

MultiBase Cosmos

Notas a la versión 4.4, release 4

BASE100

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

Índice

1.	IMPLEMENTACIONES	3
1.1	MÉTODOS DE LA CLASE MODULE	3
1.2	MÉTODOS DE LA CLASE SQLSTATEMENT	5
1.3	MÉTODOS DE LA CLASE SIMPLECONTROL	5
1.4	MÉTODOS DE LA CLASE PRNDOCUMENT	6
1.5	VARIABLES DE ENTORNO	6
1.6	MANEJO DE FICHEROS XML	6
1.7	DBMANAGER	7
2.	CORRECCIONES	8
3.	ANEXO I. REALIZACIÓN DE COPIAS DE SEGURIDAD EN CALIENTE	10
3.1	FUNCIONAMIENTO	10
3.2	VARIABLES DE ENTORNO	11
3.3	EJEMPLO	11
4.	ANEXO II. CIFRADO (ENCRIPCIÓN) DE LA BASE DE DATOS	13
4.1	ACCESO A BASES DE DATOS CIFRADAS	13
4.2	VARIABLES DE ENTORNO	14
4.3	CREACIÓN DE UNA BASE DE DATOS CIFRADA	14
4.4	REPARACIÓN Y CHEQUEO DE BASES DE DATOS CIFRADAS	15
4.4.1	<i>Comando trepidx</i>	15
4.4.2	<i>Comando tchkidx</i>	16

1. Implementaciones

- Cosrep. Nuevo parámetro “-sqlserver servidoresql”. Evita que se muestre la ventana de selección de formato SQL para el que se va a generar el SQL de creación de la base de datos a partir del repositorio. Los valores son: MultiBase, Informix, Oracle, SQL Server.
- Nuevo método [FtpPutFile](#) de la clase Module. Permite enviar un fichero a un servidor a través de FTP. Los ficheros se envían en binario.
- Nuevo método [CallWebService](#) de la clase Module. Permite establecer una comunicación entre Cosmos y un Servicio Web.
- Nuevo método [TTXmlGetEncodedNodeText](#) de TTXMLDLL.DLL. Recibe como parámetro el “encoding” en el que desea que se retorne el texto del nodo.
- Nuevo método [TTXmlNewEncodedTextChild](#) de TTXMLDLL.DLL. Permite añadir un nuevo nodo de tipo texto a un nodo padre indicando el “charset” (juego) de caracteres.
- Nuevo método [GetLastStatement](#) de la clase SqlStatement. Devuelve la última instrucción SQL ejecutada. En ella se sustituirán los valores asignados a las variables host.
- Editor visual de Cosmos. Posibilidad de ver la versión y el upgrade de Cosmos.
- Nuevo método [GetSysEnvVar](#) de la clase Module. Retorna el valor de una variable de entorno del sistema que recibe como parámetro.
- Nuevo método [DrawListLines](#) de la clase SimpleControl. Permite indicar sobre qué controles tipo lista se pueden dibujar líneas separadoras entre filas.
- Posibilidad de utilizar iconos en formato PNG.
- Posibilidad de cargar ficheros PNG en controles tipo Bitmap.
- Debugger. Posibilidad de habilitar y deshabilitar la visualización de objetos tipo Array.
- Nuevo método [HideExportExcelButton](#) de la clase PrnDocument. Permite ocultar el botón ExportExcel de la ventana de preview.
- CTSQL. Posibilidad de realizar copias de seguridad en caliente de la base de datos en curso (ver [Anexo I](#)).
- CTSQL. Posibilidad de cifrar/descifrar una base de datos (ver [Anexo II](#)).
- Posibilidad de utilizar la aplicación Monitor en Windows Vista y Windows Server 2008.
- Nuevas utilidades en la Aplicación [DBManager](#).

1.1 Métodos de la Clase Module

- FtpPutFile. Permite subir un fichero a un servidor FTP.

Sintaxis:

```
FtpPutFile (remoteHost as Char ,remoteHostPort as Integer, remoteHostUser as Char, remoteHostPassword as Char, remoteHostDirectory as Char, remoteHostFile as Char, localFile as Char, showDialog as Boolean, allowCancelUpload as Boolean) return Integer
```

Parámetros:

remoteHost	IP o nombre del Host.
remoteHostPort	Número del puerto del FTP. Si se indica un número negativo o el cero el número del puerto será el 21.
remoteHostUser	Usuario.
remoteHostPassword	Password.
remoteHostDirectory	Ruta completa del directorio destino.
RemoteHostFile	Nombre del fichero.
LocalFile	Ruta completa del fichero que se va a enviar.
ShowDialog	Indica si se mostrará la ventana de progreso o no.
AllowCancelUpload	Indica si se va a dar al usuario la posibilidad de cancelar la operación.

Retorna los siguientes códigos:

- 0. Si el fichero se ha enviado correctamente.
 - -1. No existe fichero origen o no se puede acceder.
 - -2. No se puede conectar a la red.
 - -3. No se encuentra servidor.
 - -4. No se puede acceder al directorio destino.
 - -5. Error en el envío del fichero.
 - -6. Abortado por el usuario.
 - -7. Timeout.
 - -8. Mal usuario/contraseña.
 - -9. Parámetros host, remotehostfile y localfile no pueden ser nulos.
- GetSysEnvVar. Devuelve el valor de una variable de entorno del sistema.

Sintaxis:

```
GetSysEnvVar(name as Char) return Char
```

Parámetros:

Name	Nombre de la variable de entorno del sistema operativo de la cual se desea consultar su valor.
------	--

Retorna:

Un string con el valor de la variable, o null si la variable no estuviera definida.

- **CallWebService.** Permite establecer una comunicación con un Servicio Web. Al efectuar una petición, el Servicio Web la ejecuta y envía la respuesta y el resultado de esa ejecución al cliente Cosmos.

El intercambio de datos se realiza con objetos Char en formato XML.

Sintaxis:

```
CallWebService(serverURL as Char ,VAR headerSend as Char ,VAR headerReceive as Char ,outFile as Char ,SOAPAction as Char)
```

Parámetros:

serverURL	URL del servidor donde se encuentra alojado el Web Service.
headerSend	Objeto de la clase Char que contiene, en formato XML, la descripción de la operación que se desea ejecutar (Request). Retorna un Char con el mensaje POST enviado al servidor. Este parámetro se pasa por referencia.
headerReceive	Objeto de la clase Char que recibe la descripción, en formato XML, de la respuesta del servidor (Response) a la petición lanzada desde el parámetro headerSend. Este parámetro es pasado por referencia.
outFile	Ruta del fichero donde se almacena la respuesta a la ejecución de la operación. Esta respuesta será en formato XML.
SOAPAction	Indica el valor de la cabecera SOAPAction para la operación indicada en el parámetro headerSend.

NOTA: El protocolo que soporta el cliente Cosmos para los servicios Web es SOAP (Simple Object Access Protocol).

1.2 Métodos de la Clase SqlStatement

- **GetLastStatement.** Devuelve la última instrucción SQL ejecutada. En ella se sustituirán los valores asignados a las variables host.

Sintaxis:

```
GetLastStatement() return Char
```

Retorna:

Un Char con la instrucción SQL si el método se ejecuta antes de la llamada al método "execute".

Un Char con la instrucción SQL y los valores de las variables host si la llamada al método se realiza después del método "execute".

1.3 Métodos de la Clase SimpleControl

- **DrawListLines.** Permite indicar sobre qué controles de tipo lista se pueden dibujar líneas separadoras entre filas.

Este método depende del valor de la variable de entorno DRAWLISTLINES.

Si el valor de la variable es TRUE las líneas se dibujarán en todos los controles.

Si el valor de la variable es FALSE no se dibujará ninguna línea.

Si la variable de entorno no está definida, las líneas se dibujarán en los controles en los que se aplique este método.

Sintaxis:

```
DrawListLines (drawlistlines as boolean)
```

Parámetros:

drawlistlines	Indica si se desean pintar las líneas o no.
---------------	---

1.4 Métodos de la Clase PrnDocument

- HideExportExcelButton. Permite ocultar el botón ExportExcel del preview.

Sintaxis:

```
HideExportExcelButton (hide as boolean)
```

Parámetros:

hide	Indica si se desea ocultar el botón que aparece en la ventana del preview.
------	--

Sus posibles valores son TRUE y FALSE.

El valor por defecto es FALSE (no se oculta el botón).

1.5 Variables de Entorno

LOCKWAITMS

Nueva variable de entorno en el motor Windows para establecer el tiempo de espera entre reintento de bloqueos. El valor de esta variable indica el número de milisegundos que transcurrirán entre un intento de bloqueo y el siguiente. Es una variable de entorno del gestor de base de datos CTSQL y debe definirse en el fichero de configuración del motor CTSQL.

1.6 Manejo de Ficheros XML

- TTXmlGetEncodedNodeText. Este método recibe como parámetro el encoding en el que desea que se retorne el texto del nodo.

Sintaxis:

```
TTXmlGetEncodedNodeText (Doc as integer, Node as integer, Charset as integer)  
return char
```

Parámetros:

Doc	Identificador del documento XML.
-----	----------------------------------

Node	Identificador del nodo del cual queremos consultar el texto asociado.
------	---

Charset Identificador del set de caracteres con el que se va a crear esta propiedad.

Retorna:

El texto del nodo.

- `TTXmlNewEncodedTextChild`. Este método permite crear un nuevo nodo XML de tipo texto, indicando el nodo padre del documento XML al que pertenece, y pudiendo pasarle, además, el "charset" (juego de caracteres) como parámetro.

Sintaxis:

```
TTXmlNewEncodedTextChild(Doc as integer, ParentNode as integer, name as Char, Content as Char, charset as integer) return integer
```

Parámetros:

Doc Identificador del documento XML.

ParentNode Identificador del nodo padre.

name Nombre del nuevo nodo.

Content Texto que se va a añadir.

Charset Identificador del juego de caracteres con el que se va a crear esta propiedad.

Retorna:

El identificador del nodo.

1.7 DBManager

- Posibilidad de generar un fichero en el que se descarguen los datos de una tabla utilizando la instrucción Insert. Esta opción se podrá llevar a cabo desde el menú de columnas de una tabla.
- Posibilidad de generar el fichero Load para carga de la base de datos, teniendo en cuenta su integridad referencial. Esta opción es posible realizarla cuando se genera el esquema de la base de datos. Si la base de datos tiene referencias de una tabla a sí misma, o referencias cíclicas, esta opción no se podrá utilizar.
- Posibilidad de generar instrucciones Insert, Update y Delete genéricas sobre una tabla, eligiendo las columnas que se necesitan. Esta opción se podrá realizar desde el menú de columnas de una tabla.
- Posibilidad de realizar consultas, total registro y borrar registros de la tabla desde el menú de tablas de la aplicación.

2. Correcciones

- Ttchartdll.dll. Al definir una leyenda con más de 11 ítems sólo se mostraban los 11 primeros.
Corregido
- Prnpag32.dll. Al previsualizar un listado que superaba el número máximo de controles se producía un error de ejecución.
Corregido
- Prnpag32. Al generar un listado con esta dll a partir de una determinada página, no mostraba los datos si en el bucle se utilizaba el método freePage de dicha dll.
Corregido
- Ttxmldll.dll. El método no representaba correctamente los acentos ni las “ñ”. Se ha implementado el método TTXmlGetEncodedNodeText.
Corregido
- Al hacer una exportación a Excel de listas con decimales con formato, éste no aparecía de manera correcta.
Corregido
- Al crear un control box sin label siendo el ancho de 2 píxeles, se mostraba mal el borde.
Corregido
- Al crear controles en tiempo de ejecución, en algunos casos retornaba identificadores ya existentes en la clase Form.
Corregido
- La función RemoveChart no funcionaba correctamente, pudiendo dejar bloqueada la aplicación.
Corregido
- Ctsql. Error de ejecución cuando se utilizaba una expresión con las cláusulas like o matches.
Corregido
- Ctsql. Problema al borrar tablas temporales que estaban activas en sentencias preparadas y no liberadas
Corregido
- Ctsql. Error al parsear una query con la instrucción outer join entre 3 tablas.
Corregido
- Ctsql. Si se lanzaba un procedimiento almacenado desde un trigger y la tabla sobre la que se aplicaba éste tenía una columna tipo Decimal o Money se producía un error.
Corregido
- Ctsql. Al utilizar dos tablas remotas en la misma instrucción select mostraba el siguiente mensaje: “Tabla no encontrada”.
Corregido
- Método FetchTuple de la clase SqlCursor. Si antes de ejecutar este método se ejecutaba una instrucción que devolvía un error en el mismo objeto SqlServer, el método FetchTuple no leía el siguiente registro y no devolvía ninguna tupla.
Corregido

- Cosrun. Consumo excesivo de memoria.
Corregido
- Cosrep. Si alguna columna del repositorio tenía definido el atributo default con valor " " (espacio en blanco), al modificar dicho atributo e indicar el mismo valor, eliminaba los espacios en blanco y cambiaba el valor a null.
Corregido
- Si un control tipo Grid con una tabla asociada tenía marcada la propiedad "Automatic Sort" y no se realizaba ninguna consulta, al pulsar en la cabecera ordenaba la lista completa.
Corregido
- ODBC. Se ha mejorado el manejo del driver para los tipos de datos TIME, DATE y DATETIME desde Access.
Corregido
- Entorno de Desarrollo. No se liberaba memoria en ciertas búsquedas de cadenas de caracteres en archivos de proyectos.
Corregido
- Entorno de Desarrollo. Al editar las propiedades de un control, si se cambiaba la coordenada "Y", no se reflejaba la modificación.
Corregido
- Debugger. Al refrescar la ventana de visualización de variables no se cerraba el árbol.
Corregido
- Método SubString. Si el segundo parámetro de este método era un número superior a 32766 se producía un error de protección general.
Corregido
- Método ReadWordInto. Si el fichero que se estaba procesando tenía una línea en blanco, la ejecución del programa no finalizaba correctamente.
Corregido
- Exportación a Excel. Si la última celda del listado era una variable no se mostraba el valor.
Corregido

NOTA. Se deshace la siguiente corrección:

Si se tenía asignado el número 3 a una tecla rápida y se pulsaba [Alt Gr] para escribir almohadilla (#), Cosmos ejecutaba la acción asociada a la tecla y no escribía la almohadilla, ya que dejaban de funcionar los aceleradores de label. Para escribir el carácter "#" se debía utilizar [Alt+3] en lugar de [Alt Gr+3]. Esta corrección se había implementado en la versión 4.4 de Cosmos.

Esta corrección provocaba que no funcionasen los aceleradores de teclado con [Alt+letra].

3. Anexo I. Realización de copias de seguridad en caliente

La utilidad diseñada para este fin se denomina **ttbackupdb**. Esta utilidad permite la realización “en caliente” de una copia de seguridad de la base de datos en un directorio, recibido como parámetro, y garantizando la total integridad de los datos.

Para llevar a cabo este proceso es requisito imprescindible que el CTSQL que se esté utilizando sea CTSQL Monitor y que el Monitor esté arrancado.

Al comenzar el proceso de copia de seguridad, la utilidad **ttbackupdb** tiene que comunicarse con los procesos CTSQL que accedan a la base de datos con el fin de notificarles que se va a llevar a cabo una copia de seguridad, garantizando de esta forma el buen funcionamiento de la herramienta.

Sintaxis:

```
./ttbackupdb -source <DDBB> -dest <DDBB>  
             -exclude <fichero de exclusión>  
./ttbackupdb -v  
./ttbackupdb -purge
```

Parámetros:

-source	Directorio donde se encuentra la base de datos sobre la que se va a efectuar la copia de seguridad.
-dest	Directorio donde se va a guardar la copia de seguridad.
-exclude	Permite indicar un fichero que contiene una lista de ficheros de tablas que se desean excluir de la copia de seguridad. Este fichero es un fichero de texto en el que cada línea debe contener el nombre de los ficheros DAT e IDX asociados a la tabla sin indicar su extensión (“.dat”, “.idx”).
-purge	Sirve para limpiar la zona de memoria compartida entre el comando ttbackupdb y las sesiones CTSQL. Este parámetro sólo debe utilizarse en caso de que el proceso haya fallado por si en la zona de memoria compartida hubiera quedado “marcado” como que todavía se estuviese ejecutando la copia seguridad.

3.1 Funcionamiento

El funcionamiento de la utilidad **ttbackupdb** es el siguiente:

Cuando se lanza el proceso de copia de seguridad, **ttbackupdb** comprueba que el Monitor está iniciado. Si no es así, muestra un mensaje de error y el programa se cancela. Si el Monitor está iniciado, **ttbackupdb** envía un mensaje a todas las instancias CTSQL que están conectadas a la base de datos objeto de la copia. Cada vez que una sesión CTSQL recibe esta notificación se registran en un fichero temporal todas las modificaciones que se realizan en la base de datos durante la copia de seguridad.

Cuando todas las sesiones están enteradas de que hay una copia de seguridad en curso, **ttbackupdb** comienza a copiar tablas.

Al finalizar la copia de los datos, **ttbackupdb** lo notifica a los CTSQL conectados a la base de datos, ordenándoles que dejen de trabajar hasta que termine de integrar los datos modificados en la base de datos durante el proceso de copia.

Una vez que todos los CTSQL han recibido esta notificación, **ttbackupdb** integra en la copia de seguridad los datos modificados en la base de datos origen durante el proceso de copia, y que han sido guardados por cada sesión CTSQL en el fichero temporal referido anteriormente.

Cuando **ttbackupdb** termina de integrar los datos, envía la orden de continuar a todos los procesos CTSQL, quita de la zona de memoria compartida el flag de copia de seguridad y borra los ficheros temporales.

Este comando sólo copiará tablas que se encuentren dentro del directorio de la base de datos o en un directorio hijo. Si hubiese tablas de esa base de datos cuyos ficheros DAT e IDX estuviesen en un directorio distinto al de la base de datos, dichas tablas no serían copiadas.

3.2 Variables de entorno

DBBACKUPTMPDIR

Esta variable de entorno indica el lugar donde se van a almacenar los ficheros temporales que se generen durante el proceso de copia. Su valor por defecto es /tmp.

Esta variable de entorno se define en el fichero de configuración del Monitor (monitor.ini).

El directorio debe tener permiso de lectura y escritura para las sesiones CTSQL. Si no es así, el proceso fallará en su última fase, no pudiendo garantizarse la consistencia de datos.

3.3 Ejemplo

```
export MONITOR=$TRANSDIR
./ttbackupdb/home/ctl/origen.dbs/home/ctl/destino.dbs
CTSQL-Monitor v1.1
Max sessions: 100.
Shared memory Connection - OK.
Monitor STARTED.
Hora de inicio de backup: Apr 02 19:16:52
Enviando señal de backup activo.
1ª fase. Backup activo.
Haciendo backup del directorio/home/ctl/origen.dbs en /home/ctl/destino.dbs
/home/ctl/origen.dbs/systables.dat --> /home/ctl/destino.dbs/systables.dat <-- OK
/home/ctl/origen.dbs/systables.idx --> /home/ctl/destino.dbs/systables.idx <-- OK
/home/ctl/origen.dbs/sysviews.dat --> /home/ctl/destino.dbs/sysviews.dat <-- OK
/home/ctl/origen.dbs/sysviews.idx --> /home/ctl/destino.dbs/sysviews.idx <-- OK
/home/ctl/origen.dbs/sysforeign.dat --> /home/ctl/destino.dbs/sysforeign.dat <-- OK
/home/ctl/origen.dbs/sysforeign.idx --> /home/ctl/destino.dbs/sysforeign.idx <-- OK
/home/ctl/origen.dbs/systabauth.dat --> /home/ctl/destino.dbs/systabauth.dat <-- OK
/home/ctl/origen.dbs/systabauth.idx --> /home/ctl/destino.dbs/systabauth.idx <-- OK
/home/ctl/origen.dbs/sysindexes.dat --> /home/ctl/destino.dbs/sysindexes.dat <-- OK
/home/ctl/origen.dbs/sysindexes.idx --> /home/ctl/destino.dbs/sysindexes.idx <-- OK
/home/ctl/origen.dbs/client_150.dat --> /home/ctl/destino.dbs/client_150.dat <-- OK
/home/ctl/origen.dbs/client_150.idx --> /home/ctl/destino.dbs/client_150.idx <-- OK
/home/ctl/origen.dbs/sysusers.dat --> /home/ctl/destino.dbs/sysusers.dat <-- OK
/home/ctl/origen.dbs/sysusers.idx --> /home/ctl/destino.dbs/sysusers.idx <-- OK
/home/ctl/origen.dbs/sysprocedu.dat --> /home/ctl/destino.dbs/sysprocedu.dat <-- OK
/home/ctl/origen.dbs/sysprocedu.idx --> /home/ctl/destino.dbs/sysprocedu.idx <-- OK
/home/ctl/origen.dbs/syscollati.dat --> /home/ctl/destino.dbs/syscollati.dat <-- OK
/home/ctl/origen.dbs/syscollati.idx --> /home/ctl/destino.dbs/syscollati.idx <-- OK
/home/ctl/origen.dbs/syssynonym.dat --> /home/ctl/destino.dbs/syssynonym.dat <-- OK
/home/ctl/origen.dbs/syssynonym.idx --> /home/ctl/destino.dbs/syssynonym.idx <-- OK
/home/ctl/origen.dbs/syscolumns.dat --> /home/ctl/destino.dbs/syscolumns.dat <-- OK
/home/ctl/origen.dbs/syscolumns.idx --> /home/ctl/destino.dbs/syscolumns.idx <-- OK
/home/ctl/origen.dbs/syscolattr.dat --> /home/ctl/destino.dbs/syscolattr.dat <-- OK
```

```
/home/ctl/origen.dbs/syscolattr.idx --> /home/ctl/destino.dbs/syscolattr.idx <-- OK
/home/ctl/origen.dbs/systrigger.dat --> /home/ctl/destino.dbs/systrigger.dat <-- OK
/home/ctl/origen.dbs/systrigger.idx --> /home/ctl/destino.dbs/systrigger.idx <-- OK
/home/ctl/origen.dbs/sysdepend.dat --> /home/ctl/destino.dbs/sysdepend.dat <-- OK
/home/ctl/origen.dbs/sysdepend.idx --> /home/ctl/destino.dbs/sysdepend.idx <-- OK
/home/ctl/origen.dbs/syscolauth.dat --> /home/ctl/destino.dbs/syscolauth.dat <-- OK
/home/ctl/origen.dbs/syscolauth.idx --> /home/ctl/destino.dbs/syscolauth.idx <-- OK
/home/ctl/origen.dbs/sysremote.dat --> /home/ctl/destino.dbs/sysremote.dat <-- OK
/home/ctl/origen.dbs/sysremote.idx --> /home/ctl/destino.dbs/sysremote.idx <-- OK
2ª fase. Copia hecha.
Enviando señal stop a los procesos CTSQL.
3ª fase. Bloqueo de acceso a la base de datos.
4ª fase. Copiando de los temporales creados por los SQL.
Esperando a la finalización de los procesos CTSQL.
Espera finalizada.
Cargando datos de log de backup de procesos CTSQL.
Datos cargados.
Actualizando de fichero.
5ª fase. Chequeo de tablas copiadas que ya no existen en la BASE DE DATOS.
6ª fase. Desbloqueo el acceso a la BASE DE DATOS y los SQLs activos.
Backup finalizado.
Hora de fin de backup: Apr 02 19:16:52
```

4. Anexo II. Cifrado (encriptación) de la base de datos

La versión 3.6 upgrade 10 del gestor de base de datos CTSQL incluye una nueva funcionalidad que permite tanto cifrar (*encrypt*) una base de datos existente como crear una base de datos cifrada (*encriptada*).

El comando que permite ejecutar esta nueva funcionalidad de cifrado es **ttcryptdb**.

El directorio de instalación de esta utilidad depende del sistema operativo empleado:

- En Linux/Unix: \$TRANSDIR/bin.
- En Windows: Directorio bin del directorio de instalación de MultiWay.

Sintaxis:

```
./ttcryptdb -source <source DDBB> -dest <dest DDBB>
[-password <dbatabase password>]
```

Parámetros:

-source	Path completo de la base de datos que se va a cifrar (incluido el directorio dbs). Este parámetro es obligatorio.
-dest	Path completo donde se creará la base de datos cifrada (incluido el directorio dbs). Este parámetro es obligatorio.
-password	Contraseña de protección de acceso a la base de datos. El valor de este parámetro debe ser alfanumérico y de 16 bytes como máximo. No se trata de una palabra de acceso a la base de datos cifrada, pero cuando se acceda a ésta será necesario indicar su valor en la variable de entorno DBCRYPTPWD . Aunque no es obligatorio, se recomienda utilizarla.

Ejemplos:

```
./ttcryptdb -source /home/ctl/desencrip.dbs
             -dest /home/ctl/encrip.dbs
./ttcryptdb -source /home/ctl/desencrip.dbs
             -dest /home/ctl/encrip.dbs -password mipasswd
```

El comando **ttcryptdb** sólo cifrará tablas que se encuentren dentro del directorio de la base de datos o en un directorio hijo. Si hubiese tablas de esa base de datos cuyos ficheros DAT e IDX estuviesen en un directorio distinto al de la base de datos, dichas tablas no serían cifradas.

4.1 Acceso a bases de datos cifradas

En el motor CTSQL se ha implementado una nueva funcionalidad que permite crear y acceder a bases de datos cifradas.

El motor CTSQL es totalmente compatible con las bases de datos existentes que no están cifradas, además de permitir la creación y el acceso a base de datos no cifradas.

Por su parte, el Monitor es incompatible con la versión 1.0, ya que la estructura de la memoria compartida es diferente.

La versión 3.6 upgrade 10 del gestor CTSQL necesita la versión 1.1 de Monitor.

4.2 Variables de entorno

CRYPTDB

Esta variable de entorno indica al gestor de base de datos si la base de datos a la que se va a conectar está cifrada o no.

Se define en el fichero de configuración del CTSQL (ctsql.ini) si la conexión es cliente-servidor. Si la conexión es local se define en la sección [Environment] del fichero de configuración de Cosmos (cosmos.ini).

Sus posibles valores son: YES y NO.

Si no se define el gestor de base de datos se asume que la base de datos no está cifrada.

Si se asigna el valor YES y la base de datos no está cifrada, CTSQL mostrará el siguiente mensaje de error: "Base de datos no encontrada o sin permiso del sistema", y lo mismo sucederá si el valor de la variable es NO con la base de datos cifrada.

DBCRYPTPWD

Esta variable de entorno indica al gestor de la base de datos cuál es la contraseña de protección de acceso a la base de datos cifrada a la que se va a acceder.

Si la base de datos ha sido creada con el comando **ttcryptdb**, el valor de la variable de entorno DBCRYPTPWD debe coincidir con el indicado en el parámetro "-password" de dicho comando. Esta variable de entorno no se deberá definir si no se ha indicado este parámetro en el proceso de creación de la base de datos.

Se debe definir en el fichero de configuración del gestor de base de datos (ctsql.ini), aunque también podrá definirse en el fichero de configuración del cliente dentro de la sección donde se define la conexión.

Si no se define esta variable de entorno, CTSQL intentará acceder a la base de datos cifrada sin palabra de acceso (en caso de que se hubiese definido la variable de entorno CRYPTDB=YES).

4.3 Creación de una base de datos cifrada

La creación de una base de datos cifrada se podrá realizar desde las siguientes utilidades:

- Sql-Interactivo, tanto en versiones para Unix/Linux como Windows.
- Repositorio de Cosmos.
- Programa desarrollado en Cosmos.
- MultiBase para Windows sólo podrá crear bases de datos cifradas en conexiones cliente-servidor.

Al crear una base de datos cifrada, es recomendable definir la variable de entorno [DBCRYPTPWD](#) en el servidor (en el fichero de configuración ctsql.ini). Por razones de seguridad, no es recomendable definirla en el cliente.

En caso de que se indique en el cliente, lo más seguro es definirla desde dentro de la aplicación.

4.4 Reparación y chequeo de bases de datos cifradas

Los comandos encargados de los procesos de reparación y chequeo de bases de datos cifradas son respectivamente **trepidx** y **tchkidx**.

En la versión para Windows se han añadido dos parámetros nuevos: **-cryptdb** y **-dbcryptpwd**.

4.4.1 Comando trepidx

Sintaxis:

```
./trepidx -v <database> <table>|<-all> [-ndl|-dl] [-y] [-cryptdb]
[-dbcryptpwd <cryptpassword>]
```

Parámetros:

-v	Muestra la versión del comando y su "upgrade".
database	Nombre de la base de datos, sin extensión ".dbs", a la que pertenece(n) la(s) tabla(s) a reparar. Esta base de datos tiene que encontrarse en el directorio en curso o bien en cualquiera de los directorios indicados en la variable de entorno DBPATH. En el caso de no tener export, la variable de entorno DBPATH deberá indicar la ruta completa de la base de datos con la extensión ".dbs".
table	Nombre de la tabla de la que se desea comprobar la integridad de sus índices con respecto a los datos. Esta tabla debe pertenecer a la base de datos indicada en el parámetro anterior.
-all	Indica que se repararán todas las tablas de la base de datos especificada en el parámetro anterior.
-ndl	Indica que el/los índice/s de la/s tabla/s no tiene/n límite de duplicados.
-dl	Indica que el/los índice/s de la/s tabla/s sí tiene/n límite de duplicados.
-y	No pregunta nada al operador.
-cryptdb	Indica si la base de datos que se va a reparar está cifrada o no.
-dbcryptpwd <cryptpassword>	Contraseña de cifrado de la base de datos. Si se incluye este parámetro no es necesario indicar "-cryptdb", ya que asume que la base de datos está cifrada.

En la versión para Unix/Linux, la forma de ejecución no ha variado, simplemente se tendrán que añadir a la exportación de variables de entorno que se hacía en versiones anteriores las dos nuevas que se han creado para poder realizar el proceso de cifrado: CRYPTDB y DBCRYPTPWD.

Sintaxis:

```
./trepidx <database> <table>|<-all> [-ndl|-dl] [-y]
```

Ejemplo:

```
export CRYPTDB=YES
export DBCRYPTPWD=mipasswd
[ctl@rhel150 bin]$ ./trepidx/home/bbdd/encrip.dbs clientes
```

En este ejemplo, el comando **trepidx** accede a la base de datos “encrip.dbs”, que según las variables de entorno CRYPTDB y DBCRYPTPWD es una base de datos cifrada y con contraseña “mipasswd”, y repara la tabla “clientes”.

4.4.2 Comando tchkidx

Sintaxis:

```
tchkidx -v <database> [<table>|-all] [-ndl|-dl] [-y] [-cryptdb]
[-dbcryptpwd <cryptpassword>]
```

Parámetros:

-v	Muestra la versión del comando y su “upgrade”.
database	Nombre de la base de datos, sin extensión “.dbs”, a la que pertenece(n) la(s) tabla(s) a chequear. Esta base de datos tiene que encontrarse en el directorio en curso o bien en cualquiera de los directorios indicados en la variable de entorno DBPATH. En el caso de no tener export, la variable de entorno DBPATH deberá indicar la ruta completa de la base de datos con la extensión “.dbs”.
table	Nombre de la tabla de la que se desea comprobar la integridad de sus índices con respecto a los datos. Esta tabla debe pertenecer a la base de datos indicada en el parámetro anterior.
-all	Indica que se chequearán todas las tablas de la base de datos especificada en el parámetro anterior.
-y	No pregunta nada al usuario.
-ndl	Indica que el/los índice/s de la/s tabla/s no tiene/n límite de duplicados.
-dl	Indica que el/los índice/s de la/s tabla/s sí tiene/n límite de duplicados.
-cryptdb	Indica si la base de datos que se va a chequear está cifrada o no.
-dbcryptpwd <cryptpassword>	Indica la contraseña de cifrado de la base de datos. Si se incluye este parámetro no es necesario indicar “-cryptdb”, ya que asume que la base de datos está cifrada.

En la versión para Unix/Linux, en lugar de indicar los dos últimos parámetros habrá que exportar la variables de entorno CRYPTDB y DBCRYPTPWD.

Sintaxis:

```
tchkidx -v <database> [<table>|-all] [-ndl|-dl]
```

Los comandos **trepidx** y **tchkidx** son totalmente compatibles con bases de datos existentes que no estén cifradas.