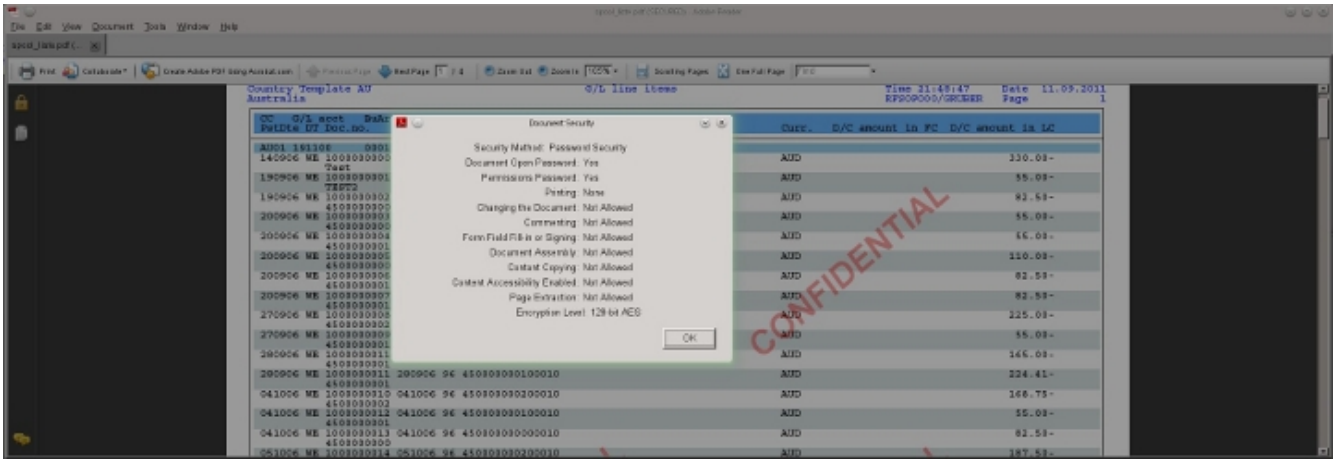

PDF | PDF Encryption

Library API Reference for Developers

Class /GRUBE/CL_PDF:

The class /GRUBE/CL_PDF is the main entry point for the PDF Encryption and Processing. It represents one PDF document.

Constructor:

Parameter:

DO_NOT_TOUCH_OBJSTREAMS (Import) – values 'X' or ' '

The parameter controls the behavior of the PDF disassembler. If it is set to 'X' object streams are not decoded, which results in significant improvement of memory consumption and performance. However if relevant PDF objects (like the root object) are hidden in object streams the disassemble process will not decode the object and further processing will fail. The default is ' ' and should work for most scenarios.

Example:
data: o_pdf type ref to /grube/cl_pdf.
create object o_pdf.

Method SET_PDF_FILE:

This method initializes the object with the PDF as xstring. Usually SAP handles PDF as internal tables. Some simple functions allow the conversion of internal tables into xstring (e.g. /grube/cl_pdf=>convert_xstring_2_table(xstring) or SCMS_BINARY_TO_XSTRING).

Parameter:

PDF (XSTRING – Import): The full PDF document as xstring.

Exceptions:

/GRUBE/CX_PDF_SYNTAX: Input PDF file has syntax errors which may result in an unusable output PDF file

Example:

```
call method cl_gui_frontend_services=>gui_upload
  exporting
    filename   = fl
    filetype   = 'BIN'
  importing
    filelength = len
  changing
    data_tab   = a.
```

```
* convert the PDF table into xstring
  loop at a into l_a.
    concatenate result l_a-line  into result in byte mode.
  endloop.
  data: o_pdf type ref to /grube/cl_pdf.
* create the PDF object
  create object o_pdf.
* load the PDF into the object (XString)
  o_pdf->set_pdf_file( result ).
```

Method CREATE_PDF:

This method creates the PDF and returns the actual state of the /GRUBE/CL_PDF object as a PDF document in an xstring format.

Parameter:

OUT (XSTRING – Returning): Returns the PDF as xstring.

Example:

```
result = o_pdf->create_pdf( ).
```

Method DO_ACTION:

This method executes the passed action (subclass of /GRUBE/CL_PDF_ACTION) for all objects in the PDF document. The encryption objects are also subclasses of /GRUBE/CL_PDF_ACTION and this function is used to encrypt the PDF.

Parameter:

IN (/GRUBE/CL_PDF_ACTION – Importing): The action, which should be executed for all objects in the PDF document..

Exceptions: /GRUBE/CX_PDF_UKNWN_STRCT_CHG Illegal Operation for Postprocessing after encryption or a similar action was executed

Example:

```
   data: o_encrypt_rc4 type ref to /grube/cl_pdf_encryption_rc4.
   create object o_encrypt_rc4 exporting keylength = 80 .

…
```

```
o_pdf->do_action( o_encrypt_rc4 ).
result = o_pdf->create_pdf( ).
```

Method ADD_WATERMARK_TILED:

This method adds multiple watermarks tiled over the whole background of the PDF. It expects a non indexed Windows Bitmap (BMP) and places the bitmap in the background of the PDF.

Parameter:

BMP (GRUBE/CL_PDF_GRAPHIC – Importing) the bitmap for the background (see reference for /GRUBE/CL_PDF_GRAPHIC and /GRUBE/CL_PDF_BITMAP)

SIZEX (Integer – Importing) horizontal size in mm (millimeters)

SIZEY (Integer – Importing) vertical size in mm (millimeters)

AS_BACKGROUND (CHAR1 – Importing) : Default ' ' - If this flag is set the bitmap is placed in the background and all other graphic will hide parts or all of the bitmap. If the flag is blank the bitmap is located on top of all other page objects and will hide parts or all of the underlying graphics, if the opacity is set to 255.

Example (Bitmap in the size 10 cm by 8 cm)

```
data: o_pdf type ref to /grube/cl_pdf.
create object o_pdf.
o_pdf->add_watermark_tiled( bmp = o_bitmap  sizex = 100 sizey = 80 as_background = ' ' ).
```

Method ADD_WATERMARK_POSITION:

This method adds a single watermark to background of the PDF. It expects a non indexed Windows Bitmap (BMP) and places the bitmap in the background of the PDF.

Parameter:

BMP (GRUBE/CL_PDF_GRAPHIC – Importing) the bitmap for the background (see reference for /GRUBE/CL_PDF_GRAPHIC and /GRUBE/CL_PDF_BITMAP)

X (Integer – Importing) horizontal position in mm (millimeters) from the lower left corner

Y (Integer – Importing) vertical position in mm (millimeters) from the lower left corner

SIZEX (Integer – Importing) horizontal size in mm (millimeters)

SIZEY (Integer – Importing) vertical size in mm (millimeters)

AS_BACKGROUND (CHAR1 – Importing) : Default ' ' - If this flag is set the bitmap is placed in the background and all other graphic will hide parts or all of the bitmap. If the flag is blank the bitmap is located on top of all other page objects and will hide parts or all of the underlying graphics, if the opacity is set to 255.

Example (single bitmap in the size 15 cm by 12 cm at 2 cm/6cm from the lower left corner)

```
data: o_pdf type ref to /grube/cl_pdf.
create object o_pdf.
o_pdf->add_watermark_position( bmp = o_bitmap  sizex = 150 sizey = 120 x = 20 y = 60
as_background = ' ' ).
```

Method ADD_TRAILER_PAGES:

This method adds trailer pages at the beginning or the end of the PDF document. The additional pages are provided as PDF document.

Parameter:

PDF_DOCUMENT (GRUBE/CL_PDF – Importing) : The PDF document with the additional pages

INSERT_AT_START: 'X' = Insert the pages at the beginning of the document. ' ' = Insert the pages at the end of the document.

Class /GRUBE/CL_PDF_ENCRYPTION_AES and /GRUBE/CL_PDF_ENCRYPTION_RC4 and /GRUBE/CL_PDF_ENCRYPTION and /GRUBE/CL_PDF_ENCRY_AES_256:

The class /GRUBE/CL_PDF_ENCRYPTION_AES handles all parts which are required for AES encryption. The class /GRUBE/CL_PDF_ENCRYPTION_RC4 handles all parts which are required for RC4 encryption. The class /GRUBE/CL_PDF_ENCRYPTION is the basis class for all encryption. The PDF encryption algorithm has dramatically changed with the introduction of AES 256 encryption. So the new object /grube/CL_PDF_ENCRY_AES_256 can be used instead of /grube/cl_pdf_encryption_aes or /grube/cl_pdf_encryption_rc4 to encrypt the PDF with AES-256.

Constructor:

Parameter:

KEYLENGTH (Integer – Import – Optional – Default 128)

The parameter sets the keylength to 128.

Please note: the constructor of /grube/CL_PDF_ENCRY_AES_256 sets the keylength to 256.

Example:
data: o_encrypt_aes type ref to /grube/cl_pdf_encryption_aes.
create object o_encrypt_aes exporting keylength = 128 .
create object o_pdf.

Method SET_PERMISSION_EX:

This method sets the permissions on the PDF file. The creator can restrict certain actions (e. g. forbid printing of the document)

Parameter: (all: “X' means permission is set)

PRINT_ALLOWED: Allow printing
MODIFY_ALLOWED: Modification of PDF is allowed
COPY_ALLOWED: Copy / Paste of document contents is allowed
ADD_ANNOTATIONS_ALLOWED: User can add annotations to the PDF
FILL_FIELDS_ALLOWED: User is allowed to fill input fields (if they exist)
COPY_ACCESSIBILITY_ALLOWED: Specific extensions for accessibility are switched on

ASSEMBLE_ALLOWED: Assemble (like rotation, insertion, deletion of pages) is allowed
PRINT_HIGHRES_ALLOWED: High resolution Printing is allowed

Example: (Very restrective)

```
create object o_encrypt_aes exporting keylength = 128 .
 o_encrypt->set_permission_ex( print_allowed = ' '
                               modify_allowed = ' '
                               copy_allowed = ' '
                               add_annotations_allowed = ' '
                               fill_fields_allowed = ' '
                               copy_accessibility_allowed = ' '
                               assemble_allowed = ' '
                               print_highres_allowed = ' ' ).
```

Method SET_PASSWORDS:

This method sets the owner and the user password of the PDF. If the PDF is opened with the owner password some programs can alter the permissions. The user password should only allow the access specified under permissions. If you need to set binary passwords, please use the method SET_PASSWORDS_X instead.

Parameter:

USER_PASSWORD (String – Importing): The User password
OWNER_PASSWORD (String – Importing): The owner password.

Exceptions:

CX_SY_CODEPAGE_CONVERTER_INIT : Codepage Errors please use set_passwords_x
CX_SY_CONVERSION_CODEPAGE : Codepage Errors please use set_passwords_x
CX_PARAMETER_INVALID_TYPE : Codepage Errors please use set_passwords_x
CX_PARAMETER_INVALID_RANGE : Codepage Errors please use set_passwords_x

Example:

```
create object o_encrypt_aes exporting keylength = 128
o_encrypt->set_passwords( user_password = 'TEST' owner_password = 'ATEST' ).
```

Method SET_PASSWORDS_X:

This method sets the owner and the user password of the PDF. If the PDF is opened with the owner password some programs can alter the permissions. The user password should only allow the access specified under permissions..

Parameter:

USER_PASSWORD (xstring – Importing): The User password (binary)
OWNER_PASSWORD (xstring – Importing): The owner password (binary).

Example:

```
data: owner type xstring.
data: user type xstring.
create object o_encrypt_aes exporting keylength = 128.
owner = '54455354'.
user = ' 4154455354'.
o_encrypt->set_passwords_x( user_password = user owner_password = owner ).
```

Method GENERATE_PASSWORDS:

This method generates the internal passwords for the PDF (/O and /U entry) document. Even if the same passwords are used for different documents the /O and /U are different.

Parameter:

PDF (/GRUBE/CL_PDF – Importing): The PDF document which should be encrypted.

Exceptions:

/GRUBE/CX_PDF_DOCID_MISSING: DOCID does not exists , encryption is impossible
/GRUBE/CX_PDF_MD5_ERROR: Error calculating MD5 Hash

Example:

```
data: o_encrypt_rc4 type ref to /grube/cl_pdf_encryption_rc4.
data: o_pdf type ref to /grube/cl_pdf.


create object o_pdf.
create object o_encrypt_rc4 exporting keylength = 80 .
```

```
    o_pdf->set_pdf_file( pdf_xstring ).

*  set the user permissions for the PDF file
    o_encrypt->set_permission_ex( print_allowed = p_print1
                                  modify_allowed = p_modify
                                  copy_allowed = p_copy
                                  add_annotations_allowed = p_annot
                                  fill_fields_allowed = p_fields
                                  copy_accessibility_allowed = p_access
                                  assemble_allowed = p_assem
                                  print_highres_allowed = p_hires ).

* set the user and the owner password for the PDF
   o_encrypt->set_passwords( user_password = upass owner_password = opass ).
* generate the passwords
   o_encrypt->generate_passwords( o_pdf ).

* encrypt the PDF
   o_pdf->do_action( o_encrypt ).

*  get the pdf
    result = o_pdf->create_pdf( ).
```

Class /GRUBE/CL_PDF_BITMAP and /GRUBE/CL_PDF_GRAPHIC:

The class /GRUBE/CL_PDF_BITMAP handles all functions related to Windows Bitmaps (BMP). The class /GRUBE/CL_PDF_GRAPHIC is the basis class for all graphical PDF objects.

Constructor:

Parameter:

OPACITY (Integer – Import – Optional – Default 32 – Range 0-255)

The parameter sets opacity of the BMP. 0 = total transparent/invisible 255 = opaque

Example:

```
data: o_bitmap type ref to /grube/cl_pdf_bitmap.
create object o_bitmap exporting opacity = 32.
```

Method SET_GRAPHIC_XSTRING:

This method sets the bitmap as xstring. It expects the bitmap as a non indexed (no color

palette) with 3 or 4 byte per pixel standard Windows Bitmap (BMP).

Parameter:

IN (xstring – Importing): The bitmap as xstring

Example:

```
   data: o_pdf type ref to /grube/cl_pdf.
   data: o_bitmap type ref to /grube/cl_pdf_bitmap.

   create object o_bitmap exporting opacity = p_opa.
*   load the bitmap as XString
    o_bitmap->set_graphic_xstring( x_bmp ).

* create the PDF object
  create object o_pdf.
* load the PDF into the object (XString)
  o_pdf->set_pdf_file( result ).

  o_pdf->add_watermark_position( bmp = o_bitmap  sizex = 150 sizey = 120 x = 20 y = 60
as_background = ' ' ).
result = o_pdf->create_pdf( ).
```

Example Programs:

Please find some example programs under:

http://www.pdfencryption.de/adownloadabap.php