

Conversion SDK v11

32-bit and 64-bit

API Reference

PDF SDK for Developers

Copyright © 1995-2016 Visual Integrity LLC/Square One bv. All rights reserved.

NOTICE: All information contained herein is the property of Square One bv. No part of this publication (whether in hardcopy or electronic form) may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Square One bv.

The Graphics Connection is a trademark of Square One bv. PostScript and PDF is a registered trademark of Adobe Systems Incorporated. All other trademarks are the property of their respective owners.

This publication and the information herein is furnished AS IS, is subject to change without notice, and should not be construed as a commitment by Square One bv. Square One bv assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied, or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes, and non-infringement of third party rights.

1. Getting Started

Thank you for purchasing the 32-bit and 64-bit Conversion SDK Developers Kit. With it, you'll have the power and flexibility you need to create/edit PDF files.

To get started, you should have:

Conversion SDK Developers Kit (either downloaded via FTP or on CD ROM)
License Agreement
Conversion SDK API PDF Reference Manual
License Key

It's a good idea to review the Conversion SDK API PDF Reference manual thoroughly before starting. You should be familiar with all appropriate functions and parameters before beginning to use this Developers Kit.

Support is at hand

If you have questions or need help using The Conversion SDK Developers Kit, please contact us during Central European business hours (8:30am – 6:00pm). Our contact information is:

Square One bv
Oosteinde 34
2361HE Warmond
The Netherlands



+31 71 362 7297



+31 71 890 0567



support@visual-integrity.com



www.visual-integrity.com

2. The Conversion SDK API PDF functions

In this section, you will find a description of all the functions of the API (Application Programming Interface) of the Conversion SDK PDF. The file **vgpsflow.h** is included in the developers kit.

Conversion SDK also contains demo source programs which demonstrates most of the API functions. See the `vcexdlg.cpp` file in **Visual Studio 2010 C++ PDF engine Example** directory.

Vg2Pdf_CreateFile

```
Vg2Pdf_CreateFile(void **pdf, char *filename, int pdfversion)
```

Description *Vg2Pdf_CreateFile* creates a PDF-file. Using the *pdf* pointer the contents of the PDF-file can be “rendered” using the API functions below.

Parameters

<i>pdf</i>	Handle to the PDF data of the PDF file
<i>fileName</i>	The name of the PDF file
<i>pdfversion</i>	Options are PDF15, PDFA1A and PDFA1B

Return Value Zero if PDF file is created.

Remarks Version number of the PDF-file is 1.5

Vg2Pdf_CloseFile

```
Vg2Pdf_CloseFile(void *pdf);
```

Description *Vg2Pdf_CloseFile* saves and closes the PDF file.

Parameters *pdf* Pointer to the PDF data

Return Value Always zero.

Remarks

Vg2Pdf_CreateLayer

Vg2Pdf_CreateLayer(void **layer, void *pdf, char *layerName, int onoff)

Description Vg2PDF_CreateLayer creates a new PDF layer with layerName. Visibility of the new layer can be set on or off.

Parameters	<i>layer</i>	Handle to the PDF layer data
	<i>pdf</i>	Pointer to the PDF data
	<i>layerName</i>	The name of the layer
	<i>onoff</i>	Set visibility of layer on or off

Return Value Returns zero if layer has been created

Remarks All layers should be created right after the Vg2Pdf_CreateFile call.

Vg2Pdf_BeginPage

Vg2Pdf_BeginPage(void *pdf, double w, double h)

Description Vg2Pdf_BeginPage creates a new PDF page.

Parameters	<i>pdf</i>	Pointer to the PDF data
	<i>w, h</i>	Width and height of page size in PDF units (1/72 inch)

Return Value Always zero.

Remarks

Vg2Pdf_EndPage

Vg2Pdf_EndPage(void *pdf)

Description Vg2Pdf_EndPage closes the PDF page.

Parameters	<i>pdf</i>	Pointer to the PDF data
-------------------	------------	-------------------------

Return Value Always zero.

Remarks

Vg2Pdf_OpenLayer

Vg2Pdf_OpenLayer(void *pdf, void *layerID)

Description Vg2Pdf_OpenLayer makes the layer with `layerId` the current layer.

Parameters

<i>pdf</i>	Pointer to the PDF data
<i>layerId</i>	Pointer of the layer (See Vg2Pdf_CreateLayer)

Return Value Always zero.

Remarks

Vg2Pdf_CloseLayer

Vg2Pdf_CloseLayer(void *pdf, void *layerID)

Description Vg2Pdf_CloseLayer closes the layer.

Parameters

<i>pdf</i>	Pointer to the PDF data
<i>layerId</i>	Pointer of the layer (See Vg2Pdf_CreateLayer)

Return Value Always zero.

Remarks

Below the API functions to add (=render) objects (=contents) to a PDF page. The current page is the page created with Vg2Pdf_BeginPage. The origin (0,0) of a PDF page is in the lower left corner. Units are in PDF units (1/72 inch).

API functions to set stroke, fill or text color.

Vg2Pdf_SetStrokeColor

```
Vg2Pdf_SetStrokeColor(void *pdf, unsigned char r, g, b)
```

Description	Vg2Pdf_SetStrokeColor sets the stroke (=outline) color of the new vector object to the specified RGB color.	
Parameters	<i>pdf</i>	Pointer to the PDF data
	<i>r, g, b</i>	R, G and B value (0 – 255)
Return Value	Always zero.	
Remarks		

Vg2Pdf_SetFillColor

```
Vg2Pdf_SetFillColor(void *pdf, unsigned char r, g, b)
```

Description	Vg2Pdf_SetFillColor sets the fill color of a new vector object to the specified RGB color.	
Parameters	<i>pdf</i>	Pointer to the PDF data
	<i>r, g, b</i>	R, G and B value (0 – 255)
Return Value	Always zero.	
Remarks		

The functions Vg2Pdf_NoStroke and Vg2Pdf_NoFill sets the stroke or fill to transparent.

Vg2Pdf_SetTextColor

```
Vg2Pdf_SetTextColor(void *pdf, unsigned char r, g, b)
```

Description	Vg2Pdf_SetTextColor sets the fill color of a new text object to the specified RGB color.	
Parameters	<i>pdf</i>	Pointer to the PDF data
	<i>r, g, b</i>	R, G and B value (0 – 255)
Return Value	Always zero.	
Remarks		

API functions to set the line width, style and cap, join options.

Vg2Pdf_SetDash

Vg2Pdf_SetDash (void *pdf, int dashId)

Description Vg2Pdf_SetDashId sets line style of the stroke (=outline) of a new vector object.

Parameters *pdf* Pointer to the PDF data
dashId See defines in vgpsflow.h

Return Value Always zero.

Vg2Pdf_SetLineCap

Vg2Pdf_SetLineCap (void *pdf, int capId)

Description Vg2Pdf_SetLineCap sets line end of the stroke (=outline) of a new vector object.

Parameters *pdf* Pointer to the PDF data
capId BUTT = 0; ROUND = 1; SQUARE = 2

Return Value Always zero.

Vg2Pdf_SetLineJoin

Vg2Pdf_SetLineJoin (void *pdf, int joinId)

Description Vg2Pdf_SetLineJoin sets line join of the stroke (=outline) of a new vector object.

Parameters *pdf* Pointer to the PDF data
joinId ROUND = 0; BEVEL= 1; MITER = 2

Return Value Always zero.

Vg2Pdf_SetLineWidth

Vg2Pdf_SetLineWidth (void *pdf, double lw)

Description Vg2Pdf_SetLineWidth sets line width in mm of the stroke (=outline) of a new vector object.

Parameters *pdf* Pointer to the PDF data
lw line width in mm

Return Value Always zero.

API functions to render (=add) objects. The objects are rendered using the current color settings.

Vg2Pdf_DrawLine

```
Vg2Pdf_DrawLine(void *pdf, CuPoint *pt1, *pt2)
```

Description Vg2Pdf_DrawLine renders (=add) a line to the PDF page.

Parameters

<i>pdf</i>	Pointer to the PDF data
<i>pt1, pt2</i>	Pointers to x,y coordinates of begin and end point in PDF units (1/72 inch)

Return Value Always zero.

Vg2Pdf_DrawPoly

```
Vg2Pdf_DrawPoly(void *pdf, CU_IRPOLY_PARAM *cupoly, int close)
```

Description Vg2Pdf_DrawPoly renders (=add) a polyline or polygon to the PDF page.

Parameters

<i>pdf</i>	Pointer to the PDF data
<i>cupoly</i>	Array of points
<i>close</i>	If close poly will be closed (= polygon)

Return Value Always zero.

Vg2Pdf_DrawRect

```
Vg2Pdf_DrawRect(void *pdf, CuPoint *pt1, *pt2)
```

Description Vg2Pdf_DrawRect renders (=add) a rectangle to the P the PDF page.

Parameters

<i>pdf</i>	Pointer to the PDF data
<i>pt1, pt2</i>	Pointers to x,y coordinates of lower left and upper right

Return Value Always zero.

Vg2Pdf_DrawCircle

```
Vg2Pdf_DrawCircle(void *pdf, CU_CIRCLE_PARAM *cucircle)
```

Description Vg2Pdf_DrawCircle renders (=add) a circle to the P the PDF page.

Parameters

<i>pdf</i>	Pointer to the PDF data
<i>cucircle</i>	pointer to circle data, middle point and radius

Return Value Always zero.

Vg2Pdf_DrawEllipse

Vg2Pdf_DrawEllipse(void *pdf, CU_ELLIPSE_PARAM *cuellipse)

Description	Vg2Pdf_DrawEllipse renders (=add) an ellipse to the PDF page.	
Parameters	<i>pdf</i>	Pointer to the PDF data
	<i>cuellipse</i>	Pointer to ellipse data, middle point, a and b axis
Return Value	Always zero.	

Vg2Pdf_DrawPath

Vg2Pdf_DrawPath(void *pdf, CuPathDef *p)

Description	Vg2Pdf_DrawPath renders (=add) a set of Bezier paths to the PDF page.	
Parameters	<i>pdf</i>	Pointer to the PDF data
	<i>p</i>	Pointer to the Bezier paths (see custom.h)
Return Value	Always zero.	

Vg2Pdf_DrawJpgImage

Vg2Pdf_DrawJpgImage(void *pdf, char *jpgName, double *pts)

Description	Vg2Pdf_DrawJpgImage renders (=add) a JPEG file as image object to the PDF page.	
Parameters	<i>pdf</i>	Pointer to the PDF data
	<i>jpgName</i>	Name of the JPEG file
	<i>pts</i>	6 points of the parallelogram of the image object
Return Value	Returns zero if succeeds.	

Vg2Pdf_DrawImage

Vg2Pdf_DrawImage(void *pdf, int depth, w, h, char *imdata, int imlength, double *pts)

Description	Vg2Pdf_DrawImage renders (=add) an image object to the PDF page.	
Parameters	<i>pdf</i>	Pointer to the PDF data
	<i>depth</i>	Color Depth of image
	<i>w, h</i>	width and height of image in pixels
	<i>imdata</i>	image data (= pixel values)
	<i>imlength</i>	length in bytes of image data (no padding)
Return Value	Returns zero if succeeds.	

API functions to add links or tooltips to a PDF page.

Vg2Pdf_CreateLink

```
Vg2Pdf_CreateLink(void *pdf, int onpage, double *bbox, double bwidth, int bstyle, double yloc)
```

Description Vg2Pdf_CreateLink adds a link from a page to another page in the PDF file

Parameters

<i>pdf</i>	Pointer to the PDF data
<i>onpage</i>	from page onpage
<i>bbox</i>	bounding box of the clickable link
<i>bwidth</i>	line width of the bounding box
<i>bstyle</i>	style of the bounding box
<i>topage</i>	page to go to when clicked
<i>yloc</i>	y value of the <i>topage</i> to show in the top area

Return Value Returns zero if succeeds.

Vg2Pdf_CreateURLLink

```
Vg2Pdf_CreateURLLink(void *pdf, int onpage, double *bbox, double bwidth, int bstyle, char *url)
```

Description Vg2Pdf_CreateURLLink adds a link from a page to a web-site (=url)

Parameters

<i>pdf</i>	Pointer to the PDF data
<i>onpage</i>	from page onpage
<i>bbox</i>	bounding box of the clickable link
<i>bwidth</i>	line width of the bounding box
<i>bstyle</i>	style of the bounding box
<i>url</i>	url to a web-site

Return Value Returns zero if succeeds.

Vg2Pdf_CreateToolTip

```
Vg2Pdf_CreateToolTip(void *pdf, int onpage, double *bbox, char *dname, char *tooltip)
```

Description Vg2Pdf_CreateLink adds a link from with a page to another page.

Parameters

<i>pdf</i>	Pointer to the PDF data
<i>onpage</i>	from page onpage
<i>bbox</i>	bounding box of the tooltip area
<i>dname</i>	name (= title) of tooltip
<i>tooltip</i>	string (= contents) of tooltip

Return Value Returns zero if succeeds.

3. Additional Functionality

3a. API functions to add objects (= contents) on a page of an existing PDF-file

Vg2Pdf_Append2Page

```
Vg2Pdf_Append2Page(void *pdf, int pageno)
```

Description	Vg2Pdf_Append2Page sets the page with <i>pageno</i> as the current page in the PDF-file.	
Parameters	<i>pdf</i>	Pointer to the PDF data
	<i>pageno</i>	Page number to which append objects (First page is 1)
Return Value	Returns zero if page with <i>pageno</i> exist.	
Remarks	Call Vg2Pdf_EndPage when done.	

3b. API functions to add a page to an existing PDF-file

Vg2Pdf_InsertAndBeginPage

```
Vg2Pdf_InsertAndBeginPage(void *pdf, int where, double w, double h)
```

Description	Vg2Pdf_InsertAndBeginPage creates a new page an inserts it before the page <i>where</i> . The new page becomes the current page to which objects can be added	
Parameters	<i>pdf</i>	Pointer to the PDF data
	<i>where</i>	New page will be added before <i>where</i> (First page is 1)
	<i>w, h</i>	Width and height of page size in PDF units (1/72 inch)
Return Value	Returns zero if page can be added.	
Remarks	Call Vg2Pdf_EndPage when done.	

3c. API function to convert a WMF file to PDF

VgConvertWmf2Pdf

```
VgConvertWmf2Pdf(char *in, char *out)
```

Description VgConvertWmf2Pdf converts the WMF-file *in* to PDF-file *out*.

Parameters *in* Name of WMF-file
 out Name of PDF-file

Return Value Returns zero if conversion succeeds.

Remarks

3d. API function to convert a EMF file to PDF

VgConvertEmf2Pdf

```
VgConvertEmf2Pdf(char *in, char *out)
```

Description VgConvertEmf2Pdf converts the EMF-file *in* to PDF-file *out*.

Parameters *in* Name of EMF-file
 out Name of PDF-file

Return Value Returns zero if conversion succeeds.

Remarks

3d. API function to merge 2 PDF-files

Vg2Pdf_Append2Pdf

```
Vg2Pdf_Append2Pdf(char *inFile, char *appendFile)
```

Description Vg2Pdf_Append2Pdf merges 2 PDF files.

Parameters *inFile* Name of PDF-file to which the other PDF file will be appended
 appendFile Name of PDF-file which will be appended to the *inFile*

Return Value Returns zero if conversion succeeds.

Remarks

3e. API function to extract the object calls from a page

VgPdfExtractPageData

VgPdfExtractPageData(char *inFile, int pageno, char *outFile)

Description VgPdfExtractPageData extracts the object data (= contents) of the page pageno of the PDF-file inFile and saves it to the OutFile.

Parameters

<i>inFile</i>	Name of PDF-file
<i>pageno</i>	Page number
<i>outFile</i>	Name of the file to store the contents of the page

Return Value Returns zero if extracting of the contents succeeds.

4. Visual Studio 2010 C++ Example

In sub-directories *Visual Studio 2010 C++ PDF engine Example* and *Visual Studio 2010 C++ Meta2PDF Example* you will find demo source code which demonstrates how to use the Conversion SDK API PDF .The demo program is created using Visual Studio 2010.

You can use these projects as a start for your own application or just copy/paste the code you want to reuse.

The function calls to the Conversion SDK PDF kit are very simple ! Just have a look at the **vcexdlg.cpp** source file.

After compiling the example the executable **vcpdfexample.exe/vcmeta2pdfexample** is copied to the bin/Win32 or bin/x64 sub directory. You can run it from within this sub directory.

4. Running your own program

In order to run your program using the Conversion SDK libraries you need to add the following files/directories into the same directory as your program:

- Vgflow.dll
- Vgpsflow.dll
- Fonts directory