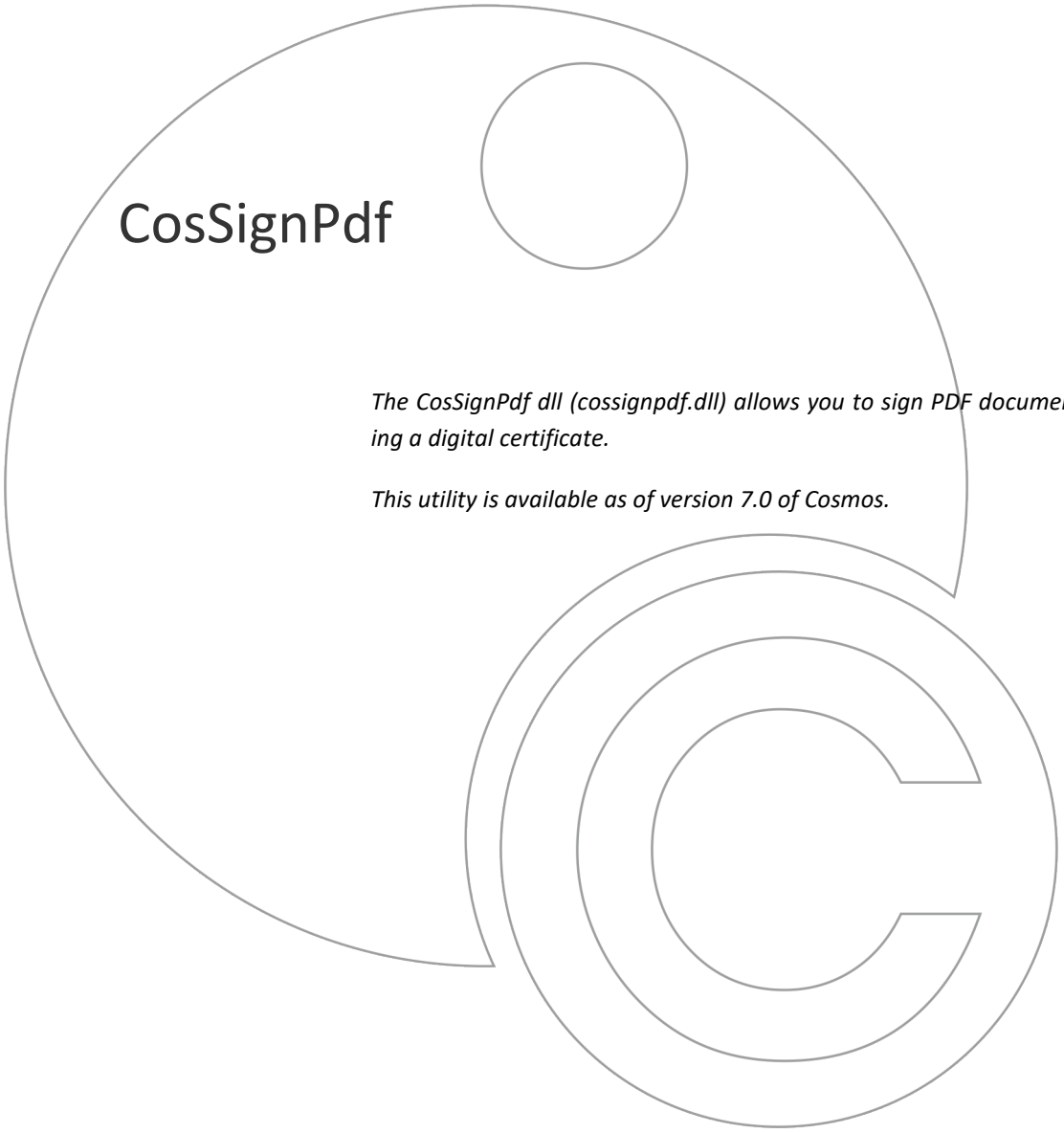
CosSignPdf

The CosSignPdf dll (cossignpdf.dll) allows you to sign PDF documents from Cosmos using a digital certificate.

This utility is available as of version 7.0 of Cosmos.



BASE100

BASE 100, S.A.
www.base100.com

Index

1. Introduction

The CosSignPdf dll (cossignpdf.dll) allows you to sign PDF documents from Cosmos using a digital certificate. This certificate must be installed in the system, either in the Windows certificate store or in a certificate store (file).

CosSignPdf uses JSignPDF software, which is freeware software developed in Java. For this reason, it is necessary that a 32-bit Java virtual machine is installed in the system, and that the directory where the JVM.DLL file is located is included in the search path.

2. Functions of the CosSignPdf DLL

The steps to sign a PDF document from Cosmos are the following:

1. Execute the CosSignPDFCreateSigner function to obtain a driver that allows us to execute the functions necessary to carry out the signature process.
2. Assignment of properties (name of the PDF file to be signed, extension of the PDF file of signed output, certificate to be used, visibility of the signature, etc.). This assignment will be made by making calls to the CosSignPDFSetProperty function (one call for each property to assign).
3. Execution of the signature process with the call to the CosSignPDFDoSign function.
4. Release of the resources used during the signature process with the execution of the CosSignPDFFreeSigner function.

2.1 CosSignPDFCreateSigner

This function returns a unique numerical identifier that will be necessary for the execution of the signature property assignment functions and for the signature process itself.

Syntax:

```
CosSignPDFCreateSigner () return Integer
```

Returns:

Unique numeric identifier needed to execute the rest of the functions of the signature process.

If the Cosmos license is not registered, it will return -1.

2.2 CosSignPDFSetProperty

This function assigns a value to a property of the signature.

Syntax:

```
CosSignPDFSetProperty (signID as integer, propertyName as char, propertyValue
as char) return Integer
```

Parameters:

signID	Identifier of the signature returned in the call to the CosSignPDFCreateSigner function.
propertyName	Name of the property to which you want to assign value. See table of properties.
propertyValue	Value that you want to assign to the property.

Returns:

0	The function has been successfully executed.
-1	The identifier passed as a parameter does not exist or if an existing property name has not been indicated.

2.3 CosSignPDFDoSign

This function allows you to make the signature of the PDF document.

Syntax:

```
CosSignPDFDoSign (signID as integer) return Integer
```

Parameters:

signID	Identifier of the signature returned in the call to the CosSignPDFCreateSigner function.

Returns:

0	The function has been successfully executed.
-1	There is no signature identifier passed as parameter.
1	Error in parameters. No PDF file or signature store has been indicated.
2	Error in the signature process.
3	The signature of at least one file has failed.
4	The signing of all PDF files has failed.

2.4 CosSignPDFSetTraceFile

This function allows you to define the name of the file where the error messages will be stored in case you want these messages not to be displayed in a window (default option, MessageBox).

Syntax:

```
CosSignPDFSetTraceFile (traceFile as char)
```

Parameters:

traceFile	Absolute path to the trace file.

2.5 CosSignPDFFreeSigner

In this function it will be indicated that the resources of the signature handler passed as a parameter are released.

After executing this function, the identifier passed as a parameter can not be used again to sign a document. A new identifier must be created by executing the CosSignPDFCreateSigner function.

Syntax:

```
CosSignPDFFreeSigner (signID as integer) return Integer
```

Parameters:

signID	Identifier of the signature returned in the call to the CosSignPDFCreateSigner function.

Returns:

0	The function has been executed successfully.
-1	The signature identifier passed as a parameter does not exist.

3. List of signature properties in the CosSignPDFSetProperty function

a) Properties related to the definition of the certificate store, certificate and passwords:

Property	Description
CERTTYPE	Indicates the type of certificates store: The possible values are: **Windows certificate store:** • WINDOWS-MY • WINDOWS-ROOT **File certificate store:** • BCPKCS12 • BKS • BOUNCYCASTLE • CASEEXACTJKS • DKS • JCEKS • JKS • PKCS12 • PKCS12-3DES-3DES • PKCS12-3DES-40RC2 • PKCS12-DEF • PKCS12-DEF-3DES-3DES • PKCS12-DEF-3DES-40RC2
KEYSTOREFILE	If the certificate store is a file, with this parameter we will indicate its absolute path.
KEYSTOREPASSWD	Password of the certificate store.
KEYALIAS	Name of certificate in the certificate store that you want to use. If no certificate is indicated, the first certificate from the certificate store will be used.
KEYPASSWD	Password of the certificate that you want to use.
KEYINDEX	If it is not possible to indicate the name of the certificate that you want to use, with this parameter we can choose it by indicating its position within the certificate store. Thus, the first certificate will be the number 0, the second the number 1, etc.

b) Properties related to the name of the source PDF, the name of the destination PDF and the folder where the signed PDF will be stored:

Property	Description
OUTFILESUFFIX	Suffix of the name of the signed PDF file. If this parameter is not indicated, the signed PDF file will have the suffix "_signed".
PDFFILENAME	Absolute path of the PDF file that you want to sign. It is possible to indicate more than one file using wildcards, for example: "C: \documents*. Pdf" indicates that all files with PDF extension that are in the folder "c:\documents" are signed.
OUTPUTFOLDER	Absolute path of the directory where you want to save the signed PDF.

c) Properties referring to the visibility / invisibility of the signature, page where it is positioned and positioning coordinates. These properties will take effect only if the VISIBLESIGNATURE property has a TRUE value.

Property	Description
VISIBLESIGNATURE	TRUE if you want the signature to be visible in the PDF document.
SIGNATURETEXT	Text that you want to show in the signature. If no value is assigned to this property, the name of the owner of the signature will be displayed.
IMAGEPATH	If you want the PDF file to show an image (digital signature, logo, etc.) in the place where the signature will be embedded, you must indicate its absolute path assigning value to this property.

Property	Description
UPPERRIGHTX	X coordinate of the upper right corner of the rectangle where the signature will be positioned.
UPPERRIGHTY	Y coordinate of the upper right vertex of the rectangle where the signature will be positioned.
LOWERLEFTX	X coordinate of the lower left corner of the rectangle where the signature will be positioned.
LOWERLEFTY	Y coordinate of the lower left corner of the rectangle where the signature will be positioned.
PAGENUMBER	Number of the page where you want to embed the signature. If no value is indicated, it will be embedded in the last page of the document.

4. Coordinate system in PDF documents

When indicating the coordinates of the rectangle where you want to position the signature in a PDF document, the following must be taken into account:

- The coordinates (0,0) of a PDF document are in the lower left corner of the document.
- Each logical unit (X and Y) is the equivalent to 1/72 of an inch, that is, 72 units is one inch (2.54 cm).
- For example, in a PDF document of A4 size (210 x 297 mm):
    - The lower left corner would have as coordinates:
        - X = 0
        - Y = 0
    - The upper left corner would have as coordinates:
        - X = 0
        - Y = (29,7/2,54) x 72 = 842
    - The upper right corner would have as coordinates:
        - X = (21/2,54) x 72 = 595
        - Y = (29,7/2,54) x 72 = 842
    - The lower right corner would have as coordinates:
        - X = (21/2,54) x 72 = 595
        - Y = 0

5. Examples

5.1 Signing PDF using the Windows store

```
public function firmaAlmacenWindows()
objects begin
    signerId as integer
    fichPDF fichPNG outputFolder as char
    i as integer
end
begin

    fichPDF = "c:\PRUSIGNPDF\input\documento.pdf";
    fichPNG = "c:\PRUSIGNPDF\firma.png";
    outputFolder = "c:\PRUJSIGNPDF\output";

    //Handler creation
    signerId = CosSignPDFCreateSigner();

    //We use Windows store certificate
    CosSignPDFSetProperty(signerId, "CERTTYPE", "WINDOWS-MY");

    //Source PDF file, destination PDF suffix, output folder
    CosSignPDFSetProperty(signerId, "OUTFILESUFFIX", "_sufijo");
    CosSignPDFSetProperty(signerId, "PDFFILENAME", fichPDF);
    CosSignPDFSetProperty(signerId, "OUTPUTFOLDER", outputFolder);

    //Visible signature and coordinates
    CosSignPDFSetProperty(signerId, "VISIBLESIGNATURE", "TRUE");
    CosSignPDFSetProperty(signerId, "IMAGEPATH", fichPNG);
    CosSignPDFSetProperty(signerId, "UPPERRIGHTX", "600");
    CosSignPDFSetProperty(signerId, "UPPERRIGHTY", "0");
    CosSignPDFSetProperty(signerId, "LOWERLEFTX", "0");
    CosSignPDFSetProperty(signerId, "LOWERLEFTY", "90");
    CosSignPDFSetProperty(signerId, "PAGENUMBER", "1");

    //Sign
    CosSignPDFDoSign(signerId).Trace();

    //Release
    CosSignPDFFreeSigner(signerId);
end
```

5.2 Signing PDF using a signature store in a JKS file

```
public function firmaKeystoreFichero()
objects begin
    signerId as integer
    fichPDF fichPNG outputFolder as char
    i as integer
end
begin

    fichPDF = "c:\PRUSIGNPDF\input\documento.pdf";
    fichPNG = "c:\PRUSIGNPDF\Box-512.png";
    outputFolder = "c:\PRUSIGNPDF\output";
```

```
    // Handler creation
    signerId = CosSignPDFCreateSigner();

    // We use certificate of certificate store in JKS format
    CosSignPDFSetProperty(signerId, "KEYSTOREFILE",
"c:\certificados\keystore.jks" );
    CosSignPDFSetProperty(signerId, "CERTTYPE", "JKS");
    CosSignPDFSetProperty(signerId, "KEYSTOREPASSWD", "certifi-
cate_store_passwd");
// We can indicate the certificate inside the certificate store with KEYALIAS
or KEYINDEX
// We use the first certificate within the certificate store
    CosSignPDFSetProperty(signerId, "KEYINDEX", "0");
    CosSignPDFSetProperty(signerId, "KEYPASSWD", "certificate_password");

    // Source PDF file, destination PDF suffix, output folder
    CosSignPDFSetProperty(signerId, "OUTFILESUFFIX", "_sufijo");
    CosSignPDFSetProperty(signerId, "PDFFILENAME", fichPDF);
    CosSignPDFSetProperty(signerId, "OUTPUTFOLDER", outputFolder);

    // Visible signature and coordinates
    CosSignPDFSetProperty(signerId, "VISIBLESIGNATURE", "TRUE");
    CosSignPDFSetProperty(signerId, "IMAGEPATH", fichPNG);
    CosSignPDFSetProperty(signerId, "UPPERRIGHTX", "600");
    CosSignPDFSetProperty(signerId, "UPPERRIGHTY", "0");
    CosSignPDFSetProperty(signerId, "LOWERLEFTX", "0");
    CosSignPDFSetProperty(signerId, "LOWERLEFTY", "90");
    CosSignPDFSetProperty(signerId, "PAGENUMBER", "1");

    //Sign
    CosSignPDFDoSign(signerId).Trace();

    //Release
    CosSignPDFFreeSigner(signerId);

end
```

5.3 Signing PDF using a signature store in P12 file with a single certificate

```
public function firmaKeystoreFicheroUnico()
objects begin
    signerId as integer
    fichPDF fichPNG outputFolder as char
    i as integer
end
begin

    fichPDF = "c:\PRUSIGNPDF\input\documento.pdf";
    fichPNG = "c:\PRUSIGNPDF\Box-512.png";
    outputFolder = "c:\PRUSIGNPDF\output";

    // Handler creation
    signerId = CosSignPDFCreateSigner();
    // We use a certificate from a certificate store in P12 format
```

```
   CosSignPDFSetProperty(signerId, "KEYSTOREFILE",
"c:\PRUSIGNPDF\certificado.p12" );
    CosSignPDFSetProperty(signerId, "KEYSTOREPASSWD", "certificate_password");

    // Source PDF file, destination PDF suffix, output folder
    CosSignPDFSetProperty(signerId, "OUTFILESUFFIX", "_sufijo");
    CosSignPDFSetProperty(signerId, "PDFFILENAME", fichPDF);
    CosSignPDFSetProperty(signerId, "OUTPUTFOLDER", outputFolder);

    // Visible signature and coordinates
    CosSignPDFSetProperty(signerId, "VISIBLESIGNATURE", "TRUE");
    CosSignPDFSetProperty(signerId, "IMAGEPATH", fichPNG);
    CosSignPDFSetProperty(signerId, "UPPERRIGHTX", "600");
    CosSignPDFSetProperty(signerId, "UPPERRIGHTY", "0");
    CosSignPDFSetProperty(signerId, "LOWERLEFTX", "0");
    CosSignPDFSetProperty(signerId, "LOWERLEFTY", "90");
    CosSignPDFSetProperty(signerId, "PAGENUMBER", "1");
    CosSignPDFSetProperty(signerId, "KEYINDEX", "0");

    //Sign
    CosSignPDFDoSign(signerId).Trace();

    //Release
    CosSignPDFFreeSigner(signerId);
end
```