

Realización de copias de seguridad en caliente

BASE100

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

Índice

1. INTRODUCCIÓN	3
2. FUNCIONAMIENTO	4
2.1 VARIABLES DE ENTORNO	4
2.2 EJEMPLO	5

1. Introducción

En este documento se describe el procedimiento y la utilidad que han de emplearse para poder realizar una copia 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 <DDBE> -dest <DDBE>
               -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.

2. 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.

2.1 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.

2.2 Ejemplo

```
export MONITOR=$TRANSDIR
./ttbackupdb/home/ctl/origen.dbs/home/ctl/destino.dbs
CTSOL-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