

CosSignFile

La dll CosSignFile (cossigfile.dll) permite firmar digitalmente archivos en formato PDF, XML y binario, así como la firma de ficheros de factura electrónica XML en formato Facturae versiones 3.2.1 y 3.2.2.

Esta utilidad está disponible a partir de la versión 7.2 de Cosmos.

La versión de la API Cossigfile.dll que se distribuye con las versiones 7.2 a 7.6.1 de Cosmos es la 1.0.0. Esta versión utiliza las librerías de la versión 1.6 de AutoFirma.

A partir de la versión 7.8 de Cosmos, la versión de la API Cossigfile.dll que se distribuye es la 1.1.0. Esta versión utiliza las librerías de la versión 1.8 de AutoFirma.

BASE100

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

Índice

1. INTRODUCCIÓN.....	3
2. FUNCIONES DE LA DLL COSSIGNFILE.....	4
2.1 CosSIGNFILECREATESIGNER	4
2.2 CosSIGNFILEFREESIGNER	4
2.3 CosSIGNFILESIGNFACTURAE.....	5
2.4 CosSIGNFILESIGNPDF.....	5
2.5 CosSIGNFILESIGNPADES.....	6
2.6 CosSIGNFILESIGNXADES.....	7
2.7 CosSIGNFILESIGNXADES_EX	8
2.8 CosSIGNFILESIGNCADES.....	10
2.9 CosSIGNFILEGETKEYSTOREFORMATS	11
2.10 CosSIGNFILEGETSIGNALGORITHMS	11
2.11 CosSIGNFILEGETPADESSIGNFORMATS	11
2.12 CosSIGNFILEGETCADESSIGNFORMATS.....	12
2.13 CosSIGNFILEGETXADESSIGNFORMATS.....	12
2.14 CosSIGNFILEGETSIGNMODES.....	13
3. FICHERO DE LOG	14
4. CONSIDERACIONES SOBRE LAS RUTAS DE LOS FICHEROS.....	15
5. EJEMPLOS	16
5.1 DEFINICIÓN DE FUNCIONES.....	16
5.2 FIRMA DE XML EN FORMATO FACTURAE	17
5.3 FIRMA DE PDF	17
5.4 FIRMA DE PDF EN FORMATO PADES	18
5.5 FIRMA DE XML EN FORMATO XADES	18
5.6 FIRMA DE XML EN FORMATO CADES	19

© Copyright BASE 100, S.A. Todos los derechos reservados. Ninguna parte de este documento puede ser reproducida ni transmitida por medio alguno sin permiso previo por escrito del titular del copyright. Todos los productos citados en este documento son marcas registradas o marcas comerciales registradas de sus respectivos propietarios.

NTUTCosSignFile - V.2

1. Introducción

La dll **CosSignFile** (cossignfile.dll) permite firmar desde Cosmos documentos en formato PDF, XML y binarios, así como ficheros XML de factura electrónica en formato Facturae, utilizando un certificado digital. Dicho certificado deberá estar instalado en el sistema en un almacén de certificados (fichero).

CosSignFile utiliza las librerías JAVA presentes en el software Facturae, que es un software de libre distribución desarrollado en Java. Por esta razón, es necesario que en el sistema esté instalada una máquina virtual de Java de 32 bits, y que en el *path* de búsqueda se incluya el directorio donde se encuentre el fichero JVM.DLL.

2. Funciones de la dll CosSignFile

Los pasos a seguir para firmar un documento desde Cosmos son los siguientes:

1. Ejecutar la función `CosSignFileCreateSigner` para obtener un manejador que nos permita la ejecución de las funciones necesarias para realizar el proceso de firma.
2. Obtención de los literales correspondientes a los valores de las propiedades del documento a firmar (formato del almacén de certificados, algoritmo de firma, formato de firma, modo de firma). Los valores de estas propiedades se podrán obtener invocando las funciones:
 - a. `CosSignFileGetKeystoreFormats`
 - b. `CosSignFileGetSignAlgorithms`
 - c. `CosSignFileGetPAdESSignFormats`
 - d. `CosSignFileGetCAAdESSignFormats`
 - e. `CosSignFileGetXAdESSignFormats`
 - f. `CosSignFileGetSignModes`.
3. Ejecución del proceso de firma con la llamada a la función correspondiente (`CosSignFileSignFacturaE`, `CosSignFileSignPDF`, `CosSignFileSignPAdES`, `CosSignFileSignXAdES`, `CosSignFileSignCAAdES`).
4. Liberación de los recursos utilizados durante el proceso de firma con la ejecución de la función `CosSignFileFreeSigner`.

2.1 CosSignFileCreateSigner

Esta función retorna un identificador numérico único que será necesario para la ejecución de las funciones de asignación de propiedades de firma y para el proceso de firma en sí.

Sintaxis:

```
CosSignFileCreateSigner () return Integer
```

Retorna:

Identificador numérico único necesario para ejecutar el resto de funciones del proceso de firma.

Si la licencia de Cosmos no se encuentra registrada, retornará -1.

2.2 CosSignFileFreeSigner

En esta función se indicará que se liberen los recursos del manejador de firma pasado como parámetro.

Después de ejecutar esta función, el identificador pasado como parámetro no podrá volver a utilizarse para firmar un documento. Se deberá crear un nuevo identificador ejecutando la función [CosSignFileCreateSigner](#).

Sintaxis:

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

Parámetros:

signID Identificador de la firma retornado en la llamada a la función [CosSignFileCreateSigner](#).

Retorna:

0 Si la función se ha ejecutado correctamente.

-1 No existe el identificador de firma pasado como parámetro.

2.3 CosSignFileSignFacturaE

Esta función firmará en formato Facturae el fichero XML pasado como parámetro.

Sintaxis:

```
CosSignFileSignFacturaE (  
    signID as integer,  
    xmlPath as char,  
    xsigPath as char,  
    keystorePath as char,  
    keystoreFormat as char,  
    keyStorePasswd as char,  
    alias as char,  
    aliasPasswd as char) return Integer
```

Parámetros:

signID	Identificador de la firma retornado en la llamada a la función CosSignFileCreateSigner .
xmlPath	Ruta absoluta del fichero XML en formato Facturae que se desea firmar.
xsigPath	Ruta absoluta del fichero XML de salida (XSIG) resultado de la firma del fichero XML indicado en el parámetro "xmlPath".
keystorePath	Ruta absoluta del fichero de almacén de certificados donde se encuentra almacenado el certificado que se desea utilizar para realizar el proceso de firma.
keystoreFormat	Cadena de caracteres que indica el formato del almacén de certificados. Los posibles valores de este parámetro se podrán obtener invocando la función CosSignFileGetKeystoreFormats (jks, pkcs12, etc.).
keystorePasswd	Contraseña del almacén de certificados especificado en el parámetro "keystorePath".
alias	Alias del certificado electrónico que se desea utilizar para realizar la firma del archivo.
aliasPasswd	Contraseña del certificado electrónico especificado en el parámetro "alias".

Retorna:

0	No existe el identificador de firma pasado como parámetro o se ha producido un error en la ejecución de la función de firma.
1	Si la función se ha ejecutado correctamente.

2.4 CosSignFileSignPDF

Esta función firmará el fichero PDF pasado como parámetro.

Sintaxis:

```
CosSignFileSignPDF (  
    signID as integer,  
    inPDFPath as char,
```

```
outPDFPath as char,  
keystorePath as char,  
keystoreFormat as char,  
keyStorePasswd as char,  
alias as char,  
aliasPasswd as char) return Integer
```

Parámetros:

signID	Identificador de la firma retornado en la llamada a la función CosSignFileCreateSigner .
inPDFPath	Ruta absoluta del fichero PDF que se desea firmar.
outPDFPath	Ruta absoluta del fichero PDF de salida resultado de la firma del fichero PDF indicado en el parámetro "inPDFPath".
keystorePath	Ruta absoluta del fichero de almacén de certificados donde se encuentra almacenado el certificado que se desea utilizar para realizar el proceso de firma.
keystoreFormat	Cadena de caracteres que indica el formato del almacén de certificados. Los posibles valores de este parámetro se podrán obtener invocando la función CosSignFileGetKeystoreFormats (jks, pkcs12, etc.).
keystorePasswd	Contraseña del almacén de certificados especificado en el parámetro "keystorePath".
alias	Alias del certificado electrónico que se desea utilizar para realizar la firma del archivo.
aliasPasswd	Contraseña del certificado electrónico especificado en el parámetro "alias".

Retorna:

0	No existe el identificador de firma pasado como parámetro o se ha producido un error en la ejecución de la función de firma.
1	Si la función se ha ejecutado correctamente.

2.5 CosSignFileSignPAdES

Esta función firmará el fichero PDF pasado como parámetro utilizando el formato PAdES (*PDF Advanced Electronic Signatures*).

Sintaxis:

```
CosSignFileSignPAdES (  
    signID as integer,  
    inFileFullPath as char,  
    outFileFullPath as char,  
    keystorePath as char,  
    keystoreFormat as char,  
    keyStorePasswd as char,  
    alias as char,  
    aliasPasswd as char,  
    algorithm as char,  
    format as char,  
    mode as char) return Integer
```

Parámetros:

signID	Identificador de la firma retornado en la llamada a la función CosSignFileCreateSigner .
inFilePath	Ruta absoluta del fichero PDF que se desea firmar.
outFilePath	Ruta absoluta del fichero PDF de salida resultado de la firma del fichero PDF indicado en el parámetro "inFilePath".
keystorePath	Ruta absoluta del fichero almacén de certificados donde se encuentra almacenado el certificado que se desea utilizar para realizar el proceso de firma.
keystoreFormat	Cadena de caracteres que indica el formato del almacén de certificados. Los posibles valores de este parámetro se podrán obtener invocando la función CosSignFileGetKeystoreFormats (jks,pkcs12, etc.).
keystorePasswd	Contraseña del almacén de certificados especificado en el parámetro "keystorePath".
alias	Alias del certificado electrónico que se desea utilizar para realizar la firma del archivo.
aliasPasswd	Contraseña del certificado electrónico especificado en el parámetro "alias".
algorithm	Algoritmo de cifrado que se desea utilizar. La lista de algoritmos soportados se podrá consultar ejecutando la función CosSignFileGetSignAlgorithms (SHA1withRSA, SHA256withRSA, etc.).
format	Formato de firma. La lista de formatos de firma soportados se podrá consultar ejecutando la función CosSignFileGetPAdESSignFormats .
mode	Modo de firma. La lista de modos de firma soportados se podrá consultar ejecutando la función CosSignFileGetSignModes (implícito, explícito, etc.).

Retorna:

0	No existe el identificador de firma pasado como parámetro o se ha producido un error en la ejecución de la función de firma.
1	Si la función se ha ejecutado correctamente.

2.6 CosSignFileSignXAdES

Esta función firmará el fichero XML pasado como parámetro utilizando el formato XAdES (XML Advanced Electronic Signatures).

Sintaxis:

```
CosSignFileSignXAdES (  
    signID as integer,  
    inFilePath as char,  
    outFilePath as char,  
    keystorePath as char,  
    keystoreFormat as char,  
    keyStorePasswd as char,  
    alias as char,  
    aliasPasswd as char,
```

```
algorithm as char,  
format as char,  
mode as char) return Integer
```

Parámetros:

signID	Identificador de la firma retornado en la llamada a la función CosSignFileCreateSigner .
inFilePath	Ruta absoluta del fichero XML que se desea firmar.
outFilePath	Ruta absoluta del fichero XML de salida resultado de la firma del fichero XML indicado en el parámetro "inFilePath".
keystorePath	Ruta absoluta del fichero de almacén de certificados donde se encuentra almacenado el certificado que se desea utilizar para realizar el proceso de firma.
keystoreFormat	Cadena de caracteres que indica el formato del almacén de certificados. Los posibles valores de este parámetro se podrán obtener invocando la función CosSignFileGetKeystoreFormats (jks,pkcs12, etc.).
keystorePasswd	Contraseña del almacén de certificados especificado en el parámetro "keystorePath".
alias	Alias del certificado electrónico que se desea utilizar para realizar la firma del archivo.
aliasPasswd	Contraseña del certificado electrónico especificado en el parámetro "alias".
algorithm	Algoritmo de cifrado que se desea utilizar. La lista de algoritmos soportados se podrá consultar ejecutando la función CosSignFileGetSignAlgorithms (SHA1withRSA, SHA256withRSA, etc.).
format	Formato de firma. La lista de formatos de firma soportados se podrá consultar ejecutando la función CosSignFileGetXAdESSignFormats .
mode	Modo de firma. La lista de modos de firma soportados se podrá consultar ejecutando la función CosSignFileGetSignModes (implícito, explícito, etc.).

Retorna:

0	No existe el identificador de firma pasado como parámetro o se ha producido un error en la ejecución de la función de firma.
1	Si la función se ha ejecutado correctamente.

2.7 CosSignFileSignXAdES_Ex

Esta función firmará el fichero XML pasado como parámetro utilizando el formato XAdES.

Sintaxis:

```
CosSignFileSignXAdES_Ex (  
    signID as integer,  
    inFilePath as char,  
    outFilePath as char,  
    keystorePath as char,  
    keystoreFormat as char,
```



```

    keyStorePasswd as char,
    alias as char,
    aliasPasswd as char,
    algorithm as char,
    format as char,
    mode as char,
    extraParams as char)
    return Integer
  
```

Parámetros:

signID	Identificador de la firma devuelto por la llamada a la función CosSignFileCreateSigner .
inPDFPath	Ruta absoluta del fichero XML que se desea firmar.
outPDFPath	Ruta absoluta del fichero XML de salida resultado de la firma del fichero XML indicado en el parámetro "inPDFPath".
keystorePath	Ruta absoluta del almacén de certificados donde se encuentra el certificado que se desea utilizar para realizar el proceso de firma.
keystoreFormat	Cadena de caracteres que indica el formato del almacén de certificados. Los posibles valores de este parámetro se podrán obtener invocando a la función CosSignFileGetKeystoreFormats (jks, pkcs12, etc.).
keystorePasswd	Contraseña del almacén de certificados especificado en el parámetro "keystorePath".
alias	Alias del certificado electrónico que se desea utilizar para realizar la firma del archivo.
aliasPasswd	Contraseña del certificado electrónico especificado en el parámetro "alias".
algorithm	Algoritmo de cifrado que se desea utilizar. La lista de algoritmos soportados se podrá consultar ejecutando la función CosSignFileGetSignAlgorithms (SHA1withRSA, SHA256withRSA, etc.).
format	Formato de firma. La lista de formatos de firma soportados se podrá consultar ejecutando la función CosSignFileGetPAdESSignFormats .
mode	Modo de firma. La lista de modos de firma soportados se podrá consultar ejecutando la función CosSignFileGetSignModes (implícito, explícito, etc.).
extraParams	Lista de parámetros adicionales. Los posibles valores son: policyIdentifier, policyIdentifierHash, policyIdentifierHashAlgorithm, policyDescription y policyQualifier.

El formato es: clave:valor

```

extraParams = "policyIdentifier":"X.XX.XXX.X.X.X.X.X.X"
+ ";"
+ "policyIdentifierHash":"V81VVGDCPen6VELRD1Ja8HARFk="
+ ";"
  
```

```
+ "policyIdentifierHashAlgorithm":"SHA-256"  
+ ";"  
+ "policyDescription":"Firma electronica"  
+ ";"  
+ "policyQualifier":"http://paginaweb";
```

2.8 CosSignFileSignCADES

Esta función firmará el fichero pasado como parámetro utilizando el formato CADES (CMS Advanced Electronic Signatures).

Sintaxis:

```
CosSignFileSignCADES (  
    signID as integer,  
    inFilePath as char,  
    outFilePath as char,  
    keystorePath as char,  
    keystoreFormat as char,  
    keyStorePasswd as char,  
    alias as char,  
    aliasPasswd as char,  
    algorithm as char,  
    format as char,  
    mode as char) return Integer
```

Parámetros:

signID	Identificador de la firma retornado en la llamada a la función CosSignFileCreateSigner .
inFilePath	Ruta absoluta del fichero que se desea firmar.
outFilePath	Ruta absoluta del fichero de salida resultado de la firma del fichero indicado en el parámetro "inFilePath".
keystorePath	Ruta absoluta del fichero de almacén de certificados donde se encuentra almacenado el certificado que se desea utilizar para realizar el proceso de firma.
keystoreFormat	Cadena de caracteres que indica el formato del almacén de certificados. Los posibles valores de este parámetro se podrán obtener invocando la función CosSignFileGetKeystoreFormats (jks,pkcs12, etc.).
keystorePasswd	Contraseña del almacén de certificados especificado en el parámetro "keystorePath".
alias	Alias del certificado electrónico que se desea utilizar para realizar la firma del archivo.
aliasPasswd	Contraseña del certificado electrónico especificado en el parámetro "alias".
algorithm	Algoritmo de cifrado que se desea utilizar. La lista de algoritmos soportados se podrá consultar ejecutando la función CosSignFileGetSignAlgorithms (SHA1withRSA, SHA256withRSA, etc.).

format	Formato de firma. La lista de formatos de firma soportados se podrá consultar ejecutando la función CosSignFileGetCAdESSignFormats .
mode	Modo de firma. La lista de modos de firma soportados se podrá consultar ejecutando la función CosSignFileGetSignModes (implícito, explícito, etc.).
Retorna:	
0	No existe el identificador de firma pasado como parámetro o se ha producido un error en la ejecución de la función de firma.
1	Si la función se ha ejecutado correctamente.

2.9 CosSignFileGetKeystoreFormats

Esta función permite consultar la lista de posibles tipos de almacenes de certificados soportados por la API.

Sintaxis:

```
CosSignFileGetKeystoreFormats (signID as integer) return Char
```

Parámetros:

signID Identificador de la firma retornado en la llamada a la función [CosSignFileCreateSigner](#).

Retorna:

Lista de posibles tipos de almacenes de certificados, separados por comas, soportados por la API. Si no existe el identificador de firma pasado como parámetro o se produce un error en la ejecución de la función retornará un String vacío.

2.10 CosSignFileGetSignAlgorithms

Esta función permite consultar la lista de posibles algoritmos de cifrado soportados por la API.

Sintaxis:

```
CosSignFileGetSignAlgorithms (signID as integer) return Char
```

Parámetros:

signID Identificador de la firma retornado en la llamada a la función [CosSignFileCreateSigner](#).

Retorna:

Lista de posibles algoritmos de cifrado, separados por comas, soportados por la API. Si no existe el identificador de firma pasado como parámetro o se produce un error en la ejecución de la función retornará un String vacío.

2.11 CosSignFileGetPAdESSignFormats

Esta función permite consultar la lista de posibles formatos de firma PAdES soportados por la API.

Sintaxis:

```
CosSignFileGetPAdESSignFormats (signID as integer) return Char
```

Parámetros:

signID Identificador de la firma retornado en la llamada a la función [CosSignFileCreateSigner](#).

Retorna:

Lista de posibles formatos de firma PAdES, separados por comas, soportados por la API. Si no existe el identificador de firma pasado como parámetro o se produce un error en la ejecución de la función retornará un String vacío.

2.12 CosSignFileGetCAdeSSignFormats

Esta función permite consultar la lista de posibles formatos de firma CADES soportados por la API.

Sintaxis:

```
CosSignFileGetCAdeSSignFormats (signID as integer) return Char
```

Parámetros:

signID Identificador de la firma retornado en la llamada a la función [CosSignFileCreateSigner](#).

Retorna:

Lista de posibles formatos de firma CADES, separados por comas, soportados por la API. Si no existe el identificador de firma pasado como parámetro o se produce un error en la ejecución de la función retornará un String vacío.

2.13 CosSignFileGetXAdESSignFormats

Esta función permite consultar la lista de posibles formatos de firma XAdES soportados por la API.

Sintaxis:

```
CosSignFileGetXAdESSignFormats (signID as integer) return Char
```

Parámetros:

signID Identificador de la firma retornado en la llamada a la función [CosSignFileCreateSigner](#).

Retorna:

Lista de posibles formatos de firma XAdES, separados por comas, soportados por la API. Si no existe el identificador de firma pasado como parámetro o se produce un error en la ejecución de la función retornará un String vacío.

2.14 CosSignFileGetSignModes

Esta función permite consultar la lista de posibles modos de firma soportados por la API.

Sintaxis:

```
CosSignFileGetSignModes (signID as integer) return Char
```

Parámetros:

signID Identificador de la firma retornado en la llamada a la función [CosSignFileCreateSigner](#).

Retorna:

Lista de posibles modos de firma, separados por comas, soportados por la API. Si no existe el identificador de firma pasado como parámetro o se produce un error en la ejecución de la función retornará un String vacío.

3. Fichero de log

La DLL CosSignFile.dll permite la creación de un log de operaciones en formato Log4Java definiendo el fichero log4j.properties en el directorio donde se encuentre el archivo CosSignFile.dll.

Ejemplo:

```
# Root logger option
log4j.rootLogger=INFO, stdout, file

# Redirect log messages to console
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
%c{1}:%L - %m%n

# Redirect log messages to a log file, support file rolling.
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=c:/tmp/log4j-cosmoSigner.log
log4j.appender.file.MaxFileSize=5MB
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
%c{1}:%L - %m%n
```

En el atributo “log4j.rootLogger” se indica el nivel de log. Los valores posibles son ALL, DEBUG, ERROR, FATAL, INFO, OFF, TRACE y WARN.

El atributo “log4j.appender.file.File” indicará la ruta del fichero de log.

Para más información, consulte la documentación de Log4j en la siguiente dirección:
<https://logging.apache.org/log4j>

4. Consideraciones sobre las rutas de los ficheros

La DLL CosSignFile.dll hace uso de una serie de clases programadas en lenguaje Java para realizar el proceso de firma de archivos XML, PDF y binarios. Por esta razón, las rutas de los archivos de almacén de firmas, ficheros XML, ficheros PDF y ficheros binarios deben tener como separador de directorio el carácter “/”, en lugar del “\” propio de Microsoft Windows.

5. Ejemplos

5.1 Definición de funciones

```
public dll "cossignfile.dll" CosSignFileCreateSigner() return integer
public dll "cossignfile.dll" CosSignFileFreeSigner(signer as integer)
public dll "cossignfile.dll" CosSignFileSignFacturaE(
    signer as integer,
    xmlPath as char,
    xsigPath as char,
    keystorePath as char,
    keystoreFormat as char,
    keyStorePasswd as char,
    alias as char,
    aliasPasswd as char) return integer

public dll "cossignfile.dll" CosSignFileSignPDF(
    signer as integer,
    inPDFPath as char,
    outPDFPath as char,
    keystorePath as char,
    keystoreFormat as char,
    keyStorePasswd as char,
    alias as char,
    aliasPasswd as char) return integer

public dll "cossignfile.dll" CosSignFileSignPAdES(
    signer as integer,
    inFileFullPath as char,
    outFileFullPath as char,
    keystorePath as char,
    keystoreFormat as char,
    keyStorePasswd as char,
    alias as char,
    aliasPasswd as char,
    algorithm as char,
    format as char,
    mode as char) return integer

public dll "cossignfile.dll" CosSignFileSignXAdES(
    signer as integer,
    inFileFullPath as char,
    outFileFullPath as char,
    keystorePath as char,
    keystoreFormat as char,
    keyStorePasswd as char,
    alias as char,
    aliasPasswd as char,
    algorithm as char,
    format as char,
    mode as char) return integer

public dll "cossignfile.dll" CosSignFileSignCAdES(
    signer as integer,
    inFileFullPath as char,
    outFileFullPath as char,
    keystorePath as char,
    keystoreFormat as char,
    keyStorePasswd as char,
    alias as char,
    aliasPasswd as char,
    algorithm as char,
    format as char,
    mode as char) return integer

public dll "cossignfile.dll" CosSignFileGetKeystoreFormats(signer as integer) return
char
public dll "cossignfile.dll" CosSignFileGetSignAlgorithms(signer as integer) return
char
```



```
public dll "cossignfile.dll" CosSignFileGetPAdESSignFormats (signer as integer) return char
public dll "cossignfile.dll" CosSignFileGetCAAdESSignFormats (signer as integer) return char
public dll "cossignfile.dll" CosSignFileGetXAdESSignFormats (signer as integer) return char
public dll "cossignfile.dll" CosSignFileGetSignModes (signer as integer) return char
```

5.2 Firma de XML en formato Facturae

```
private function firmaFacturaE_P12
objects begin
    signer as integer
    xmlPath as char
    xsigPath as char
    p12Path as char
    keyStorePasswd as char
    alias as char
    aliasPasswd as char
end
begin

    xmlPath = ProjectDir() + "\ficheros\factura_para_firmar.xml";
    xsigPath = ProjectDir() + "\ficheros\out\factura_firmada.xsig";
    p12Path = ProjectDir() + "\ficheros\p12depruebas.p12";

    xmlPath.Replace("\", "/");
    xsigPath.Replace("\", "/");
    p12Path.Replace("\", "/");

    keyStorePasswd = "contraseña";
    alias = "Pepito López";
    aliasPasswd = "contraseña";

    signer = CosSignFileCreateSigner();
    CosSignFileSignFacturaE(signer, xmlPath, xsigPath, p12Path, "pkcs12",
    keyStorePasswd, alias, aliasPasswd);
    CosSignFileFreeSigner(signer);

end
```

5.3 Firma de PDF

```
private function firmaPDF_P12
objects begin
    signer as integer
    inPDFPath as char
    outPDFPath as char
    p12Path as char
    keyStorePasswd as char
    alias as char
    aliasPasswd as char
end
begin

    inPDFPath = ProjectDir() + "\ficheros\Windows_Server_2003_Migration_Datasheet.pdf";
    outPDFPath = ProjectDir() +
"\ficheros\out\Windows_Server_2003_Migration_Datasheet.firmado.pdf";
    p12Path = ProjectDir() + "\ficheros\p12depruebas.p12";

    inPDFPath.Replace("\", "/");
    outPDFPath.Replace("\", "/");
    p12Path.Replace("\", "/");

    keyStorePasswd = "contraseña";
    alias = "Pepito López";
    aliasPasswd = "contraseña";
```

```
    signer = CosSignFileCreateSigner();
    CosSignFileSignPDF(signer, inPDFPath, outPDFPath, p12Path, "pkcs12",
    keyStorePasswd, alias, aliasPasswd);
    CosSignFileFreeSigner(signer);

end
```

5.4 Firma de PDF en formato PAdES

```
private function firmaPAdES_P12
objects begin
    signer as integer
    inFileDir as char
    outFileDir as char
    p12Path as char
    keyStorePasswd as char
    alias as char
    aliasPasswd as char
    algorithm as char
    format as char
    mode as char
end
begin

    inFileDir = ProjectDir() + "\ficheros\Windows_Server_2003_Migration_Datasheet.pdf";
    outFileDir = ProjectDir() +
"\ficheros\out\Windows_Server_2003_Migration_Datasheet_PAdES.pdf";
    p12Path = ProjectDir() + "\ficheros\p12depruebas.p12";

    inFileDir.Replace("\", "/");
    outFileDir.Replace("\", "/");
    p12Path.Replace("\", "/");

    keyStorePasswd = "contraseña";
    alias = "Pepito López";
    aliasPasswd = "contraseña";

    format = "PAdES";
    mode = "implicit";
    algorithm = "SHA512withRSA";

    signer = CosSignFileCreateSigner();
    CosSignFileSignPAdES(signer, inFileDir, outFileDir, p12Path, "pkcs12",
    keyStorePasswd, alias, aliasPasswd, algorithm, format, mode);
    CosSignFileFreeSigner(signer);

end
```

5.5 Firma de XML en formato XAdES

```
private function firmaXAdES_P12
objects begin
    signer as integer
    inFileDir as char
    outFileDir as char
    p12Path as char
    keyStorePasswd as char
    alias as char
    aliasPasswd as char
    algorithm as char
    format as char
    mode as char
end
begin

    inFileDir = ProjectDir() + "\ficheros\factura.xml";
    outFileDir = ProjectDir() + "\ficheros\out\factura_salida_XAdES.xsig";
```

```

p12Path = ProjectDir() + "\ficheros\p12depruebas.p12";

inFilePath.Replace("\", "/");
outFilePath.Replace("\", "/");
p12Path.Replace("\", "/");

keyStorePasswd = "contraseña";
alias = "Pepito López";
aliasPasswd = "contraseña";

format = "XAdES Detached";
mode = "implicit";
algorithm = "SHA512withRSA";

signer = CosSignFileCreateSigner();
CosSignFileSignXAdES(signer, inFilePath, outFilePath, p12Path, "pkcs12",
keyStorePasswd, alias, aliasPasswd, algorithm, format, mode);
CosSignFileFreeSigner(signer);

end

```

5.6 Firma de XML en formato CAdES

```

private function firmaCAdES_P12
objects begin
    signer as integer
    inFilePath as char
    outFilePath as char
    p12Path as char
    keyStorePasswd as char
    alias as char
    aliasPasswd as char
    algorithm as char
    format as char
    mode as char
end
begin

inFilePath = ProjectDir() + "\ficheros\factura.xml";
outFilePath = ProjectDir() + "\ficheros\out\factura_CAdES.xml";
p12Path = ProjectDir() + "\ficheros\p12depruebas.p12";

inFilePath.Replace("\", "/");
outFilePath.Replace("\", "/");
p12Path.Replace("\", "/");

keyStorePasswd = "contraseña";
alias = "Pepito López";
aliasPasswd = "contraseña";

format = "XAdES Detached";
mode = "implicit";
algorithm = "SHA512withRSA";

signer = CosSignFileCreateSigner();
CosSignFileSignCAdES(signer, inFilePath, outFilePath, p12Path, "pkcs12",
keyStorePasswd, alias, aliasPasswd, algorithm, format, mode);
CosSignFileFreeSigner(signer);

end

```