page number is 0 (zero indexed).
	OffsetX	Returned original file x offset of the selected page.
	OffsetY	Returned original file y offset of the selected page.
	Width	Returned width of the selected page.
	Height	Returned height of the selected page.
	Rotation	File rotation in angles. This value is only returned for HPGL/2 files if the RO instruction is present.
Returns	int	Error code, zero if file was successfully inspected. Please see GetLastError for a description of possible error codes.

2.3.24 scGetNumFilePages

Return the number of pages in the given file.

ANSI	int scGetNumFilePagesA(char *InputFile, long *Pages);	
UNICODE	int scGetNumFilePages(WCHAR *InputFile, long *Pages) ;	
Parameters	InputFile	Full path to the file to inspect.
	Pages	Returned number of pages in the file.
Returns	Int	This function returns zero if successful.

2.3.25 scGetNumPens

Return the maximum number of pens that can be used for conversion.

Syntax	int scGetNumPens(long *Pens);	
Parameters	Pens	Maximum number of pens.
Returns	int	This function returns zero if successful.

2.3.26 scGetProperty

Return the value of the property with the given ID.

Syntax	int scGetProperty(long ID);	
Parameters	ID	Value for the the property to read, the following values are available: 0. **Use Paper Size** defintion, if available, from input file. 1. **Background color**. Set the active background color to use for the conversion methods. 2. **Adjust colors**.If enabled, a color that's too close to he background color, will be inverted. For example a white line on white background will be drawn with black color if this option is enabled. 3. **Monochrome mode**. If set to 1 then monochrome (black & white) conversion is enabled. 4. **PDF output standard selection**. See scSetPDFWriteFormat function for information. 5. **PDF large format mode**. See scSetPDFLargeFormat function for information. 6. **PDF Layers**. See the scSetPDFLayers function for information. 7. **PDF monochrome raster compression**. See the PDFMonoRasterCompression property for information. 8. **TIFF single strip mode**. See the scSetTIFFSingleStrip function for information. 9. **Use RO instruction**. See the scUseROInstruction function for information. 10. **Load DWF Markup.** If set to 1 markup created by Autodesk Design Review will be loaded and included in the conversion. Set this property to 0 to ignore markup created by Autodesk Design Review. 11. **PDF Searchable**. See the PDFSearchable property for more information. 12. **DXF Loadunit**. Not used. 13. **PDF to CAD Rotation**. If set to 1 the PDF to CAD conversion will honor PDF page rotation if present. 14. **DXF Ignore White Areas**. If set to non-zero, all filled white areas will be ignored when a DXF file is written. This may be useful if you're converting PDF files to DXF because white areas may give problems in AutoCAD. 15. **DXF Write Metric**. If set to 1 DXF files will be written using millimeters, else Inches will be used. 16. **PDF Transparency**. Enable or disable use of transparency in exported PDF files.Some printer drivers doesn't support transparency, and turning off

		transparency for exported PDF files may make your files print faster. 17. **Grayscale mode**. If set to 1 then grayscale conversion is enabled. 18. **DXF Force Line to LWPolyline**. If set to 1 all line elements will be written as LWPolyline instead of Line. This will increase file size, but makes it possible to write lines with width. 19. **Use DWF Paper Size**. If set to 1 (true) all DWF files will be loaded using paper size if this is available, else the file will be loaded with drawing extents. 20. **DWF Exporter DPI**. Resolution to use for exported DWF files. 21. **DXF Extract Images**. If set to 1 all images in the original file will be extracted to PNG files located in the same folder as the exported DXF file. 22. **PDF True Color Raster Compression**. See the PDFTrueColorRasterCompression property for information. 23. **PDF create bookmarks for named views.** If enabled a bookmark will be created for each named view in the original file (for example DWF). 24. **Load Hidden Text from DWF Files.** If set to 1 (true) hidden text in DWF files will be loaded and included when converting to PDF. Set to 0 to ignore hidden text in DWF files. 25. **Measurement Viewports**. If this property is enabled (non-zero value) all DWF viewports with measurement settings will be included when PDF files are created. If set to zero, such viewports will be ignored.
Returns	Int	This function returns the requested property value.

2.3.27 scImportersPath

Set path to where the additional file importers are located.

ANSI	void scImportersPathA(char *Path);	
UNICODE	void scImportersPath(WCHAR *Path);	
Parameters	InputFile	Name of pentable file to load.
Returns	int	This function returns zero on success.

2.3.28 scLoadPentable

Load a pen table file that will be used for conversions.

ANSI	int scLoadPenTableA(char *PenTableFile);	
UNICODE	int scLoadPenTable(WCHAR *PenTableFile);	
Parameters	PenTableFile	Full path to pen table file.
Returns	int	This function returns zero on success.

2.3.29 scOCRAvailable

Check if the Tesseract OCR engine is installed on the system. If Tesseract is installed you can use the scCreateSearchablePDF method to create searchable PDF files from scanned PDF, TIFF, PNG and other file formats.
See appendix H for more information about Tesseract.

Syntax	int scOCRAvailable(long* OCRAvailable);	
Parameters	OCRAvailable	Will be set to non-zero if Tesseract is installed.
Returns	int	This function returns zero on success.

2.3.30 scPDFConform

Conform, or convert, an existing PDF file to one of the supported PDF/A standards.

ANSI	int scPDFConformA (char *SerialNumber, char *InputFile, int ConformType, char *OutputFile, BSTR *Errors);	
UNICODE	int scPDConform (WCHAR *SerialNumber, WCHAR *InputFile, int ConformType, WCHAR *OutputFile, BSTR *Errors);	
Parameters	SerialNumber	Your serial number.
	InputFile	The PDF file you want to conform to the selected PDF/A standard.
	ConformType	The conformation standard to use. Use one of the following values:
	OutputFile	Conformed output file.
	Errors	This parameter will contain one or more error messages if the conformation fail. If you call this function from C/C++ you will need to free the BSTR by calling **SysFreeString**(Errors) when it's no longer needed.
Returns	int	This function returns zero on success.

Value	**Standard**
0	Normalize
1	PDF/A-1b
2	PDF/A-2b
3	PDF/A-3b
4	PDF/A-2u
5	PDF/A-3u
6	PDF/A-4
7	PDF/A-4e

2.3.31 scPDFEncrypt

Encrypt a PDF file using password(s) and optional restriction settings.

ANSI	int PDFEncryptA(char *SerialNumber, char *OriginalFile, char *EncryptedFile, char *OpenPassword, char *OwnerPassword, long Restrictions);
UNICODE	int PDFEncrypt(WCHAR *SerialNumber, WCHAR *OriginalFile,

<table>
<tr><td></td><td colspan="2">WCHAR *EncryptedFile, WCHAR *OpenPassword,
WCHAR *OwnerPassword, long Restrictions);</td></tr>
<tr><td>Parameters</td><td>SerialNumber</td><td>Your serial number.</td></tr>
<tr><td></td><td>OriginalFile</td><td>The PDF file you want to create an encrypted copy of.</td></tr>
<tr><td></td><td>EncryptedFile</td><td>The encrypted PDF file. This will be an exact copy of the orignal file but encrypted.</td></tr>
<tr><td></td><td>OpenPassword</td><td>Optional password required to open the file.</td></tr>
<tr><td></td><td>OwnerPassword</td><td>Optional owner password.</td></tr>
<tr><td></td><td>Restrictions</td><td>Set optional user restrictions for the encrypted PDF file.
The following values are available:

<table><tr><th>Value</th><th>Restriction</th></tr><tr><td>0</td><td>No restrictions.</td></tr><tr><td>4</td><td>Deny printing.</td></tr><tr><td>8</td><td>Deny modification of contents.</td></tr><tr><td>16</td><td>Deny copying of contents.</td></tr><tr><td>32</td><td>Deny commenting.</td></tr><tr><td>3900</td><td>Deny all.</td></tr></table>
The flags can be combined, for example you may set restrictions to 20 (4+16) to deny both printing and copying.</td></tr>
<tr><td>Returns</td><td>int</td><td>This function returns zero on success.</td></tr>
</table>

2.3.32 scPDFFlattenAnnotations

Flatten either all or some annotations and create a new PDF file.

<table>
<tr><td>ANSI</td><td colspan="2">int PDFFlattenAnnotationsA (char *InputFile, char *OutputFile,
long Flags);</td></tr>
<tr><td>UNICODE</td><td colspan="2">int PDFFlattenAnnotations (WCHAR *InputFile WCHAR *OutputFile,
long Flags);</td></tr>
<tr><td>Parameters</td><td>InputFile</td><td>The original PDF file with annotations which you want to create a flattened copy of.</td></tr>
<tr><td></td><td>OutputFile</td><td>The new flattened PDF file.</td></tr>
<tr><td></td><td>Flags</td><td>Control how annotations should be flattened.
The following flattening flag values are available:
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>All annotations which have an appearance stream and which have the print flag set are flattened</td></tr><tr><td>1</td><td>Only visible annotations will be flattened.</td></tr><tr><td>2</td><td>Only markup annotations are flattened. Link, Sound, or FileAttach annotations are not markup annotations and such types will be left intact</td></tr><tr><td>4</td><td>Flatten all annotations which are not supported in PDF/A 1.</td></tr><tr><td>8</td><td>Flatten all annotations which are not</td></tr></table></td></tr>
</table>

		<table><tr><td></td><td>supported in PDF/A 2 or 3.</td></tr><tr><td>16</td><td>Form fields will be flattened.</td></tr><tr><td>4096</td><td>Link annotations will be kept.</td></tr><tr><td>8192</td><td>File attachment annotations will be kept.</td></tr><tr><td>16384</td><td>Text annotations will be kept.</td></tr></table> The flags may be combined.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.information.

2.3.33 scPDFForceOrientation

Create a copy of the input PDF file where all pages are forced (if required) to the given orientation. Pages that already are using the given orientation will not be modified. The selected pages, defined by the Pages parameter, will be rotated using the given rotation factor.

Syntax	HRESULT PDFForceOrientation(BSTR InputFile, BSTR OutputFile, BSTR Pages, LONG Orientation);	
Parameters	InputFile	Full path name for input PDF file that will be modified and copied to the output file.
	OutputFile	Full path name for rotated output file.
	Pages	String describing the pages that are to be modified. Set string to empty or use "-1" for all pages. Specify page indices as comma-separated values or ranges to process (e.g. "1, 2,3,7" or "1, 2, 3-7"). For this function the first-page index is 1.
	Orientation	The selected pages will be be forced to the given orientation. The given orientation is either Portrait or Landscape.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.information.

2.3.34 scPDFMergeAddFile

Add a PDF file to the currently merged PDF file. If you pass a file that is not a PDF file, but using another of the supported formats, it will be converted to PDF before being merged. All pages from the given file will be added to the merged output file.
scPDFMergeInit must be called before any file is added.

ANSI	int scPDFMergeAddFileA(char *PDFFileName);	
UNICODE	int scPDFMergeAddFile(WCHAR *PDFFileName);	
Parameters	PDFFileName	Full path name to the PDF file to add to the merged PDF.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.information.

2.3.35 scPDFMergeAddFileEx

Add a PDF file to the currently merged PDF file. If you pass a file that is not a PDF file, but using another of the supported formats, it will be converted to PDF before being merged.
The PageInformation parameter controls which pages from the given file that will be added to the merged output file.
scPDFMergeInit must be called before any file is added.

ANSI	int PDFMergeAddFileExA(char *PDFFileName, char *PageInformation);	
UNICODE	int PDFMergeAddFileEx(WCHAR *PDFFileName, WCHAR *PageInformation);	
Parameters	PDFFileName	Full path name to the PDF file that will be added to the currently merged PDF.
	PageInformation	Control which pages that should be imported from the given PDF file. Use ";" to separate pages, e.g.: setting PageInformation to "1,2,10,11" will import pages 1,2,10 and 11 and add them to the merged PDF file.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.36 scPDFMergeClose

Close the currently merged PDF file and output all pages to given file name.

ANSI	int scPDFMergeCloseA(char *PDFOutputName);	
UNICODE	int scPDFMergeClose(WCHAR *PDFOutputName);	
Parameters	PDFOutputName	Full path name of the new PDF file to create.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.37 scPDFMergeInit

Start a new empty PDF file prepared for merging.
Use scPDFMergeAddFile or scPDFMergeAddFileEx to add files, and finally call scPDFMergeClose to complete the merge.

ANSI	int scPDFMergeInitA(char *SerialNumber);	
UNICODE	int scPDFMergeInit(WCHAR *SerialNumber);	
Parameters	SerialNumber	Your serial number
Returns	Int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.38 scPDFOptimize

Optimize a PDF file for faster loading and rendering. The optimize process will rescale images and rebuild the structure of the input file. The function rebuilds the content streams of all pages, templates and annotations. Useless operators as well as errors in content streams will be removed. The resulting content streams are error free and usually smaller. How much optimization takes effect depends on the quality of the original PDF file.

<table>
<tr><td>ANSI</td><td colspan="2">int scPDFOptimizeA(char *SerialNumber, char *InputFile,
char *OutputFile, long MonoDPI, long ColorDPI,
long MonoCompression, long lColorCompression);</td></tr>
<tr><td>UNICODE</td><td colspan="2">int scPDFOptimize(WCHAR *SerialNumber, WCHAR *InputFile,
WCHAR *OutputFile, long MonoDPI, long ColorDPI,
long MonoCompression, long lColorCompression);</td></tr>
<tr><td>Parameters</td><td>SerialNumber</td><td>Your serial number.</td></tr>
<tr><td></td><td>InputFile</td><td>The source PDF file you want to optimize.</td></tr>
<tr><td></td><td>OutputFile</td><td>Optimized output file.</td></tr>
<tr><td></td><td>MonoDPI</td><td>Resolution to use for monochrome images in output file. Any image larger than the set resolution will be rescaled to fit this setting.</td></tr>
<tr><td></td><td>ColorDPI</td><td>Resolution to use for color images in output file. Any image larger than the set resolution will be rescaled to fit this setting.</td></tr>
<tr><td></td><td>MonoCompression</td><td>Set compression method to use for monochrome images:
<table><tr><th>Value</th><th>Compression method</th></tr><tr><td>0</td><td>Flate Compression (ZIP)</td></tr><tr><td>2</td><td>CCITT Group 3 Fax</td></tr><tr><td>3</td><td>CCITT Group 4 Fax</td></tr><tr><td>8</td><td>JBIG2</td></tr></table></td></tr>
<tr><td></td><td>ColorCompression</td><td>Set compression method to use for color images:
<table><tr><th>Value</th><th>Compression method</th></tr><tr><td>0</td><td>Flate Compression (ZIP)</td></tr><tr><td>1</td><td>JPEG</td></tr><tr><td>4</td><td>LZW</td></tr><tr><td>7</td><td>JPEG2000</td></tr></table></td></tr>
<tr><td>Returns</td><td>int</td><td>Non-zero if optimization was successful.</td></tr>
</table>

2.3.39 scPDFRotatePages

Create a rotated copy of the input PDF file. The selected pages, defined by the Pages parameter, will be rotated using the given rotation factor.

ANSI	int scPDFRotatePagesA(char *InputFile, char *OutputFile, char *Pages, LONG Rotation);

UNICODE	Int scPDFRotatePages(WCHAR *InputFile, WCHAR *OutputFile, WCHAR *Pages, LONG Rotation);	
Parameters	InputFile	Full path name for input PDF file that will be rotated and copied to the output file.
	OutputFile	Full path name for rotated output file.
	Pages	String describing the pages that are to be rotated. Set string to empty or use "-1" for all pages. Specify page indices as comma-separated values or ranges to process (e.g. "1, 2,3,7" or "1, 2, 3-7"). For this function the first-page index is 1.
	Rotation	The selected pages will be rotated using this rotation factor. Allowed values are 0, 180 or 270.
Returns	Integer	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.40 scPDFScale

Create a scaled copy of the input PDF file. Page width will be scaled using ScaleX and height will be scaled using ScaleY.

ANSI	HRESULT PDFScaleA(char *InputFile, char *OutputFile, DOUBLE ScaleX, DOUBLE ScaleY);	
UNICODE	HRESULT PDFScale(WCHAR *InputFile, WCHAR *OutputFile, DOUBLE ScaleX, DOUBLE ScaleY);	
Parameters	InputFile	Full path name for input PDF file that will be scaled and copied to the output file.
	OutputFile	Full path name for scaled output file.
	ScaleX	Scale factor to use for X direction.
	ScaleY	Scale factor to use for Y direction.
Returns	Integer	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

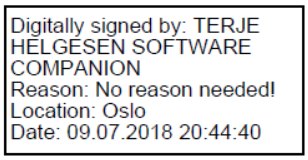
2.3.41 scPDFSignFile

Digitally sign a PDF file and add a signature image. To be able to sign a PDF file with scConverter you will need a certificate stored as a PKCS#12 file (PFX).

Syntax	int scPDFSignFile(WCHAR *SerialNumber, WCHAR *InputFile, WCHAR *OutputFile, LONG Position, WCHAR *ImageFile, BSTR CertFile, BSTR CertPassword, BSTR Reason);	
Parameters	SerialNumber	Your serial number.
	OriginalFile	The PDF file you want to create a signed copy of. If you set this parameter to an empty file name (“”) the currently loaded file will be used as original.
	OutputFile	The signed PDF file. This will be an exact copy of the orignal file but digitally signed.
	Position	The location of the signature image, either given as file or automatically generated. Following options available:

		0. Top left 1. Top center 2. Top right 3. Bottom left 4. Bottom center 5. Bottom right
	ImageFile	Name of the image file to use. This can be a scanned signature or any other image. Supported formats include PNG, JPEG and TIFF. If no image file is provided the control will create an automatic image.
	CertFile	Full path and name for the certificate file (PFX).
	CertPassword	The password to use for the certificate file.
	Reason	A string describing the reason for signing this file (optional).
	Location	A string describing the location (optional).
Returns	Integer	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

Below is an example on how an automatic created signature image may look:



2.3.42 scPDFSplit

Split a multi-page PDF file into smaller files based on the parameters.

ANSI	int scPDFSplitA(char *SerialNumber, char *InputFile, char *OutputFolder, char *FileLabel, long PagesPerFile);	
UNICODE	int scPDFSplit(WCHAR *SerialNumber, WCHAR *InputFile, WCHAR *OutputFolder, WCHAR *FileLabel, long PagesPerFile);	
Parameters	SerialNumber	Your serial number.
	InputFile	The multi-page PDF file that will be split into smaller files.
	OutputFolder	A valid folder name for the output files.
	FileLabel	Optional label to add to output filenames. By default the output files will be named "inputfilename_1.pdf", "inputfilename_2.pdf" and so on. By adding "part" as label, the output files will be named "inputfilename__part_1.pdf", "inputfilename__part_2.pdf" and so on.
	PagesPerFile	Number of pages per output file. If the input document contains 20 pages and you set page per file to 1, the control will create 20 files. If you set pages per file to e.g. 2, the control will output 10 files with 2 pages each.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.43 scPDFToCAD

Convert a PDF file to an editable vector format file, for example Autodesk DXF format.

ANSI	int scPDFToCADA(char *SerialNumber, char *InputFile, long PageNo, char *Format, char *OutputFileName);	
Syntax	int scPDFToCAD(WCHAR *SerialNumber, WCHAR *InputFileName, long Page, WCHAR *Format, WCHAR *OutputFileName);	
Parameters	SerialNumber	Your serial number.
	InputFile	Name of the PDF file to convert.
	Page	PDF page that will converted to vector format. First page is index 0.
	Format	Export format to use for conversion. Supported formats: • CGM (Computer Graphics Metafile) • DWF (Drawing Web Format) • DXF (Autodesk Drawing Exchange Format) • GBR (Gerber RS274X) • PLT (HPGL/2) • SVG (Scalable Vector Format)
	OutputFile	Name of the exported vector file.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.44 scResizeImage

Resize an image file using one of two available high quality scaling algorithms. The width and height parameters are used to determine the new size, but aspect ratio will be maintained.

ANSI	int scResizeImageA(const char* SerialNumber, const char* InputFile, const char* OutputFile, const char* Format, long Method, long Width, long Height);	
UNICODE	int scResizeImage(const WCHAR* SerialNumber, const WCHAR* InputFile, const WCHAR* OutputFile, const WCHAR *Format, long Method, long Width, long Height);	
Parameters	InputFile	Input image file to resize.
	OutputFile	Name of resized output file.
	Format	File format to use for output file. This string have to be one of the following: • BMP • HEIC • JPEG • JXL • PDF • PNG • TIFF • WEBP
	Method	Rescale algorithm selection. Following values are supported: 0. Avir

		1. Lanczos
	Width	Maxmium width of resized image.
	Height	Maxmium height of resized image.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

Notes:
c) Avir information: https://github.com/avaneev/avir
d) Lanczos information: https://en.wikipedia.org/wiki/Lanczos_resampling

2.3.45 scResizeImage2

Resize an image file using one of the available high quality scaling algorithms. The width and height parameters are used to determine the new size, but the aspect ratio will be maintained.

UNICODE	int scResizeImage2(const WCHAR* SerialNumber, HGLOBAL hDIB, long Method, long Width, long Height, HGLOBAL *DIB);	
Parameters	hDIB	HGLOBAL with DIB data to rescale.
	Method	Rescale algorithm selection. Following values are supported: 0. Avir 1. Lanczos See ResizeImage description above for more information about the different algorithms.
	Width	Maxmium width of resized image.
	Height	Maxmium height of resized image.
	DIB	Returned resized DIB data.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.46 scSaveCompareResult

Compare two files and save the result as an image file. If you set BgColor to red and FgColor to blue all information that is equal in both files will be displayed as black. Information that has been deleted will be displayed using red, and new additions are displayed using blue color.

ANSI	int SaveCompareResultA(char *BgFile, char *FgFile, char *FileOutput, char *Format, COLORREF BgColor, COLORREF FgColor, LONG ExportDPI);	
UNICODE	int SaveCompareResult(WCHAR *BgFile, WCHAR *FgFile, WCHAR *FileOutput, WCHAR *Format, COLORREF BgColor, COLORREF FgColor, LONG ExportDPI);	
Parameters	BgFile	Name of the background file to use in the comparison.
	FgFile	Name of the foreground file to use in the comparison.
	FileOutput	Name of the output file.

	Format	The file format to use for the saved image file. The following format identifiers are supported: • JPEG • PDF • PNG • TIFF
	BgColor	Color reference to use for the background file.
	FgColor	Color reference to use for the foreground file.
	ExportDPI	Resolution to use for the saved image file. Higher values give better result but consumes more memory and file size increases.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.47 scSaveCompareResultEx

Compare two files and save the result as an image file. If you set BackColor to red and FrontColor to blue all information that is equal in both files will be displayed using EqualColor. Information that has been deleted will be displayed using red, and new additions are displayed using blue color.

ANSI	int scSaveCompareResultEx(char *BackFile, char *FrontFile, char *OutputFile, char *Format, COLORREF BgColor, COLORREF FgColor, COLORREF EqualColor, LONG ExportDPI);	
UNICODE	int scSaveCompareResultEx(WCHAR *BackFile, WCHAR *FrontFile, WCHAR *OutputFile, WCHAR *Format, COLORREF BgColor, COLORREF FgColor, COLORREF EqualColor, LONG ExportDPI);	
Parameters	BackFile	Name of the background file to use in the comparison.
	FrontFile	Name of the foreground file to use in the comparison.
	OutputFile	Name of the output file.
	Format	The file format to use for the saved image file. The following format identifiers are supported: • JPEG • PDF • PNG • TIFF
	BgColor	Color to use for the background file.
	FgColor	Color to use for the foreground file.
	EqualColor	Color to use for information that are equal in both files. If you set this color to white, only the differences between the files will be visible.
	ExportDPI	Resolution to use for the saved image file. Higher values give better result but consumes more memory and the file size increases.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.48 scSaveQRImage

Create a QR barcode image based on input settings and save as a PNG file.

Syntax	Int scSaveQRImage(WCHAR *Text, LONG Size, LONG Margin, LONG ErrorLevel, WCHAR *ImageFile);	
Parameters	Text	Text string to encode as QR.
	Size	Set to 1 for default size. Setting size to 2 will create an image double as large.
	Margin	Optional white margin around QR image in pixels.
	ErrorLevel	Error level to use for QR generation.Default is 0.
	ImageFile	Full path and name of output image file (PNG).
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.49 scSetGerberApertureFileName

Set filename for a Gerber aperture table to be used when loading RS274D Gerber data.

ANSI	void scSetGerberApertureFileNameA(char *ApertureFile);	
UNICODE	void scSetGerberApertureFileName(WCHAR *ApertureFile);	
Parameters	ApertureFile	Full path to aperture file that will be used when RS-274D, also called legacy Gerber, files are loaded.
Returns	void	

2.3.50 scSetGerberFormatSettings

Configure the settings that are required to load RS-274D Gerber files correctly.

Syntax	void scSetGerberFormatSettings(BOOL Incremental, long NumDigits, long NumDecimals, LONG Units, BOOL LeadingZeros, BOOL TrailingZeros);	
Parameters	Incremental	If true incremental coordinates are used. Set to false for absolute coordinates.
	Preceding	Number of digits before decimal point.
	Succeeding	Number of digits after decimal point.
	Units	Units used for coordinates. Set to 0 for Inch and 1 for Millimeters.
	Leading	Leading zeros are suppressed if set to true.
	Trailing	Trailing zeros are suppressed if set to true.
Returns	void	No return value.

2.3.51 scSetLoadPassword

Set password to use for loading encrypted (password protected) files. Password is currently supported for encrypted DWF, DWFX and PDF files.

ANSI	void scSetLoadPasswordA(char *Password);	
UNICODE	void scSetLoadPassword(WCHAR *Password);	
Parameters	Password	Set password to use when loading password protected files.
Returns	void	No return value.

2.3.52 scSetConfigValue

This method can be used to customize the behavior for different functions and operations.
Please see SetConfigValue for a description of the available settings.

Syntax	int scSetConfigValue(long ID, long Value);	
Parameters	ID	Identifier for the setting you want to change.
	Value	Value to set for the given setting.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.53 scSetGrayscale

Enables or disables grayscale conversion.

Syntax	void scSetGrayscale (long Enable);	
Parameters	Enable	Set to non-zero (true) to enable grayscale conversion.
Returns	void	No return value.

2.3.54 scSetMargins

Add margins to the converted file.

Syntax	void SetMargins(BOOL AddMargins, DOUBLE Left, DOUBLE Top, DOUBLE Right, DOUBLE Bottom);	
Parameters	AddMargins	If TRUE margins will be added to the converted file. Set to FALSE to disable margins.
	Left	Set left margin in millimeters.
	Top	Set top margin in millimeters.
	Right	Set right margin in millimeters.
	Bottom	Set bottom margin in millimeters.
Returns	void	No return value.

2.3.55 scSetMonochrome

Enables or disables monochrome (black & white) conversion.

Syntax	void scSetMonochrome(long Enable);	
Parameters	Enable	Set to non-zero (true) to enable monochrome conversion.
Returns	void	No return value.

2.3.56 scSetPCBLayerVisible

Toggle the visibility for a layer previously added using scAddPCBLayer.

Syntax	int scSetPCBLayerVisible(long Index, BOOL Visible)	
Parameters	Index	Index to a layer previously added using the scAddPCBLayer method.
	Visible	Enable or disable visibility.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.57 scSetPDFCustomProperty

Add a custom PDF property to the next converted PDF file. Both the key and the value are user defined.

ANSI	int SetPDFPropertyA(char *Key, char *Value);	
UNICODE	int SetPDFProperty(WCHAR *Key, WCHAR *Value);	
Parameters	Key	A custom key name.
	Value	A custom value be attached to the given key name.
Returns	Int	Returns 0 if success.

2.3.58 scSetPDFLargeFormat

Enables or disables creation of large format PDF files (up to 60x60 meters).

Syntax	void scSetPDFLargeFormat(long Enable);	
Parameters	Enable	If set to non-zero large format PDF files are supported.
Returns	void	No return value.

2.3.59 scSetPDFLayers

Enable or disable creation of layered PDF files.

Syntax	void scSetPDFLargeFormat(long Enable);	
Parameters	Enable	Enable or disable creation of layered PDF files. If set to non-zero, layers and pens in input files will be converted

		to individual PDF layers.
Returns	void	No return value.

2.3.60 scSetPDFProperty

Set a PDF property with the given Key to the given string value.

ANSI	long scPDFPropertyA(long Key, char * Value);	
UNICODE	long scPDFProperty(long Key, WCHAR * Value);	
Parameters	PropertyID	Please see SetPDFProperty for a list of available key numbers.
	Value	Value to set for given property.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.61 scSetPDFWriteFormat

Select PDF standard to use for created PDF files.

Syntax	void scSetPDFWriteFormat(long FormatIdentifier);	
Parameters	FormatIdentifier	Please see the PDFWriteFormat property description for a list of supported values.
Returns	void	No return value.

2.3.62 scSetPLTPaperSize

Define custom paper size to use for converted HPGL/2 files. To use custom paper size you will need to call SetConfigValue(1,1) before the conversion.

Syntax	int scSetPLTPaperSize(LONG Unit, double Width, double Length);	
Parameters	Unit	Setup units to use for the paper size (width/length): 0. Millimeters 1. Inch 2. HPGL/2 units (1016 DPI)
	Width	Width of paper.
	Length	Length, or height, of paper.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.63 scSetPenTableEntry

Set width, color and styles for the given pen table entry. Pen tables can be used to make, for example, thin lines thicker by setting a pen entry to a specific width. Pen tables are support for most CAD formats, which include DWF, CGM and PLT.

Syntax	int scSetPenTableEntry(long Pen, double Width, long Color,

	long LineStyle, long EndStyle);	
Parameters	Pen	Pen number to modify.
	Width	Pen width in millimeters. Set width to 0.0 if you want to keep the line width as defined in the file.
	Color	Pen Color (Windows COLORREF). When specifying an RGB color, the color value has the following hexadecimal form: 0x00bbggrr. Set a color 0xFFFFFFFF to use the color as defined in the file.
	Style	Windows GDI pen style. The following styles are supported: #define PS_SOLID 0 #define PS_DASH 1 /* ------- */ #define PS_DOT 2 /* ..... */ #define PS_DASHDOT 3 /* _._._ */ #define PS_DASHDOTDOT 4 /* _.._.._ */ Set to -1 to use the style defined in the file.
	Endstyle	The following line end styles are supported: 3. Round style (same as GDI PS_ENDCAP_ROUND) 4. Butt style (same as GDI PS_ENDCAP_FLAT) 5. Square style (same as GDI PS_ENDCAP_SQUARE) Set to -1 to use the style defined in the file
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.64 scSetProperty

Set a property with the given ID to the given value.

Syntax	long scSetProperty(long PropertyID, long Value);	
Parameters	PropertyID	Please see scGetProperty for a list of available ID's.
	Value	Value to set for given property.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.65 scSetPropertyString

Set a property string with the given ID to the given value.

ANSI	long scSetPropertyA(long PropertyID, char *Value);	
UNICODE	long scSetProperty(long PropertyID, WCHAR *Value);	
Parameters	PropertyID	Value for the the property to read, the following values are available: 4. **Word converter name.** Set the filename of the Word to PDF converter to use, for example "scWordToPDF.exe". 5. **Excel converter name**. Set the filename of the Excel to PDF converter to use, for example "scExcelToPDF.exe".

		6. **Powerpoint converter name**. Set the filename of the Powerpoint to PDF converter to use, for example "scPPTtoPDF.exe". 7. **Chinese PDF font.** Set the facename to use when creating searchable PDF files with Chinese text. Default value is "SimSun".
	Value	String value to set for given property.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.66 scSetTIFFCompression

Set compression types to use for exported TIFF files.
Please see SetTIFFCompression for a list of supported compression types.

Syntax	int scSetTIFFCompression(LONG Monochrome, LONG Color, LONG TrueColor);	
Parameters	Monochrome	Set compression type to use for monochrome, 1-bit, files.
	Color	Set compression type to use for color files, 4-bit or 8-bit, files
	TrueColor	Set compression type to use for true color, 24-bit, files.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.67 scSetTIFFSingleStrip

Enable or disable creation of single-strip TIFF files. By default, the component will create multi-strip files. Single-strip files are usually smaller (in terms of file size) than multi-strip files. A multi-strip file may, however, have some advantages (depends on application):

- Less memory can be used to process the image because an application can process one strip at a time.
- Random access to image data is less complicated.

Syntax	void scSetTIFFSingleStrip(long Enable);	
Parameters	Enable	Set to non-zero to create single-strip TIFF files.
Returns	void	No return value.

2.3.68 scSetTIFFTag

Override one of the available TIFF tags to the given value (strings only). These tags will be written to all TIFF files created by the control.

ANSI	long scSetTIFFTagA(long Index, char * Value);	
UNICODE	long scTIFFTag (long Index, WCHAR * Value);	
Parameters	Index	Please see SetTIFFTag for a list of available key

		numbers.
	Value	Value to set for given property.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.69 scStartPCB

Start a new empty PCB. Use the scAddPCBLayer function to add Gerber, Excel and HPGL/2 files as layers to this PCB. When all required layers are added you may use the scConvertPCBLayersToCAD function to create a PDF, DXF or SVG output file based on the added files.

Syntax	long scStartPCB(WCHAR *SerialNumber);	
Parameters	SerialNumber	Your serial number.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

2.3.70 scUsePenTable

Enable or disable pen table settings. If set to non-zero, the converted files be converted using the current pen table settings. Pen tables are supported for PLT (HPGL, HPGL/2 and Calcomp), CGM and DWF file formats.

Syntax	void scUsePenTable(long Enable);	
Parameters	Enable	Enable or disable pen table settings.
Returns	void	No return value.

2.3.71 scUseROInstruction

Enable or disable use of RO (rotation) instruction in HPGL/2 files.

Syntax	void scUseROInstruction(long Enable);	
Parameters	Enable	If set to non-zero if RO instruction, if present in the input file, will be used during conversion.
Returns	void	No return value.

2.3.72 scTIFFSplit

Split a multi-page TIFF file into multiple single paged files (one per page).

ANSI	int TIFFSplitA(char *SerialNumber, char *InputFile, char *OutputFolder, char *FileLabel);	
UNICODE	int TIFFSplit(WCHAR *SerialNumber, WCHAR *InputFile, WCHAR *OutputFolder, WCHAR *FileLabel);	
Parameters	SerialNumber	Your serial number.
	InputFile	The multi-page PDF file that will be split into smaller

		files.
	OutputFolder	A valid folder name for the output files.
	FileLabel	Optional label to add to output filenames. By default the output files will be named "inputfilename_1.tif", "inputfilename_2.tif" and so on. By adding "part" as label, the output files will be named "inputfilename__part_1.tif", "inputfilename__part_2.tif" and so on.
Returns	int	Returns 0 if successful. Any other value indicates an error, see GetLastError for more information.

3 Appendixes

In the Appendixes section you will find additional information about scConverter.

3.1 Appendix A System Requirements

Supported Operating Systems

- Windows 7, 32 and 64 bit.
- Windows 8, 32 and 64 bit.
- Windows 8.1, 32 and 64 bit.
- Windows 10, 32 and 64 bit.
- Windows 11
- Windows Server 2008R2
- Windows Server 2012
- Windows Server 2016
- Windows Server 2019
- Windows Server 2022
- Windows Server 2025
- Supports Terminal Server and Citrix.
-

Minimum System Requirements

Display
1024 x 768

Processor
2 GHz

Memory
64-bit Windows: 8 GB
32-bit Windows: 4 GB

Hard Disk / SSD
500MB available space

3.2 Appendix B Supported File Formats

3.2.1 *Input Formats*

scConverter can open the following formats for conversion:

Format Description	Extension(s) used by Format
Adobe PDF	PDF
Autodesk DWF 2D	DWF
Autodesk DWFx 2D	DWFx
Autodesk Inventor 2D[4]	IDW
AVIF Image Format	AVIF
Calcomp Plotter Format	PLT, 907
CALS Type 1 CCITTG-4 Raster	CAL, CG4
Computer Graphics Metafile [1]	CGM
EDMICS C4 Tiled Raster	CG4, TG4
Excellon Drill Format	DRL
Gerber RS-274D, RS-274X, X2	GBR, GBX, many different
Graphics Interchange Format	GIF
High Efficiency Image File Format	HEIC
HPGL Plotter Format	PLT, HP, Many different
HPGL/2 Plotter Format	PLT, HP2, Many different
HP-RTL Plotter Format	PLT, Many different
Intergraph Raster Format (type 9, 24, 27 and 65)	CIT, TG4, COT, RLE, RGB
JPEG Image Format	JPG
JPEG2000 Image Format	JP2, J2K
JPEG-XL	JXL
KIPGL Plotter Format	PLT, HP2, Many different
PNG Image Format	PNG
PSD Adobe Photoshop	PSD
SVG [3]	SVG
TIFF Image Format [2]	TIF, TIFF
WEBP Google Image Format	WEBP
Windows Bitmap Format	BMP

Notes:
1. CGM Binary and Clear Text encodings are supported.
2. Supported TIFF compression methods: Uncompressed, Packbits, LZW, Inflate, JPEG, CCITT G3 and G4.
3. SVG can only be converted to an image format (e.g.: PNG).
4. Requires that Autodesk Apprentice Server are installed on the system. See Appendix C.

Optional Office Formats

If the target system has Microsoft Office 2007 or newer installed, you can in addition use scConverter to convert the following file formats:

Format Description	Extension(s) used by Format
Microsoft Word	DOC, DOCX. RTF, TXT
OpenDocument	ODT
Microsoft Excel	XLS, XLSX
Microsoft PowerPoint	PPT, PPTX

Note that you may use LibreOffice instead of Microsoft Office to enable conversion of Office formats. Just install LibreOffice and scConverter will automatically use it if MS Office is not present.

3.2.2 *Output Formats*

scConverter can convert to the following output formats:

Format Description	**Default Extension**
Adobe PDF and PDF/A	PDF
Adobe Postscript	PS
Autodesk DWF 2D	DWF
Autodesk DXF	DXF
AVIF Image Format	AVIF
CALS Type 1 CCITTG-4 Raster [1]	CAL
Computer Graphics Metafile	CGM
Excellon Drill Format	DRL
Gerber RS-274X	GBX
High Efficiency Image File Format	HEIC
HPGL/2 Plotter Format	PLT
HP-RTL Plotter Format	PLT
JPEG Image Format	JPG
JPEG2000 Image Format	JP2
JPEG-XL Image Format	JXL
Paintbrush PCX Format	PCX
Portable Network Graphics (PNG)	PNG
Tagged Image File Format (TIFF) [2]	TIF
WEBP Google Image Format	WEBP
Windows Bitmap Format	BMP
Windows Enhanced Metafile	EMF
Windows Metafile	WMF

Notes:
1. MIL-PRF-28002B Type 1.
2. Supported compression methods for TIFF writer include Uncompressed, CCITT-G3, CCITT-G4, LZW, Inflate, JPEG and Packbits.

3.3 Appendix C Redistributables

You will find all the files you need to redistribute with your application under a folder named **"C:\Users\Public\Documents\ScConverter SDK\redist"**.

If you have installed 32-bit SDK the redistributable are in a subfolder named x86.
If you have installed 64-bit SDK the redistributable are in a subfolder named x64.

None of the support DLL's requires registration on the target system, only the main component, scconverter.dll, needs to be registered if you use the COM interface.
All DLL files should be installed in the same folder as scconverter.dll.
Here is a list of all redistributable files and their purpose:

Name	Description	Optional	Register
scconverter.dll	The converter component	No	Yes
scCoder.dll	Support module	No	No
dynapdf.dll	PDF reader and writer	No	No
scrwAVIF.dll	AVIF importer and exporter	Yes	No
scrdCalcomp.dll	Calcomp Plotter format importer	Yes	No
scrdCGM.dll	CGM importer (Computer Graphics Metafile)	Yes	No
scrdDRL.dll	Excellon drill format importer	Yes	No
scrwDWF.dll	Autodesk DWF and DWFx 2D importer	Yes	No
scwrDWF6.dll	Autodesk DWF exporter	Yes	No
scrwGBX.dll	Gerber RS-274D and RS-274X importer	Yes	No
scrwGIF.dll	GIF importer	Yes	No
scrwHEIC.dll	HEIC importer and exporter	Yes	No
scrdINGR.dll	Intergraph raster file format importer	Yes	No
scrwJPGEG200.dll	JPEG2000 importer and exporter	Yes	No
scrwJXL.dll	JPEG-XL importer and exporter	Yes	No
scrdPSD.dll	PSD importer	Yes	No
scrwWebP.dll	Google WebP importer and exporter	Yes	No

If you want to convert PDF files you will need to install the PDF font resource files. These files are in a subfolder of the redistribution folder named "**Resource**". All files and subfolders to this folder should be installed as subfolders to the folder where you place your copy of scconverter.dll.
For example, if you install scconverter to a folder named "c:\programfiles\scconverter" you must create a subfolder to this folder which is named "c:\programfiles\scconverter\Resource".
This folder must be an exact copy of the "Resource" folder content.
Other files that you may include with your installation:

Name	Description	Optional
sRGB.icc	PDF/A output intent profile.	No
OfficeToPDF.exe	Generic Microsoft Office formats to PDF converter.	Yes

scExcelToPDF.exe	Microsoft Excel to PDF converter.	Yes
scPPTToPDF.exe	Microsoft PowerPoint to PDF converter.	Yes
scIDWToDWF.exe	Autodesk Inventor IDWF to DWF converter.	Yes
Heif.dll	Required if you want to read or write HEIC files.	Yes
Libx265.dll	Required if you want to read or write HEIC files.	Yes
Libde265.dll	Required if you want to read or write HEIC files.	Yes
jxl.dll	Required if you want to read or write JPEG-XL files.	Yes
jxl_cms.dll	Required if you want to read or write JPEG-XL files.	Yes
jxl_threads.dll	Required if you want to read or write JPEG-XL files.	Yes
lcms2.dll	Required if you want to read or write JPEG-XL files.	Yes
brotlicommon.dll	Required if you want to read or write JPEG-XL files.	Yes
brotlidec.dll	Required if you want to read or write JPEG-XL files.	Yes
brotlienc.dll	Required if you want to read or write JPEG-XL files.	Yes
hwy.dll	Required if you want to read or write JPEG-XL files. Only for 64-bit.	Yes

Notes:
Autodesk Inventor IDW support required Autodesk Apprentice Server to be installed.
Download directly from Autodesk here:
https://www.autodesk.com/support/technical

3.4 Appendix D Paper Sizes

The ConvertToPaperSize and ConvertToPaperSizeEx methods accepts the following predefined paper format constants:

Index	Paper size
0	ISO A0 (841 x 1189 mm)
1	ISO A1 (594 x 841 mm)
2	ISO A2 (420 x 594 mm)
3	ISO A3 Extra (322 x 445 mm)
4	ISO A3 (297 x 420 mm)
5	ISO A4 Extra (236 x 322 mm)
6	ISO A4 Plus (210 x 330 mm)
7	ISO A4 (210 x 297 mm)
8	ISO A5 Extra (174 x 235 mm)
9	ISO A5 (148 x 210 mm)
10	ISO A6 (105 x 148 mm)
11	ISO B0 (1000 x 1414 mm)
12	ISO B1 (707 x 1000 mm)
13	ISO B2 (500 x 707 mm)
14	ISO B3 (353 x 500 mm)
15	ISO B4 (250 x 353 mm)
16	ISO B5 Extra (201 x 276 mm)
17	ISO B5 (176 x 250 mm)
18	ISO B6 (125 x 176 mm)
19	ISO C6 (162 x 229 mm)
20	Engineering A (8.5 x 11 inch)
21	Engineering B (11 x 17 inch)
22	Engineering C (17 x 22 inch)
23	Engineering D (22 x 34 inch)
24	Engineering E (34 x 44 inch)
25	Architectural A (9 x 12 inch)
26	Architectural B (12 x 18 inch)
27	Architectural C (18 x 24 inch)
28	Architectural D (24 x 36 inch)
29	Architectural E (36 x 48 inch)
30	Architectural E2(30 x 42 inch)
31	Letter (8.5 x 11 inch)

32	Legal (8.5 x 14 inch)
33	Tabloid (11 x 17 inch)
34	Slide (11 x 7.33 inch)
35	Ledger (17 x 11 inch)
36	Executive (7.25 x 10.5 inch)
37	Statement (5.5 x 8,5 inch)
38	ISO A1F (594 x 1189 mm)
39	ISO A3F (297 x 594 mm)

3.5 Appendix E Merging PDF Files

scConverter makes it easy to merge any number of PDF files into a single multi-page PDF file.
The PDF files may have different page sizes and any number of pages.

To merge the files you will first have to tell scConverter that you want to start a merge by using the **PDFMergeInit**() method.

You may after this call start to add files to the new merged document.
To add all pages from a PDF file you may use:

PDFMergeAdd("c:\mergein\file1.pdf");
This call will include all pages from the file named file1.pdf

To add only certain pages from a file you can use **PDFMergeAddEx**.
Here are some samples on how it's used:

PDFMergeAddEx("c:\mergein\file2.pdf", "1;2;3;");
This call will include page 1,2 and 3 from the file named file2.pdf

PDFMergeAddEx("c:\mergein\file3.pdf", "1;20;)
This call will include page 1 and 20 from the file named file3.pdf

When you have added all the files and pages you need you finally call

PDFMergeClose("c:\mergeout\output.pdf");

All files and pages will be written to new file, in this case, named output.pdf.

Please note that any filename sent to component using the above methods have to include the full path including folder names.

To merge two files and include selected pages, to a new PDF, all you have to do is to add the following 4 lines to your code:

```
sc.PDFMergeStart();
sc.PDFMergeAddEx( "c:\mergein\file1.pdf", "1;2;3;");
sc.PDFMergeAddEx( "c:\mergein\file2.pdf", "1;20;”);
sc.PDFMergeClose("c:\mergeout\output.pdf");
```

If you want to include all pages from the source same documents you can use:

```
sc.PDFMergeStart();
sc.PDFMergeAdd( "c:\mergein\file1.pdf"  );
sc.PDFMergeAdd(  "c:\mergein\file1.pdf" );
sc.PDFMergeClose( "c:\mergeout\output.pdf" );
```

3.6 Appendix F Markup XML Format

Software Companions XML Markup Format Description

The Software Companions XML markup format can be easily created by a third-party application or by using a text editor. By using this format you can create your own markup files from scratch. All the different markup element types available in ScConverter can be created from a XML file. A XML formatted markup file may be added calling the **AddFromFile** method. It's also possible to load XML markup data by using the **MarkupCreateFromXML** method. The default extension for a Software Companions XML Markup file is .scmx, but .xml may also be used.

File Format Specification

The file must start with the following header:

```
<?xml version="1.0" encoding="UTF-8"?>
<SCMarkupFormat>
```

Only UTF-8 encoding is currently supported.
The header and elements section must then follow. The layers section may optionally be included.
A complete xml markup file example is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<SCMarkupFormat>
<Header>
   <Unit>mm</Unit>
</Header>
<Layers>
   <layer name="Markup Layer 1" color="#0000FF" enabled='1'/>
   <layer name="Markup Layer Two" color="#00ff00" enabled='1'/>
</Layers>
<Elements>
   <Element>
      <type>Text</type>
      <layer>0</layer>
      <page>0</page>
      <insertx>10</insertx>
      <inserty>30</inserty>
      <user>Peter</user>
      <color>#FF0000</color>
      <text>This is a text</text>
      <rotation>45</rotation>
      <font height='10' facename='Times New Roman'/>
   </Element>
</Elements>
</SCMarkupFormat>
```

This xml markup file defines a single text element. The insert position is 10mm from left and 30 mm from bottom of the drawing. The markup coordinate system origin is always the lower left corner of the drawing. The coordinates can be given in millimeters, inches or native (1016 DPI) values. All color values are coded as red, green and blue intensities in hexadecimal notation (HTML standard - #RRGGBB). The Layers section defines two layers. Please see the different section descriptions below for more information.

Header Section

This section contains settings that are common for all markup elements.
The supported entries in this section are:

Unit

Accepted values:

Value	Description
mm	Coordinates and sizes are given in millimeters.
inch	Coordinates and sizes are given in inches.
mil	Coordinates and sizes are given in 1/1000 inch.
native	Coordinates and sizes are given in native coordinates (1016 DPI).

Layers Section

This section contains layer definitions. Each layer definition can have three different attributes.
The first defined layer will have index 0, the next one 1, etc. The layer index is used when elements are defined.

The description of each attribute is given below:

Attribute Name	Description
Name	Name to use for the layer.
Color	Default color to use for elements placed on this layer.
Enabled	Enter a value different from 0 to make the layer visible.

Sample layer entry:

<Layer name="Markup Layer 1" color="#0000FF" enabled='1'/>

This entry will create a layer that is named "Markup Layer 1", with default color blue and it will be visible.

Elements Section

This section contains one or more Element entries.

```
<Elements>
   <Element>
       //data for first element
   </Element>
   <Element>
       //data for second element
   </Element>
   <Element>
       //data for third element
   </Element>
</Elements>
```

3.6.1 Element Section Keywords

Each element entry defines a new markup element. Some of the entry keywords are common for all element types, and some are type specific.
The following table lists the keywords that are common for all elements:

Keyword	Description
type	Markup element type keyword. The following element types are available: **Arrow** **Barcode** **Circle** **Ellipse** **Erase Polygonal Area** **Erase Rectangular Area** **Eraser** **Line** **Picture** **Polygon** **Polyline** **Rectangle** **Revision Cloud** **Rounded Rectangle** **Rubber Stamp** **Shape** **Symbol** **Text**
handle	A unique element numerical ID to be applied to the element. If not provided the control will allocate an ID.
hyperlink	Contains path to a document or an URL.
layer	Layer index to use for the markup element. If not provided the default layer will be used.
locked	Set the element lock flag. A locked element (non-zero value) cannot be moved or modified.
page	Page index where the markup element will be placed. Page indexes are zero based. If not provided page 0 is assumed. You can set page to -1 to force the markup element to be displayed on all pages.
transparent	If defined, the markup element will be transparent.
user	Name of the user that have created the element. If not provided the Windows user name will be applied to the markups.

3.6.2 Arrow Element

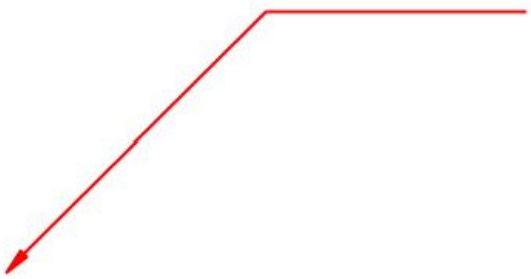
The arrow element requires the following keywords, in addition to the common keywords:

Keyword	Description
points	This section contains one point entry for each point in the arrow polyline. Each point is defined with an x and y attribute. See the xml definition below for an example on how to describe point entries.
color	Define draw color to use for the element.
arrowhead	Define type of arrow header to use. Supported arrow types: 0: open 1: closed 2: filled.
arrowsize	Define size of arrow header.
linewidth	Define width of the arrow lines. The width is defined using the active unit. This entry is optional.

Sample arrow element xml definition:

```
<element>
   <type>Arrow</type>
   <layer>0</layer>
   <page>0</page>
   <color>#FF0000</color>
   <arrowhead>2</arrowhead>
   <arrowsize>5</arrowsize>
   <linewidth>0.3</linewidth>
   <points>
      <point x='100' y='100'/>
      <point x='150' y='150'/>
      <point x='200' y='150'/>
   </points>
</element>
```

The xml data to the left will create the following arrow element:



3.6.3 Barcode Element

The barcode element requires the following keywords, in addition to the common keywords:

Keyword	Description
center	Barcode center coordinates. The center coordinate is defined by setting x and y to a percentage of the document extents, or absolute values. Sample usages: * * Add the barcode centered at document center. * * Add the barcode horizontally centered and 40 millimeters above the bottom of the page.
size	Set the dimensions for the barcode in percentage of current document extents, or absolute values. The extents are defined using two attributes named w and h. Sample definitions: *<size w="10%" h="10%"/>* Create a barcode with width equal to 10% of document page width, and the height will be 10% of document page height. *<size w="40" h="40"/>* Create a barcode with width and height set to 40 units.
barcodetype	The type of barcode encoder to use. The following barcode standards are supported: • aztec • datamatrix • code128a • code128b • code128c • code39 • code93 • ean13 • pdf417 • qr
text	The text to encode using the selected barcode type.
barcodeincludetext	Include the actual text as part of the barcode image.
Barcodelabel	Optional label text added to the barcode image.

Sample barcode element xml definition:

```
<element>
   <type>Barcode</type>
   <barcodetype>QR</barcodetype>
   <layer>0</layer>
   <page>0</page>
   <center x="50%" y="50"/>
   <size x="40" h="40"/>
   <text>This is a text</text>
</element>
```

The xml data to the left will create the following barcode element:

3.6.4 Circle Element

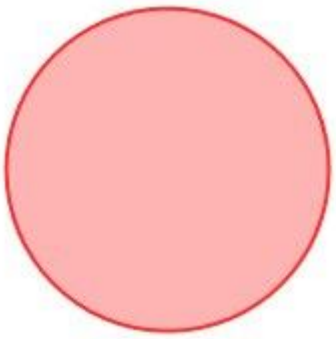
The circle element requires the following keywords, in addition to the common keywords:

Keyword	Description
center	Circle center coordinates. The center coordinate is defined by setting x and y to a percentage of the document extents, or absolute values. Sample usages: * * Add the circle centered at document center. * * Add the circle with center 50 millimeters from the left and 40 millimeters above the bottom of the page.
radius	The radius of circle in active the unit.
fillcolor	Define the fill color to use for the element.
color	Define the outline color to use for the element.
linewidth	Define the width of the outline. The width is defined using the active unit. This entry is optional.
linestyle	Define line style to use for the outline. This entry is optional.
fillstyle	Define fill style to use for the element. This entry is optional.

Sample circle element xml definition:

```
<element>
   <type>Circle</type>
   <center x=”50%” y=”50”/>
   <radius>50</radius>
   <layer>0</layer>
   <page>0</page>
   <color>#FF0000</color>
   <fillcolor>#FFA0A0</fillcolor>
   <fillstyle>1</fillstyle>
   <linewidth>1.0</linewidth>
   <transparent/>
</element>
```

The xml definition to the left will create the following circle:



3.6.5 Erase Polygonal Area Element

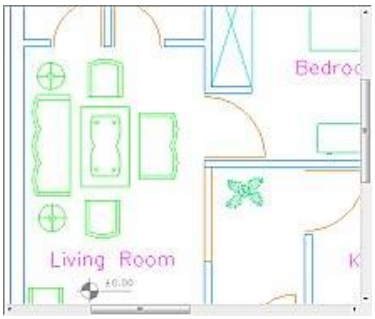
Define an area to hide (erase) using a polygon defintion. This feature is also known as wipeout. The erase polygonal area element requires the following keywords, in addition to the common keywords:

Keyword	Description
points	This section contains one point entry for each point in the polygon. Each point is defined with an x and y attribute. See the xml definition below for an example on how to describe point entries.

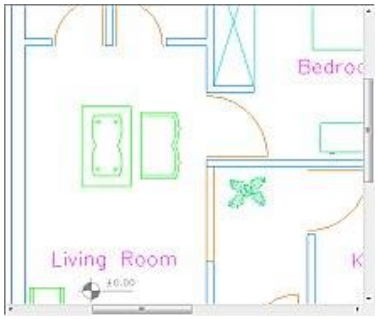
Sample erase polygonal area element xml definition:

```
<element>
   <type>Erase Polygonal Area</type>
   <layer>0</layer>
   <page>0</page>
   <points>
      <point x='150' y='190'/>
      <point x='150' y='290'/>
      <point x='200' y='290'/>
      <point x='200' y='265'/>
      <point x='175' y='265'/>
      <point x='175' y='215'/>
      <point x='200' y='215'/>
      <point x='200' y='190'/>
      <point x='150' y='190'/>
   </points>
</element>
```

The xml data to the left will erase the polygonal area as shown below when used for sample file *compare_revA.plt*:



Before xml markup is loaded



After xml markup is loaded.

3.6.6 Erase Rectangular Area Element

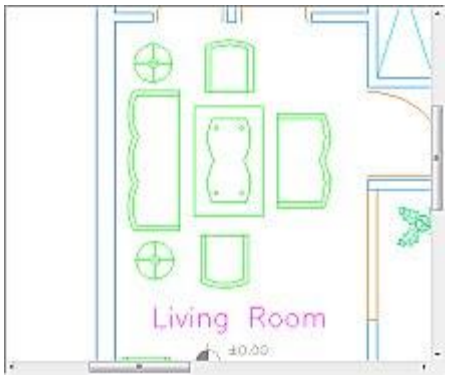
Define an area to hide (erase) using a rectangle. This feature is also known as wipeout. The erase rectangular area element requires the following keywords, in addition to the common keywords:

Keyword	Description
boundary	Define boundary for the created rectangle. The boundary is defined using four attributes. These attributes are named: x1, y1, x2 and y2. Sample boundary entry: *<boundary x1='150' x2='190' y1='230' y2='290'/>*

Sample erase rectangular area element xml definition:

```
<element>
<type>Erase Rectangular Area</type>
   <layer>0</layer>
   <page>0</page>
   <boundary x1='150' y1='190'
    x2='230' y2='290'/>
</element>
```

The xml data to the left will erase the rectangular area as shown below when used for sample file *compare_revA.plt*:



Before xml markup is loaded.

After xml markup is loaded.

.

3.6.7 Eraser Element

Define a polyline that will hide drawing contents. The polyline defined will be drawn using the current background color and with the width defined by the *linewidth* setting. The eraser element requires the following keywords, in addition to the common keywords:

Keyword	Description
points	This section contains one point entry for each point in the polyline. Each point is defined with an x and y attribute. See the xml definition below for an example on how to describe point entries.

Sample eraser element xml definition:

```
<element>
   <type>Eraser</type>
   <layer>0</layer>
   <page>0</page>
   <linewidth>5.0</linewidth>
   <points>
      <point x='150' y='190'/>
      <point x='150' y='290'/>
      <point x='200' y='290'/>
      <point x='200' y='265'/>
      <point x='175' y='265'/>
      <point x='175' y='215'/>
      <point x='200' y='215'/>
      <point x='200' y='190'/>
   </points>
</element>
```

3.6.8 Line Element

The line element uses the following keywords, in addition to the common keywords:

Keyword	Description
boundary	Define boundary for the created line. The boundary is defined using four attributes. These attributes are named: x1, y1, x2 and y2. Sample boundary entry: *<boundary x1='80' x2='180' y1='10' y2='110'/>*. This will draw a line will between x1,y1 and x2,y2.
color	Define line color to use for the element.
linewidth	Define the width of the line in active units. This entry is optional.
linestyle	Define line style to use for the line. This entry is optional.

Sample line element xml definition:

```
<element>
   <type>Line</type>
   <layer>0</layer>
   <page>0</page>
   <boundary x1='80' x2='180' y1='10' y2='110'/>
   <color>#00FF00</color>
   <linewidth>1.0</linewidth>
   <linestyle>0</linestyle>
</element>
```

3.6.9 Picture Element

The picture element requires the following keywords, in addition to the common keywords:

Keyword	Description
center	Picture center coordinates. The center coordinate is defined by setting x and y to a percentage of the document extents, or absolute values. Sample usages: * * Add the picture centered at document center. * * Add the picture horizontally centered and 40 millimeters above the bottom of the page.
size	Set the dimensions for the picture in percentage of current document extents, or absolute values. The extents are defined using two attributes named w and h. Sample definitions: *<size w="10%" h="10%"/>* Add the picture with width equal to 10% of document page width, and the height will be 10% of document page height. *<size w="40" h="40"/>* Add the picture with width and height set to 40 units. If you do not define size the actual picture dimension will be used, based on the image file dpi settings.
filename	Full path to the image file that will be added as markup. Please note that either double back slash (\\), or a single slash (/) must be used. If you skip the directory part and just enter the filename, the control will search for the image file in the same folder as the markup file.

Sample picture element xml definition:

```
<element>
   <type>Picture</type>
   <layer>0</layer>
   <page>0</page>
   <center x="50%" y="50%"/>
   <size w="100" h="30"/>
   <filename>sclogo.png</filename>
</element>
```

The xml data to the left will add the following picture:

3.6.10 Polygon Element

The polygon element uses the following keywords, in addition to the common keywords:

Keyword	Description
points	This section contains one point entry for each point in the polygon. Each point is defined with an x and y attribute. Please see the polygon element definition below for an example on how to describe point entries.
color	Define line color to use for the element.
fillcolor	Define fill color to use for the element.
color	Define outline color to use for the element.
linewidth	Define width of the outline. The width is defined using the active unit. This entry is optional.
linestyle	Define line style to use for the outline. This entry is optional.
fillstyle	Define fill style to use for the element. This entry is optional.

Sample polygon element xml definition:

```
<element>
   <type>Polygon</type>
   <layer>0</layer>
   <page>0</page>
   <color>#FF0000</color>
   <fillcolor>#CCFFCC</fillcolor>
   <linewidth>0</linewidth>
   <fillstyle>1</fillstyle>
      <points>
         <point x='10' y='10'/>
         <point x='20' y='20'/>
         <point x='30' y='10'/>
         <point x='20' y='0'/>
      </points>
</element>
```

The xml data to the left will create the following 4-point polygon:

3.6.11 Polyline Element

The polyline element uses the following keywords, in addition to the common keywords:

Keyword	Description
points	This section contains one point entry for each point in the polyline. Each point is defined with an x and y attribute. See the polyline element definition below for an example on how to describe point entries.
color	Define line color to use for the element.
fillcolor	Define fill color to use for the element.
color	Define outline color to use for the element.
linewidth	Define width of the polyline. The width is defined using the active unit. This entry is optional.
linestyle	Define line style to use for the outline. This entry is optional.
fillstyle	Define fill style to use for the element. This entry is optional.

Sample polyline element xml definition:

```
<element>
   <type>Polyline</type>
   <layer>0</layer>
   <page>0</page>
   <color>#FF0000</color>
   <linewidth>0.5</linewidth>
      <points>
         <point x='10' y='10'/>
         <point x='15' y='15'/>
         <point x='20' y='10'/>
         <point x='25' y='15'/>
      </points>
</element>
```

The xml data to the left will create the following 4-point polyline:

3.6.12 Rectangle Element

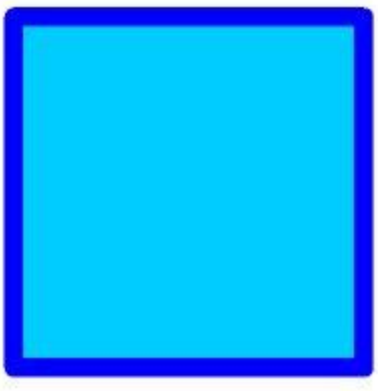
The rectangle element uses the following keywords, in addition to the common keywords:

Keyword	Description
boundary	Define boundary for the created rectangle. The boundary is defined using four attributes. These attributes are named: x1, y1, x2 and y2. Sample boundary entry: *<boundary x1='80' x2='180' y1='10' y2='110'/>*
fillcolor	Define fill color to use for the element.
color	Define outline color to use for the element.
linewidth	Define width of the outline. The width is defined using the active unit. This entry is optional.
linestyle	Define line style to use for the outline. This entry is optional.
fillstyle	Define fill style to use for the element. This entry is optional.

Sample rectangle element xml definition:

```
<element>
   <type>Rectangle</type>
   <layer>0</layer>
   <page>0</page>
   <boundary x1='10' x2='90' y1='10' y2='90'/>
   <color>#0000FF</color>
   <fillcolor>#00CCFF</fillcolor>
   <fillstyle>1</fillstyle>
   <linewidth>0</linewidth>
   <transparent/>
</element>
```

The xml data to the left will create the following rectangle:



3.6.13 Revision Cloud Element

The revision cloud element uses the following keywords, in addition to the common keywords:

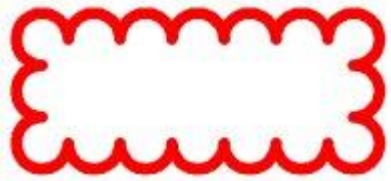
Keyword	Description
boundary	Define boundary for the created revision cloud. The boundary is defined using four attributes. These attributes are named: x1, y1, x2 and y2. Sample boundary entry: *<boundary x1='80' x2='180' y1='10' y2='110'/>*
fillcolor	Define fill color to use for the element.
color	Define outline color to use for the element.
linewidth	Define width of the outline. The width is defined using the active unit. This entry is optional.
linestyle	Define line style to use for the outline. This entry is optional.
fillstyle	Define fill style to use for the element. This entry is optional.
Arclimit	Set the maximum radius for the arcs in the revision cloud. The radius is given in current units. This entry is optional.

s

Sample revision cloud element definition:

```
<element>
   <type>Revision Cloud</type>
   <layer>0</layer>
   <page>0</page>
   <boundary x1='10' x2='110' y1='10' y2='50'/>
   <color>#FF00FF</color>
   <fillstyle>0</fillstyle>
   <linewidth>3</linewidth>
   <transparent/>
</element>
```

The xml data to the left will create the following revision cloud:



3.6.14 Rounded Rectangle Element

The rounded rectangle element uses the following keywords, in addition to the common keywords:

Keyword	Description
boundary	Define boundary for the created rounded rectangle. The boundary is defined by using four attributes. These attributes are named: x1, y1, x2 and y2. Sample boundary entry: *<boundary x1='80' x2='180' y1='10' y2='110'/>*
fillcolor	Define fill color to use for the element.
color	Define outline color to use for the element.
linewidth	Define the width of the outline. The width is defined using the active unit. This entry is optional.
linestyle	Define line style to use for the outline. This entry is optional.
fillstyle	Define fill style to use for the element. This entry is optional.

Sample rectangle element xml definition:

```
<element>
   <type>Rounded Rectangle</type>
   <layer>0</layer>
   <page>0</page>
   <boundary x1='50' x2='90' y1='50' y2='60'/>
   <color>#FF0000</color>
   <fillcolor>#CCCCCC</fillcolor>
   <fillstyle>1</fillstyle>
   <linewidth>2</linewidth>
   <transparent/>
</element>
```

The xml data to the left will create the following rectangle:

3.6.15 Rubber Stamp Element

The rubber stamp element uses the following keywords, in addition to the common keywords:

Keyword	Description
center	Define center point for the rubber stamp in percentage of document extents or using absolute coordinates. The center point is defined using two attributes and these are named x and y. Sample definitions: * * This will place the stamp centered on the document page. * * This will place the stamp with center 100 millimeters from the left and 100 millimeters above the bottom of the page.
size	Set the extents for the rubber stamp in percentage of current document page extents, or absolute values. The extents are defined using two attributes named w and h. Sample definitions: *<size w="20%" h="10%"/>* This will create a stamp with width equal to 20% of document page width, and the height will be 10% of document page height. *<size w="80" h="20"/>* This will create a stamp with a width of 80 units and height of 20 units.
rotation	Define stamp rotation in degrees.
backcolor	Define stamp background color.
textcolor	Define color to use for the stamp text.
text	Stamp text.
font	Define font to use for the stamp text. The font is defined using several different attributes, each of them correspond to the Windows LOGFONT definition. The following font attributes are supported: height - Same as LOGFONT lfHeight. This value will recalculated. orientation - Same as LOGFONT lfOrientation. weight - Same as LOGFONT lfWeight. italic - Same as LOGFONT lfItalic. underline - Same as LOGFONT lfUnderline. strikeout - Same as LOGFONT lfStrikeout. charset - Same as LOGFONT lfCharset. outprecision - Same as LOGFONT lfOutPrecision. clipprecision - Same as LOGFONT lfClipPrecision. quality - Same as LOGFONT lfQuality. pitch - Same as LOGFONT lfPitchAndFamily. facename - Same as LOGFONT lfFaceName. Sample font definition: *<font italic='1' facename='Times New Roman'/>*
linewidth	Define the width of the outline. The width is defined using the active unit. This entry is optional.

The XML description below will add a stamp with its center at document center (50% of the document width, and 50% of the document height).
The stamps width will be 20% of document width, and the height will be 10% of the document height.
The stamp rotation is set to 0 degrees.

Sample rubber stamp element definition:

```
<element>
   <type>Rubber Stamp</type>
   <layer>0</layer>
   <page>0</page>
   <center x="50%" y="50%"/>
   <size w="20%" h="10%"/>
   <text>Draft</text>
   <rotation>0</rotation>
   <textcolor>#FF0000</textcolor>
   <backcolor>#FFCCCC</backcolor>
   <font italic='0'
        facename='Times New Roman'/>
   <transparent/>
</element>
```

The xml data to the left will create the following rubber stamp:

3.6.16 Shape Element

The shape element requires the following keywords, in addition to the common keywords:

Keyword	**Description**
center	Shape center coordinates. The center coordinate is defined by setting x and y to a percentage of the document extents, or by using absolute values. Sample usage: * * Add the shape centered at document center. * * Add the shape 50 millimeters from the left and 40 millimeters above the bottom of the document, if millimeter is the active unit.
shapetype	One of the predefined shape types: 0. Approved/Checkmark 1. Rejected
shapesize	Size of the shape element in current units.

Sample symbol element XML definition:

```
<element>
   <type>Shape</type>
   <layer>0</layer>
   <page>0</page>
   <center x=”50%” y=”50%”/>
   <shapetype>0</shapetype>
   <shapesize>10</shapesize>
</element>
```

The xml data to the left will add the following shape to the file:

3.6.17 Symbol Element

The symbol element requires the following keywords, in addition to the common keywords:

Keyword	**Description**
boundary	Define boundary for the created symbol. The boundary is defined using four attributes. These attributes are named: x1, y1, x2 and y2. Sample boundary entry: *<boundary x1='100' x2='150' y1='200' y2='250'/>*
symbollibrary	The name of the symbol library to use.
symbolname	The name of the symbol to use.

Sample symbol element XML definition:

```
<element>
   <type>Symbol</type>
   <layer>0</layer>
   <page>0</page>
   <boundary x1='100' x2='150' y1='100' y2='150'/>
   <symbollibrary>Sample</symbollibrary>
   <symbolname>Copyright</symbolname>
</element>
```

The xml data to the left will add the following symbol to the file:

3.6.18 Text Element

The text element requires the following keywords, in addition to the common keywords:

Keyword	Description
center	Text center coordinate. Text center coordinate is defined by setting x and y to a percentage of the document extents, or absolute coordinates. Sample usage: * * This will add the text centered at document center. * * This will add 100 millimeters from the left and 20 millimeters above the bottom of the page.
insertx	Text insert x coordinate. This value is the absolute coordinate for the horizontal text origin. You may also use percentage of the document width, for example: *<insertx>25%</insertx>* You should use either text center as described above or use insertx together with inserty to define the text origin.
inserty	Text insert y coordinate. This value is the absolute coordinate for the vertical text origin. You may also use percentage of the document height, for example: *<inserty>50%</inserty>* You should use either text center as described above or use inserty together with insertx to define the text origin.
alignment	The alignment setting defines the origin of the text to insert. Please note that this setting can only be used together with the insertx and insert keywords. The following values are available: 0. Origin is center of the text. 1. Origin is top left of the text. 2. Origin is bottom left of the text. 3. Origin is bottom right of the text. 4. Origin is top right of the text.
rotation	Text rotation in degrees.
textcolor	Define color to use for the text.
text	The text string to display.
font	Define font to use for the text. The font is defined using several different attributes, each of them correspond to the Windows LOGFONT defintition. The following font attributes are supported: height - Same as LOGFONT lfHeight. width - Samer as LOGFONT lfWidth. orientation - Same as LOGFONT lfOrientation. weight - Same as LOGFONT lfWeight. italic - Same as LOGFONT lfItalic. underline - Same as LOGFONT lfUnderline. strikeout - Same as LOGFONT lfStrikeout. charset - Same as LOGFONT lfCharset. outprecision - Same as LOGFONT lfOutPrecision. clipprecision - Same as LOGFONT lfClipPrecision. quality - Same as LOGFONT lfQuality. pitch - Same as LOGFONT lfPitchAndFamily. facename - Same as LOGFONT lfFaceName.

	Sample font defintion: *<font italic='1' height='10' facename='Times New Roman'/>*

Sample text element definition:

```
<element>
   <type>Text</type>
   <layer>0</layer>
   <page>0</page>
   <center x="50%" y="50%"/>
   <textcolor>#FF0000</textcolor>
   <text>This is a text</text>
   <rotation>45</rotation>
   <font height='10'
    facename='Times New Roman'/>
</element>
```

The xml data to the left will create the following text element:

This description will add a red text element with its center at document center and rotated 45 degrees. The text height is set to be 10 units.

3.6.19 Line and Fill Style Settings Used for Element Definitions

Line style

The line style can be defined using one of the following values:
0 Solid line.
1 Dashed line.
2 Dotted line.
3 Dash-dot line.
4 Dash-dot-dot line.

Fill style

The fill style can have one the following values:
0 Outlined (no fill).
1 Solid fill.
2 Horizontal hatch.
3 Vertical hatch.
4 Crosshatched.
5 Diagonal hatch.
6 Diagonal cross hatching.

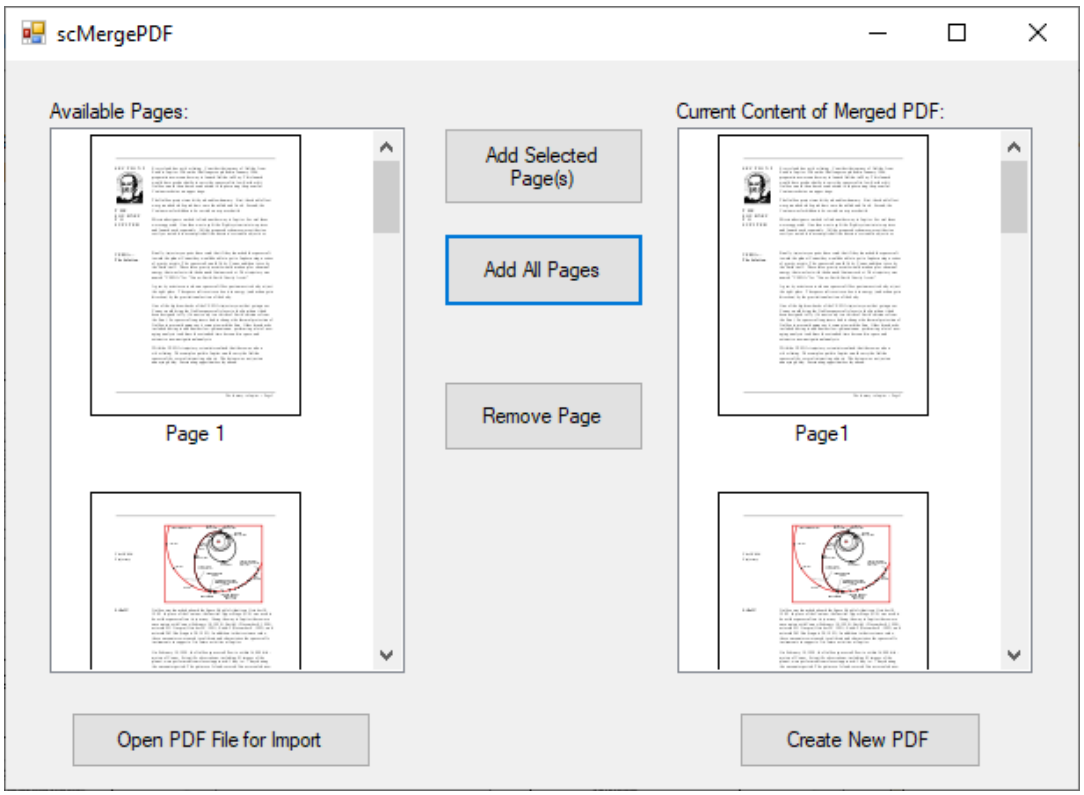
3.7 Appendix G SDK Examples

C# and C++ Source Code Examples

The SDK includes example applications, written in C# and C++ (MFC).
All these examples are installed to the following folder named:
"C:\Users\Public\Documents\scConverter SDK\examples"
Full source code and corresponding project files can be found as subfolders to this folder. You can also find more sample code in our support forum:
https://forums.softcomps.com

scMergePDF - A C# PDF Merge Example

This C# example demonstrates how to create a new PDF by merging different PDF files. Using the scConverter merge functionality you may include all pages, or individual pages, from the source PDF files.
Below you will find a screenshot of the PDF merge example application:

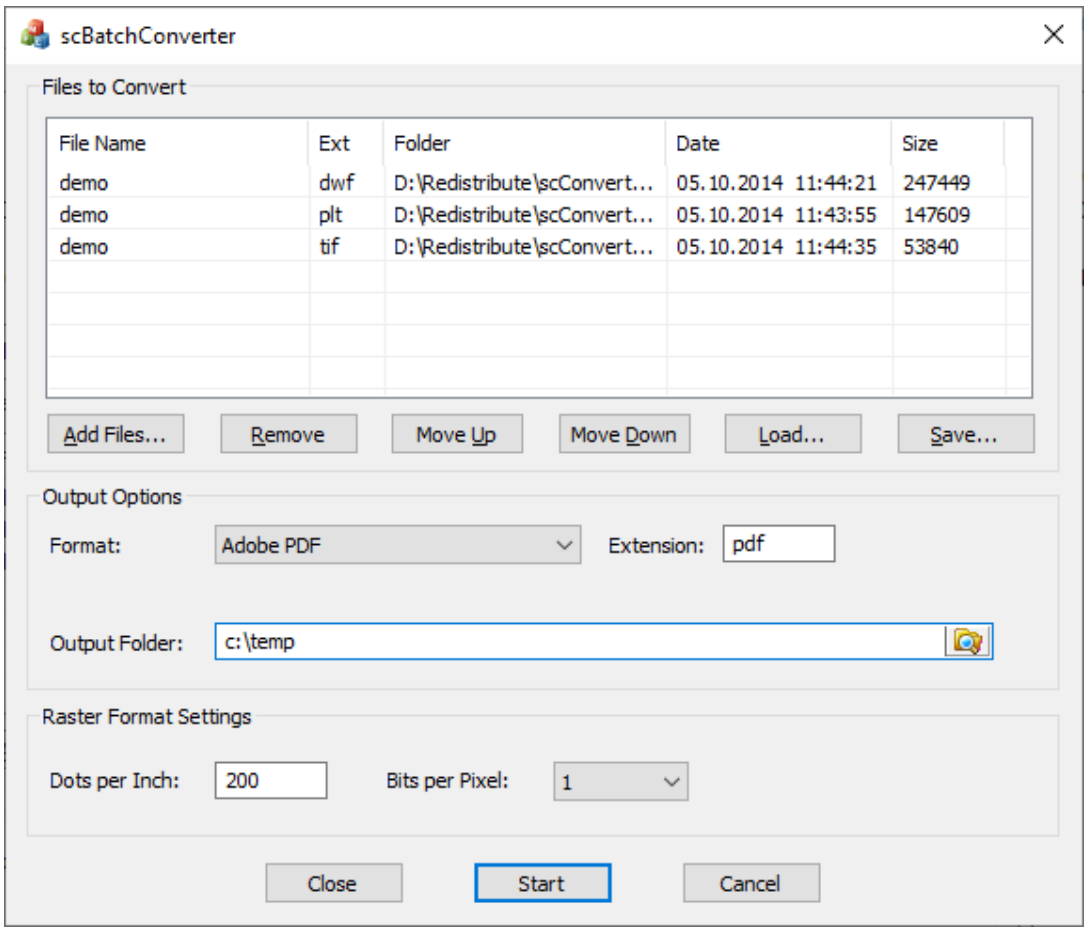


This application is very easy to use; you open different PDF files one by one using the **Open PDF File for Import** button. After the file is opened a thumbnail of each page in the file will be listed in the left page list. You may then either add all pages using the **Add all Pages** button or select one or more of the pages in the left page list and then add these using the **Add Selected Page(s)** button. When

all desired files and pages have been added, you may create the new merged PDF file using the **Create New PDF** button.

scBatchConverter - A C++ Batch Converter Example

The MFC/C++ sample application is a fully working Batch Converter utility using the scConverter to do the actual conversion.
With this application you can select files to a batch list, select output format and destination folder and finally batch convert all the files in the list.
Here is a screenshot of the batch converter example application with some of the sample files that are included in the SDK:



scConvSharp - A C# command line converter

This C# sample application is a very simple command line converter that accepts three arguments: Input filename, output filename and a format identifier. The supported format identifiers are the same as those accepted by the Convert method.

scConform - A C# PDF/A commmand line converter

This C# example is a command line converter that accepts two arguments: Input filename and output filename. This example will convert, or conform, the input PDF file to a PDF/A-3b compliant output file.

scExtractQR - A C# QR decode command line application

This example is a command line application that will find all QR codes in the input file and display the decoded text for each one found in the console.

scMarkupTester - A C# Markup XML Example

This C# application accepts one input file and one output file. When you press the **Convert with Markup** button the output file will be created with several different annotation types added to it. The annotations are added by using the Software Companions markup XML specification, see appendix F.

Sample Files

A set of different sample files can be found in the **"C:\Users\Public\Documents\scConverter SDK\samples"** folder. You may use these files to test the above example, or your own, applications.

3.8 Appendix H Tesseract OCR Engine

Tesseract OCR Engine Information

Tesseract OCR (Optical Character Recognition) engine is a free open-source software developed by Google. It is designed to recognize and extract text from images or scanned documents. Tesseract was initially developed at Hewlett-Packard Laboratories in the 1980s and later released as open source in 2005. Google took over the project in 2006 and has been actively maintaining and improving it since then.

Tesseract has undergone significant improvements over the years and is known for its high accuracy in recognizing printed text. However, its performance may vary depending on the quality of the input image, the complexity of the text, and the language being used.

The Tesseract OCR engine is a powerful tool for extracting text from images and documents. It has gained popularity due to its accuracy, language support, and active development community. However, it's important to note that while Tesseract performs well on printed text, it may not be as effective in recognizing handwriting or text with complex layouts.

Tesseract is an open-source project, which means that its source code is freely available for anyone to use, modify, and distribute. The project has a dedicated community of developers who actively contribute to its development, bug fixes, and feature enhancements.

You may download the latest available Tesseract installation from the following site:

https://github.com/UB-Mannheim/tesseract/wiki