

SII_PDF Interface

UnlockDLL(long IUnlock)	3
OpenPDF(filename, ignore protect, user pass, owner pass, pdf handle)	4
ConvertPDFToTextFile(first page, last page, pdf handle, output file)	5
ConvertPDFToText(first page, last page, pdf handle, buffer, size)	6
ConvertPDFToHTMLFile(first page, last page, pdf handle, output file)	7
ConvertPDFToHTML(first page, last page, pdf handle, buffer, size)	8
ClosePDF(pdf handle)	9
GetNumPages(pdf handle)	10
SetTextEncoding(char * encodingName)	10
SetErrQuiet(BOOL errQuiet)	10
Text output settings	12
SetwordMinSpaceWidth (double width)	12
GetwordMinSpaceWidth () 0.15	12
SetwordMaxSpaceWidth (double width)	12
GetwordMaxSpaceWidth () 2.0	12
SetwordDefMinSpaceWidth (double width)	12
GetwordDefMinSpaceWidth () .09	12
SetwordDefMaxSpaceWidth (double width)	12
GetwordDefMaxSpaceWidth () 1.5	12
SetupMaxDeltaX(double deltaX)	13
GetdupMaxDeltaX() .4	13
SetupMaxDeltaY(double deltaY)	13
GetdupMaxDeltaY() .4	13
SetlineOverlapSlack(double slack)	13
GetlineOverlapSlack() .3	13
SetlineMaxBaselineDelta(double delta)	13
GetlineMaxBaselineDelta() .1	13
SetlineMaxFontSizeRatio(double maxsize)	13
GetlineMaxFontSizeRatio() 1.4	13
SetlineMinDeltaX (double deltaX)	14
GetlineMinDeltaX () -0.5	14
SetlineMinSuperscriptOverlap(double MinSuper)	14
GetlineMinSuperscriptOverlap() 0.3	14
SetlineMinSubscriptOverlap(double MinSub)	14
GetlineMinSubscriptOverlap() 0.3	14
SetlineMinSubscriptFontSizeRatio (double size)	14
GetlineMinSubscriptFontSizeRatio () 0.4	14
SetlineMaxSubscriptFontSizeRatio (double size)	14
GetlineMaxSubscriptFontSizeRatio () 1.01	14
SetlineMinSuperscriptFontSizeRatio (double size)	14
GetlineMinSuperscriptFontSizeRatio () 0.4	14
SetlineMaxSuperscriptFontSizeRatio (double size)	14
GetlineMaxSuperscriptFontSizeRatio () 1.01	14
SetlineMaxSubscriptDeltaX (double maxsub)	15
GetlineMaxSubscriptDeltaX () 0.2	15
SetlineMaxSuperscriptDeltaX (double maxsuper)	15

SII_PDF Interface

GetlineMaxSuperscriptDeltaX () 0.2.....	15
SetblkMaxSpacing (double maxspace).....	15
GetblkMaxSpacing () 2.0.....	15
SetblkMaxFontSizeRatio(double maxsize).....	15
GetblkMaxFontSizeRatio() 1.3.....	15
SetblkOverlapSlack (double overlap).....	15
GetblkOverlapSlack () 0.5.....	15
SetflowMaxDeltaY(double maxvert).....	16
GetflowMaxDeltaY() 1.0.....	16
Appendix A – Output Formatting Issues.....	17
Unicode without embedded map.....	17
Font without embedded fonts.....	17
Graphics.....	17
Lines, curves and other drawing tools.....	17
Order of objects on the page.....	17
Color of text on page.....	17
Superscript/Subscript.....	17
Misshapen text boxes.....	17
Bold text.....	18
Fonts and positioning.....	18
Tiny fonts.....	18
Annotations.....	18
Graphic “paths”.....	18
Links.....	18

SII_PDF Interface

UnlockDLL(long IUnlock)

Description:

Unlocks the PDF DLL library of functions for use..

Input Fields:

IUnlock *long* *Code used to unlock the dll*

Return codes:

1 – DLL Unlocked

0 – DLL NOT unlocked (invalid unlock code sent)

Usage:

UnlockDLL(871464) ;

SII_PDF Interface

OpenPDF(filename, ignore protect, user pass, owner pass, pdf handle)

Description:

Opens the PDF file for reading and loads the catalog of the pdf for information on the document (such as number of pages).

Input Fields:

<i>Filename</i>	<i>Char *</i>	<i>Contains the full path and filename for the input PDF file.</i>
<i>Ignore Protect</i>	<i>BOOL</i>	<i>TRUE – Ignores password protection on the PDF FALSE – Passwords will be used (if needed)</i>
<i>User password</i>	<i>Char *</i>	<i>Password for the person USING the pdf</i>
<i>Owner password</i>	<i>Char *</i>	<i>Password for the creator of the pdf</i>
<i>Pdf Handle</i>	<i>long *</i>	<i>Handle for the open PDF</i>

Return codes:

LONG

Status returned:

- 0 – success*
- 1 - couldn't open the PDF file*
- 2 - couldn't read the page catalog*
- 3 - PDF file is damaged*
- 4 - file is encrypted and password was incorrect or not supplied*
- 5 – User does not have permission to read*

Usage:

```
long pdf;
```

```
long status;
```

```
status = OpenPDF("input.pdf", TRUE, NULL, NULL, &pdf );
```

SII_PDF Interface

ConvertPDFToTextFile(first page, last page, pdf handle, output file)

Description:

Reads in the open PDF file from the first page designated to the last page designated, outputting each page to an ASCII text format to the passed filename, retaining the basic layout of the text.

Input Fields:

<i>First page</i>	<i>long</i>	<i>First page to begin converting. If the value is less than 1 then it is defaulted to the first page of the PDF.</i>
<i>Last page</i>	<i>long</i>	<i>Last page to convert. If the value is less than 1 then it is defaulted to the last page of the PDF</i>
<i>Pdf Handle</i>	<i>long *</i>	<i>Handle of the open PDF (obtained by calling OpenPDF)</i>
<i>Output file</i>	<i>char *</i>	<i>Full path and filename for the converted output file.</i>

Return codes:

1(TRUE) – file converted successfully. Zero (FALSE) – file conversion had problems.

Usage:

```
long pdf;
```

```
long status;
```

```
status = OpenPDF( "input.pdf", TRUE, NULL, NULL, &pdf );
```

```
status = ConvertPDFToTextFile( 0, 0, &pdf, "outfile.txt" );
```

SII_PDF Interface

ConvertPDFToText(first page, last page, pdf handle, buffer, size)

Description:

Reads in the open PDF file from the first page designated to the last page designated, outputting each page to an ASCII text format to the passed buffer in memory..

Input Fields:

<i>First page</i>	<i>long</i>	<i>First page to begin converting. If the value is less than 1 then it is defaulted to the first page of the PDF.</i>
<i>Last page</i>	<i>long</i>	<i>Last page to convert. If the value is less than 1 then it is defaulted to the last page of the PDF</i>
<i>Pdf Handle</i>	<i>long *</i>	<i>Handle of the open PDF (obtained by calling OpenPDF)</i>
<i>Buffer</i>	<i>char *</i>	<i>Address of the buffer to receive the converted data.</i>
<i>Size</i>	<i>int</i>	<i>size of the buffer to receive the converted data</i>

Return codes:

1(TRUE) – file converted successfully. Zero (FALSE) – file conversion had problems.

Usage:

```
long pdf;  
long status;  
char buffer[1000];  
  
status = OpenPDF( "input.pdf", TRUE, NULL, NULL, &pdf );  
status = ConvertPDFToText( 0, 0, &pdf, buffer, sizeof(buffer) );
```

SII_PDF Interface

ConvertPDFToHTMLFile(first page, last page, pdf handle, output file)

Description:

Reads in the open PDF file from the first page designated to the last page designated, outputting each page to an ASCII text format to the passed filename, retaining the basic layout of the text. In addition, simple HTML headers and footers are appended to the file to build an HTML layout.

Input Fields:

<i>First page</i>	<i>long</i>	<i>First page to begin converting. If the value is less than 1 then it is defaulted to the first page of the PDF.</i>
<i>Last page</i>	<i>long</i>	<i>Last page to convert. If the value is less than 1 then it is defaulted to the last page of the PDF</i>
<i>Pdf Handle</i>	<i>long *</i>	<i>Handle of the open PDF (obtained by calling OpenPDF)</i>
<i>Output file</i>	<i>char *</i>	<i>Full path and filename for the converted output file.</i>

Return codes:

1(TRUE) – file converted successfully. Zero (FALSE) – file conversion had problems.

Usage:

```
long pdf;  
long status;
```

```
status = OpenPDF( "input.pdf", TRUE, NULL, NULL, &pdf );  
status = ConvertPDFToHTMLFile( 0, 0, &pdf, "outfile.html" );
```

SII_PDF Interface

ConvertPDFToHTML(first page, last page, pdf handle, buffer, size)

Description:

Reads in the open PDF file from the first page designated to the last page designated, outputting each page to an ASCII text format to the passed buffer in memory. In addition, simple HTML headers and footers are added to the data to produce an HTML layout.

Input Fields:

<i>First page</i>	<i>long</i>	<i>First page to begin converting. If the value is less than 1 then it is defaulted to the first page of the PDF.</i>
<i>Last page</i>	<i>long</i>	<i>Last page to convert. If the value is less than 1 then it is defaulted to the last page of the PDF</i>
<i>Pdf Handle</i>	<i>long *</i>	<i>Handle of the open PDF (obtained by calling OpenPDF)</i>
<i>Buffer</i>	<i>char *</i>	<i>Address of the buffer to receive the converted data.</i>
<i>Size</i>	<i>int</i>	<i>size of the buffer to receive the converted data</i>

Return codes:

0 – file converted successfully. Non-zero – file conversion had problems.

Usage:

```
long pdf;
```

```
long status;
```

```
char buffer[1000];
```

```
status = OpenPDF( "input.pdf", TRUE, NULL, NULL, &pdf );
```

```
status = ConvertPDFToHTML( 0, 0, &pdf, buffer, sizeof(buffer) );
```

SII_PDF Interface

ClosePDF(pdf handle)

Description:

Closes the open PDF handle and releases all memory associated with the open PDF.

Input Fields:

*Pdf Handle long * Handle of the open PDF (obtained by calling OpenPDF)*

Return codes:

0 – handle closed Non-zero – closing had issues

Usage:

```
long *pdf;
```

```
long status;
```

```
char buffer[1000];
```

```
status = OpenPDF( "input.pdf", TRUE, NULL, NULL, &pdf );
```

```
status = ConvertPDFToText( 0, 0, &pdf, buffer, sizeof(buffer) );
```

```
ClosePDF ( &pdf ) ;
```


SII_PDF Interface

SII_PDF Interface

Text output settings

The text output settings all return VOID for the set and DOUBLE for the Get.

The set function takes a double as the only parameter and sets the variable within the DLL equal to the passed parameter.

The get function returns to the caller the current setting of the variable.

These functions are used to adjust the text output for special situations. (see appendix A)

SetwordMinSpaceWidth (double width)

GetwordMinSpaceWidth () 0.15

SetwordMaxSpaceWidth (double width)

GetwordMaxSpaceWidth () 2.0

Description:

Minimum and maximum inter-word spacing (as a fraction of the average character width).

This is the smallest and largest space between 2 letters to make it the same word.

If two words are split and should NOT be, adjust the max.

If two words are not split and they SHOULD be – adjust the min.

SetwordDefMinSpaceWidth (double width)

GetwordDefMinSpaceWidth () .09

SetwordDefMaxSpaceWidth (double width)

GetwordDefMaxSpaceWidth () 1.5

Description:

Default min and max inter-word spacing (when the average character width is unknown – if there isn't a font def for the font).

These are the default spacing to use if NO EMBEDDED FONT exists in the PDF file. Normally this is not used (most PDF files have embedded font details)

SII_PDF Interface

SetupMaxDeltaX(double deltaX)

GetdupMaxDeltaX() .4

SetupMaxDeltaY(double deltaY)

GetdupMaxDeltaY() .4

Description:

Max difference in x,y coordinates (as a fraction of the font size) allowed for duplicated text (fake boldface, drop shadows) which is to be discarded.

If duplicate text is in the output as a result of “fake” boldface, try decreasing these numbers until the duplicate text is removed. If VALID duplicates are being erroneously removed, increase the values here to restore the text to the output.

SetlineOverlapSlack(double slack)

GetlineOverlapSlack() .3

Description:

Min overlap (as a fraction of the font size) required for two lines to be considered vertically overlapping.

If 2 lines overlap on the PDF , they must be either merged onto the SAME line in the output OR moved to separate lines. Increase this number if output is desired to be on the SAME line, decrease this value to make sure output is on separate lines.

SetlineMaxBaselineDelta(double delta)

GetlineMaxBaselineDelta() .1

Description:

Max difference in baseline y coordinates (as a fraction of the font size) allowed for words which are to be grouped into a line, not including sub/superscripts.

Similar to the overlap, this is used to group a “sentence” together all on the same line

SetlineMaxFontSizeRatio(double maxsize)

GetlineMaxFontSizeRatio() 1.4

Description:

Max ratio of font sizes allowed for words which are to be grouped into a line, not including sub/superscripts.

If multiple fonts are used within the SAME line, we either need to keep all text on the same output line, or potentially move the different font portion to its own line, depending on the size. Increase this number for text output to be on same line, decrease it to force different fonts to be on separate lines.

SII_PDF Interface

SetlineMinDeltaX (double deltaX)

GetlineMinDeltaX () -0.5

Description:

Min spacing (as a fraction of the font size) allowed between words which are to be grouped into a line.

SetlineMinSuperscriptOverlap(double MinSuper)

GetlineMinSuperscriptOverlap() 0.3

SetlineMinSubscriptOverlap(double MinSub)

GetlineMinSubscriptOverlap() 0.3

Description:

Minimum vertical overlap (as a fraction of the font size) required for superscript and subscript words.

SetlineMinSubscriptFontSizeRatio (double size)

GetlineMinSubscriptFontSizeRatio () 0.4

SetlineMaxSubscriptFontSizeRatio (double size)

GetlineMaxSubscriptFontSizeRatio () 1.01

SetlineMinSuperscriptFontSizeRatio (double size)

GetlineMinSuperscriptFontSizeRatio () 0.4

SetlineMaxSuperscriptFontSizeRatio (double size)

GetlineMaxSuperscriptFontSizeRatio () 1.01

Description:

Min/max ratio of font sizes allowed for sub/superscripts compared to the base text.

If superscript and sub script are used within the PDF and are based on FONT SIZES within the same text line(x,y), changing the min and max will adjust whether or not the text is on the same line or a new line as the existing text it is super or subbing.

SII_PDF Interface

SetlineMaxSubscriptDeltaX (double maxsub)

GetlineMaxSubscriptDeltaX () 0.2

SetlineMaxSuperscriptDeltaX (double maxsuper)

GetlineMaxSuperscriptDeltaX () 0.2

Description:

Max horizontal spacing (as a fraction of the font size) allowed before sub/superscripts.

If super/sub scripts are present as the same font, but on a different row, adjusting these values will allow you to either merge the super/sub to the same row as the text, or make it a different row.

SetblkMaxSpacing (double maxspace)

GetblkMaxSpacing () 2.0

Description:

Maximum vertical spacing (as a fraction of the font size) allowed for lines which are to be grouped into a block.

This allows lines to be grouped so columns can line up properly. If a “grid” is displayed for instance and each row is far apart (due to whitespace desired in the PDF or whatever) then you can increase this value to ensure that the columns will properly line up.

If the output is mistakenly assuming the lines are together, when in fact they are NOT and the discrepancy in the column is desired – then DECREASE this value to ensure the rows are NOT grouped together.

SetblkMaxFontSizeRatio(double maxsize)

GetblkMaxFontSizeRatio() 1.3

Description:

Max ratio of primary font sizes allowed for lines which are to be grouped into a block.

Similar to the blkMaxSpacing – this is for grouping different rows with different FONTS. This is so different fonts will still line up in the output file if they are part of the same “column”. Unless the font is DRASTICALLY different in size – which is where this value comes in.

SetblkOverlapSlack (double overlap)

GetblkOverlapSlack () 0.5

Description:

Min overlap (as a fraction of the font size) required for two blocks to be considered vertically overlapping.

SII_PDF Interface

SetflowMaxDeltaY(double maxvert)

GetflowMaxDeltaY() 1.0

Description:

Max vertical offset (as a fraction of the font size) of the top and bottom edges allowed for blocks which are to be grouped into a flow.

SII_PDF Interface

Appendix A – Output Formatting Issues

Unicode without embedded map

Unicode is supported as long as the map is included within the PDF. A default map is coded, so if Unicode map is NOT enclosed, it still MAY work, but could end up not matching exactly.

Font without embedded fonts

PDF creators usually embed the font type used within the pdf itself, but if a user explicitly selects to NOT embed the font, then issues can arise with the proper spacing. If a standard font is used, the DLL can recover and use the encodings provided within the code. If TRUETYPE fonts are used, there is no support for external font files, therefore a default will be used and may cause the spacing to be off.

Graphics

Graphics are obviously not supported within text extraction, but depending on how the text “interacts”, a graphic with text surrounding it MAY cause the text to not be in the proper position. In MOST cases, the text is drawn within its own “text box” and graphics will not affect the text position.

Lines, curves and other drawing tools

These may change the position of surrounding text and could cause text to be moved to an undesired location in the pure text output.

Order of objects on the page

Text boxes that are “hidden” behind graphics or other text boxes in the PDF will still be extracted and shown in the text output. There is no provision for the ORDER of text objects – ALL text objects will be processed.

Color of text on page

Color of text is ignored. If a text section is made the same color as the background to remove it instead of removing the text box itself – this text will still be in the output.

Superscript/Subscript

Superscript and subscript fonts may end up appearing either on the same line, or on a line by itself – depending on the size and the settings provided.

Misshapen text boxes

Text boxes that are distorted, backwards (pulled the wrong way when creating the PDF) etc – WILL be processed. In the majority of cases – this text is most likely “hidden” behind a graphic and forgotten – but

SII_PDF Interface

will show up in the ACSII processing. If “extra” characters seem to appear in the output that don’t show in the PDF viewer, then this most likely is the case.

Bold text

Some PDF creators duplicate text at a slightly off x,y coordinate – basically copying the text to be bold and moving the duplicate a little lower and a little to the right – resulting in the same text twice on the screen at different spacing – creating the bold effect. When we extract text – this can result in text being duplicated. Settings have been provided to adjust for this.

Fonts and positioning

Certain fonts mixed with other fonts COULD cause issues when trying to maintain the spacing/order of the original PDF. The best possible format is maintained.

Tiny fonts

Fonts within a PDF that are UNDER 3 points are ignored.

Annotations

Annotations are not currently supported and are ignored.

Graphic “paths”

Graphic paths (embedded graphics) are not supported and are ignored.

Links

Links are not followed and not processed, however the text itself (depending on how the link was created) should still be maintained in the output.