



MultiBase Cosmos

Notas a la versión 6.0, rel. 4

BASE100

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

Índice

1. IMPLEMENTACIONES	3
2. FUNCIONES DE LA COSHTTPDLL	4
2.1 COSHTTPSETCERTIFICATE	4
2.2 COSHTTPSETSSLKEY	4
2.3 COSHTTPSETCERTIFICATE TYPE	5
2.4 COSHTTPSETPRIVATEKEYPASSWORD	5
3. CTSQL.....	6
4. VARIABLES DE ENTORNO	7
5. CORRECCIONES	8

© 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.

[NT_CO_6.0.4_v1.1]

1. Implementaciones

En esta versión se ha implementado la posibilidad de realizar autenticaciones con certificado de cliente en un servidor web al que se accede con la DLL COSHTTPDLL.

Esta nueva funcionalidad va a permitir, por ejemplo, la autenticación con certificado para el envío de facturas al servicio web del SII (Suministro Inmediato de Información) del Ministerio de Hacienda y Función Pública.

Los ficheros del certificado tendrán que estar en formato PEM (*Privacy-enhanced Electronic Mail*). Si el certificado no estuviera en este formato, habría que convertirlo utilizando alguna de las herramientas disponibles en el mercado.

2. Funciones de la Coshttpdll

A continuación se detallan las funciones de esta DLL implementadas para la autenticación con certificado cliente en un servidor web.

2.1 CosHttpSetCertificate

Esta función permite indicar a la DLL la ruta del fichero de certificado. Los ficheros del certificado tendrán que estar en formato PEM.

Declaración:

```
public dll "coshttpdll.dll" CosHttpSetCertificate(requestID as integer, certificateFile as char) return integer
```

Parámetros

requestID	Número entero con el ID de la REQUEST creada con la función CosHttpNewRequest.
certificateFile	Ruta del fichero de certificado.

Retorna:

0	Si no hubo error.
-1	En caso de que no exista el ID de la REQUEST pasada como primer parámetro.

2.2 CosHttpSetSSLKey

Esta función permite indicar a la DLL la ruta del fichero de clave privada.

Declaración:

```
public dll "coshttpdll.dll" CosHttpSetSSLKey(requestID as integer, sslKey as char) return integer
```

Parámetros:

requestID	Número entero con el ID de la REQUEST creada con la función CosHttpNewRequest.
sslKey	Ruta del fichero de clave privada.

Retorna:

0	Si no hubo error.
-1	En caso de que no exista el ID de la REQUEST pasada como primer parámetro.

2.3 CosHttpSetCertificateType

Esta función permite indicar a la DLL el tipo del fichero de certificado especificado en la función CosHttpSetCertificate.

Declaración:

```
public dll "coshttdll.dll" CosHttpSetCertificateType(requestID as integer,  
certificateType as char) return integer
```

Parámetros:

requestID	Número entero con el ID de la REQUEST creada con la función CosHttpNewRequest
certificateType	Tipo de certificado. Valor soportado: "PEM".

Retorna:

0	Si no hubo error.
-1	En caso de que no exista el ID de la REQUEST pasada como primer parámetro.

Si no se ejecuta esta función, el tipo por defecto será "PEM".

2.4 CosHttpSetPrivateKeyPassword

Esta función permite indicar la contraseña de la clave privada del certificado.

Declaración:

```
public dll "coshttdll.dll" CosHttpSetPrivateKeyPassword(requestID as integer,  
passwd as char) return integer
```

Parámetros:

requestID	Número entero con el ID de la REQUEST creada con la función CosHttpNewRequest.
Passwd	Contraseña de la clave privada del certificado.

Retorno:

0	Si no hubo error.
-1	En caso de que no exista el ID de la REQUEST pasada como primer parámetro.

NOTA: Para más información, consulte el **Ejemplo de autenticación certificado** en el apartado "API para conexiones HTTP: Coshttdll" en el manual de APIs de la ayuda Cosmos.

3. CTSQL

El motor CTSQL optimiza las sentencias SQL para elegir un índice de acceso cuando se encuentra varios operadores OR por una misma columna que forma parte de un índice (optimización por rangos).

Si se encuentra más de una columna en los OR de la WHERE de la sentencia SQL, por defecto no optimiza.

Se ha implementado la variable de entorno OPTIMIZECOMPOSITEORRANGE para que el motor optimice la sentencia SQL cuando en la cláusula WHERE se encuentre más de una columna por rango.

Los posibles valores son:

TRUE/YES	El motor CTSQL intentará optimizar cuando se encuentre más de una columna por rango en la WHERE.
FALSE/NO	El motor CTSQL no realizará ninguna optimización cuando en la cláusula WHERE se encuentre más de una columna por rango.

Esta variable de entorno se define en el fichero de entorno del motor (ctsql.ini).

4. Variables de entorno

ODBCDESCRIBEVARCHAR

La función `SqlDescribe` de la clase `SqlServer` permitirá retornar **varchar** en lugar de **char** para las columnas de tipo **varchar** en conexiones ODBC.

Para ello, se deberá definir en la conexión la variable de entorno **ODBCDESCRIBEVARCHAR** con el valor `TRUE` o `YES`.

Si no se define esta función o se define con el valor `NO` o `FALSE`, retornará **char**.

OPTIMIZECOMPOSITEORRANGE

Si se define esta variable de entorno a `YES` o `TRUE`, el motor `CTSQL` intentará optimizar cuando se encuentre más de una columna por rango en la cláusula `WHERE`.

5. Correcciones

- Error de ejecución en la función GetLookColsFKTable de la DLL REPDLL.DLL cuando la suma de los nombres de las columnas de lookup de una FK superaba los 200 bytes.
- En algunos casos, el método GetProperty de la clase ACTIVEX retornaba FALSE cuando debería retornar TRUE.
- Cuando un PC tenía más de un monitor y se abría un Form sin padre con Open(NULL) o Run(NULL), el Form se abría en el primer monitor en lugar de abrirse en el monitor donde se encontraba el Form actual.
- Error de ejecución en el método SqlExec de la clase SqlServer cuando la frase a ejecutar superaba los 8.192 bytes.
- Se ha optimizado la búsqueda de nodos en ficheros XML cuando se utiliza la DLL TTXMLDLL.