

microsoft.public.sqlserver.server: Re: Drop all the connections to the DataBase

Usage: Just run this complete script in the master database to create this stored procedure. As far as syntax is

concerned, this procedure works very similar to the system stored procedure `sp_dboption`. It has an additional parameter `@wait`, which can be used, to wait for a specified number of seconds, before killing the connections.

The settable database option names need to be specified in full. For example, the option name 'single' is considered invalid and 'single user' is considered valid.

To bring pubs database into single user mode:

```
EXEC sp_dboption2 'pubs', 'single user', 'true'
```

To bring pubs database into single user mode. Wait for 30 seconds, for current connections to leave and start killing the connections after 30 seconds:

```
EXEC sp_dboption2 'pubs', 'single user', 'true', 30
```

To bring pubs database into read/write mode:

```
EXEC sp_dboption2 'pubs', 'read only', 'false'
```

To bring pubs database into read/write mode. Wait for 30 seconds, for current connections to leave and start killing the connections after 30 seconds:

```
EXEC sp_dboption2 'pubs', 'read only', 'false', 30
```

```
*****  
*****/
```

```
DECLARE @dbid int, @spid int, @execstr varchar(15), @waittime varchar(15),  
@final_chk int
```

—Only the following options require that, no other connections should access the database

```
IF (LOWER(@optname) IN ('offline', 'read only', 'single user')) AND  
(LOWER(@optvalue) IN('true', 'false'))
```

```
BEGIN
```

—Determining whether to wait, before killing the existing connections

```
IF @wait > 0
```

```
BEGIN
```

```
SET @waittime = (SELECT CONVERT(varchar, DATEADD(s, @wait, GETDATE()),  
14))
```

```
WAITFOR TIME @waittime —Wait the specified number of seconds
```

```
END
```

```
SET @dbid = DB_ID(@dbname) —Getting the database_id for the specified  
database
```

microsoft.public.sqlserver.server: Re: Drop all the connections to the DataBase

```
--Get the lowest spid
TryAgain:
SET @spid = (SELECT MIN(spид) FROM master..sysprocesses WHERE dbid =
@dbid)

WHILE @spid IS NOT NULL
BEGIN
IF @spid <> @@SPID --To avoid the KILL attempt on own connection
BEGIN
SET @execstr = 'KILL ' + LTRIM(STR(@spid))
EXEC(@execstr) --Killing the connection
END
--Get the spid higher than the last spid
SET @spid = (SELECT MIN(spид) FROM master..sysprocesses WHERE dbid =
@dbid AND spид > @spid)
END

END

SET @final_chk = (SELECT COUNT(spид) FROM master..sysprocesses WHERE dbid =
@dbid)
IF (@final_chk = 0) OR (@final_chk = 1 AND DB_NAME() = @dbname)
BEGIN
EXEC sp_dboption @dbname, @optname, @optvalue --Calling sp_dboption to
complete the job
END
ELSE
BEGIN
GOTO TryAgain --New connections popped up, or killed connections aren't
cleaned up yet, so try killing them again
END
END
```

"Ram" <anonymous@discussions.microsoft.com> wrote in message
news:F0E88B1A-A105-4E24-9BEE-CEB5FC97F532@microsoft.com...

> Hi,

> I am using the following code in a batch file to take the back up of a
database. but some times it fails with the message

>

> *"Cannot access the database because it is being used by another process."*

Is there any way i can force all the connections

> to the database to be dropped using code. Any help will be greatly
appreciated.

>

> *isql -b -S localhost -E -U xyzuser -P xyz -Q "sp_detach_db 'JMS', 'true'"*

>

> *copy C:\JJMSdb\JMS_Data.MDF C:\JJMSdb\JMS_Data_2004-03-29_14-54-5933.MDF*

>

> *copy C:\JJMSdb\JMS_Log.LDF C:\JJMSdb\JMS_Log_2004-03-29_14-54-5933.LDF*

>

> *isql -b -S localhost -E -U xyzuser -P xyzpwd -Q "sp_attach_db @dbname =*

Re: Drop all the connections to the DataBase

microsoft.public.sqlserver.server: Re: Drop all the connections to the DataBase

```
'JMS', @filename1 = 'C:\JJMSdb\JMS_Data.MDF', @filename2 =  
'C:\JJMSdb\JMS_Log.LDF'"
```

>

> *Thanks,*

> *Ram*

>

>

>

>