DBCC Secrets

Shhh

Mark Kusma
Manager, Tactical Support
Sybase CS&S
August 9, 2007

TECHWAVE 2007
SYBASE USER TRAINING & SOLUTIONS CONFERENCE

August 6–10 | Mandalay Bay | Las Vegas, Nevada

DBCC Secrets

What this session is

- We will look at several DBCC commands that are primarily undocumented
- We avoid commands that will intentionally crash systems, or cause major problems
- We look at some commands that must be used with caution

What this session is not

- This is not an exhaustive list of DBCC commands
- There are close to 300 DBCC commands

DBCC Secrets

We’ll cover the syntax, version info, usage, and potential issues of several DBCC commands. We’ll also look at ‘when’ and ‘why’ to use them.

These commands should be used with dbcc traceon(3604)

- Not shown on subsequent slides

dbcc cacheremove

Syntax

- dbcc cacheremove(dbid | dbname, objid | objname)

Version

- Added prior to ASE 11.9

Usage

- Used to remove a descriptor (DES) from cache (“open objects”)
- If object is a table
  - the IDES (“open indexes”) and PDES (“open partitions”) is also removed
  - Writes dirty buffers to disk
  - Does NOT remove pages from data cache
  - In-memory statistics flushed to systabstats and sysstatistics
- Places the DES back on the FREE chain
- Cannot be used with system tables
- Must have sa_role
  - Cannot be granted

dbcc cacheremove

Potential Issues

- Object may be in use and cannot be removed from cache

When and Why

- Use when system table info has been manually updated for this object
  - Example, changing uid in sysobjects to change owner
- Use when scavenging is taking place and you can’t increase the “number of <item>” config param due to memory constraints
  - Either remove rarely used objects, or remove many and let ASE recache the active objects
- Use to remove single statements from statement cache
  - See new option on dbcc purgesqlcache – 15.0.2 What’s New

dbcc cacheremove

```
sp_monitorconfig 'open objects‘
go
```

```
Usage information at date and time: Jul 11 2007 10:37AM.
 Name                      Num_free    Num_active  Pct_act Max_Used
         Reuse_cnt
 ------------------------- ----------- ----------- ------- -----------
         -----------
 number of open objects            447          53  10.60           53
         0
```

```
dbcc cacheremove(test,p_abc)
go
sp_monitorconfig 'open objects‘
go
```

```
Usage information at date and time: Jul 11 2007 10:38AM.
 Name                      Num_free    Num_active  Pct_act Max_Used
         Reuse_cnt
 ------------------------- ----------- ----------- ------- -----------
         -----------
 number of open objects            448          52  10.40           53
                   0
```

dbcc checkindex

Syntax

- dbcc checkindex(tbl_name | tabid, indid [,bottom_up | NULL [,ptn_name | ptnid]])

Version

- Added in ASE 12.5.3 under CR 338495

Usage

- Similar to checktable, but only executes for one index
- Option bottom_up only for DOL tables
- Partition option applies to 15.0, and higher
- Must have sa_role or be the table owner
  - Can be granted, however, granted at database level (or all databases)

dbcc checkindex

Potential Issues

- Index corruption may be found, requiring action

When and Why

- Use when checktable will take too long to process all indexes
- Use ‘bottom_up’ to verify each row of a DOL table has a corresponding index leaf row
  - Catch 644-type errors
  - If error found, rebuild index and check others

dbcc checkindex

```
dbcc checkindex(abc_dol,2,bottom_up)
go
Checking table 'abc_dol' (object ID 784002793): Logical page size is 2048 bytes.
Table has 99 data rows.
Index has 99 leaf rids.
The total number of data pages in this table is 99.
```

dbcc dbcacheremove

Syntax

- dbcc dbcacheremove(dbid | dbname)

Version

- Added in ASE 12.5.3 and 12.0.0.8 ESD2 under CR 334132
- Fix in 12.5.4 ESD3 and 15.0.2 for infected with 11 CR 438354

Usage

- Used to remove a database descriptor (DBTABLE) from cache (“open databases”)
- Places the DBTABLE back on the FREE chain
- Cannot be used on system databases or user tempdbs
- Most commonly addresses 3101 errors trying to load database
- Has been on “request list” for long time
  - Preference was to fix the cause of wrong dbt_keep value
- Must have sa_role
  - Cannot be granted

dbcc dbcacheremove

Potential Issues

- Database may really be in use and cannot be removed from cache

When and Why

- Typically for Msg 3101 (database in use) during load db
- dbt_keep may be off due to some previous error or failed task cleanup
  - Previously had to cycle ASE to clear the ‘keep count’
- Several lines of output indicate usage info and success/failure
- Usage and result is also logged in errorlog

dbcc dbcacheremove

```
dbcc dbcacheremove(test)
go
Attempt to uncache the database 'test' with dbid 4.
The descriptors keep count is 1 and the system tasks keep count is 0.
The descriptors hot count is 0 and the detached keep count is 0.
The per process keep count for spid 12 is 1.
The database is set to current by spid 12.
The database 'test' could not be uncached.
```

```
dbcc dbcacheremove(pubs2)
go
Attempt to uncache the database 'pubs2' with dbid 5.
The descriptors keep count is 0 and the system tasks keep count is 0.
The descriptors hot count is 0 and the detached keep count is 0.
The database 'pubs2' has been uncached.
```

dbcc delete_dol_datapage

Syntax

- dbcc delete_dol_datapage(dbid, objid, page, [noblock])

Version

- Added in ASE 12.5.3 ESD3 and 15.0 ESD2 under CR 383241

Usage

- Used to deallocate a corrupt datapage from a DOL table
- Requires an EX_TAB lock on the table, unless ‘noblock’ is used
- Must have sa_role
  - Cannot be granted
- Cannot be used with Archive Databases at this time
- Use only table scans after this, such as bcp to recover good data

dbcc delete_dol_datapage

Potential Issues

- Indexes pointing to this page may cause errors and should be dropped and recreated
- Data WILL be lost

When and Why

- 6xx errors on a page and trying to bcp data out fails
- As a last resort, you can deallocate the page so that bcp can run

Example Setup

```
create table abc_dol (a int, b char(1500)) lock datarows
go
insert abc_dol values(1,'ABC')
go 100
create index x_abc_dol on abc_dol (a)
go
```

dbcc delete_dol_datapage

```
declare @vers int, @page int, @dbid int, @obj int
select @dbid = db_id(), @obj = object_id('abc_dol'), @vers = @@version_as_integer
if @vers < 15000
  select @page = first+1 from sysindexes where id = @obj and indid = 0
else
  select @page = firstpage+1 from syspartitions where id = @obj and indid = 0
dbcc delete_dol_datapage(@dbid,@obj,@page)
Forced deallocation of page on dbid = 4, objid = 784002793, pageid = 890, object
will be consistent for table scans only.
Some info from the page being deallocated in dbid 4: pgno = 890, objid =
784002793, ptnid = 784002793, nextpg = 0, prevpg = 0, timestamp = 0000 00001cad.
select count(*) as "Index Count" from abc_dol
select count(*) as "Table Count" from abc_dol (index 0)
Index Count
 -----------
         100
 Table Count
 -----------
          99
```

dbcc lock_release

Syntax

- dbcc lock_release(spid, level, type, dbid, objid, [page [,row]] [,’force’])

Version

- Added in ASE 12.5.3 ESD2 and 15.0 ESD2 under CR 374681

Usage

- spid must have been badly terminated (P3_PSS_SICK or failed termination)
- “level” is the lock level: “table” | “page” | “row”
- “type” is the lock type and depends on the lock level
  - ex_tab, sh_tab, ex_int, sh_int
  - ex_page, up_page, sh_page - “page” required
  - ex_row, up_row, sh_row - “page, row” required
- ‘force’ must be used if the target spid has an active transaction – Caution!!
- Must have sa_role
  - Cannot be granted

dbcc lock_release

Potential Issues

- Forcing the release of a transactional lock may cause inconsistency of data
- An open transaction may still be a problem for truncating the log
  - (See dbcc dbreboot at end of presentation)

When and Why

- Use to clean up phantom locks not released by a badly terminated spid
- spid may be holding a lock that prevents accessing or dropping object

dbcc lock_release

```
dbcc lock_release(12,'table','ex_int',4,880003135)
go
Spid 12 was not badly terminated. Cannot release the given lock.
```

```
dbcc lock_release(12,'table','ex_int',4,880003135)
go
The transactional 'exclusive intent' lock for spid 12, dbid 4, objid 880003135,
pageno 0, rowno 0 cannot be released. Use the 'force' option.
```

```
dbcc lock_release(12,'table','ex_int',4,880003135,force)
go
The 'exclusive intent' lock at 0x0000000200e3ddb0 for spid 12, dbid 4, objid
880003135, pageno 0, rowno 0 was forcibly released.
```

dbcc logprint/printolog/traceprint

Syntax

- dbcc logprint(“multiple lines”,”can be printed”)
- dbcc printolog(“only one string allowed”)
- dbcc traceprint(“destination depends”,”on traceflags 3604,3605,3623”)

Version

- Added prior to 11.9

Usage

- logprint requires sybase_ts_role
- printolog requires sa_role or ha_role
- traceprint does not require specific role
  - Does not include the errorlog prefix info:
    - 00:00000:00021:2007/06/24 19:37:40.83 server

dbcc logprint/printolog/traceprint

Potential Issues

- May affect errorlog monitoring applications

When and Why

- Use in a script to output progress information or debugging info
- Can be used to trigger errorlog monitors to page someone, or take action
- Use to record when certain procs are run, such as threshold procs, etc.

dbcc logprint/printolog/traceprint

```
dbcc logprint('This is a test', 'for the TechWave', 'Presentation')
go
```

Errorlog output:

```
00:00000:00011:2007/07/16 22:31:26.72 server  This is a test
00:00000:00011:2007/07/16 22:31:26.72 server  for the TechWave
00:00000:00011:2007/07/16 22:31:26.72 server  Presentation
```

```
dbcc printolog('This is for the TechWave Presentation')
go
```

Errorlog output:

```
00:00000:00011:2007/07/16 22:34:43.51 server  This is for the TechWave Presentation
```

dbcc logprint/printolog/traceprint

```
dbcc traceoff(3605)
dbcc traceprint('This is a test', 'for the TechWave', 'Presentation')
go
```

ASE Console output (does not go in errorlog):

```
This is a test
for the TechWave
Presentation
```

```
dbcc traceon(3605)
dbcc traceprint('This is a test', 'for the TechWave', 'Presentation')
go
```

Errorlog output:

```
This is a test
for the TechWave
Presentation
```

dbcc logprint/printolog/traceprint

```
dbcc traceoff(3605) -- must be off, otherwise 3623 ignored
dbcc traceon(3623)
dbcc traceprint('This is a test', 'for the TechWave', 'Presentation')
go
```

Errorlog output:

```
00:00000:00011:2007/07/16 22:42:14.06 server  This is a test
00:00000:00011:2007/07/16 22:42:14.06 server  for the TechWave
00:00000:00011:2007/07/16 22:42:14.06 server  Presentation
```

```
dbcc traceon(3604,3623) -- also sends to isql session, but no prefix info
dbcc traceprint('This is a test', 'for the TechWave', 'Presentation')
go
```

Errorlog output:

```
00:00000:00011:2007/07/16 22:42:32.20 server  This is a test
00:00000:00011:2007/07/16 22:42:32.20 server  for the TechWave
00:00000:00011:2007/07/16 22:42:32.20 server  Presentation
```

dbcc memdump

Syntax

- dbcc memdump([optional string])

Version

- Added in ASE 12.5.1 under CR 323061

Usage

- Used to capture a shared memory dump
- Required setup

  sp_shmdumpconfig ‘add’,’dbcc’,null,10,’<dir>’
  sp_configure ‘dump on conditions’,1
  see also sp_configure ‘number of dump threads’

  <some activity>
- Execution
  dbcc memdump(‘Test Point 1’)
- Must have sa_role to execute
  - Cannot be granted

dbcc memdump

Potential Issues

- Engines are halted by default while the memdump is captured – this is best

When and Why

- Can be used when no other way to trigger memdump (no error or message)
- Tech Support may request memdumps when diagnosing issues related to performance, proc cache fragmentation, or other
- Memdump analysis requires access to source code
- The optional text is recorded in the errorlog and can be used to identify the test case, or test progress

dbcc memdump

```
dbcc memdump(‘Start test’)
exec myproc1
exec myproc2
dbcc memdump(‘After proc2’)
exec myproc3
exec myproc4
dbcc memdump(‘End of test’)
go
```

dbcc proc_cache

Syntax

- dbcc proc_cache(free_unused | flush_elc)

Version

- Added in 12.5.4 ESD5 and 15.0.2 under CR 446389

Usage

- free_unused frees unused procedures from cache
- flush_elc flushes pages from Engine Local Cache to global pool
- Should be used with care and when system is not heavily used
- Must have sa_role or sybase_ts_role to execute
  - Cannot be granted

dbcc proc_cache

Potential Issues

- Procs flushed from cache may need to be read from disk and recompiled
- Removing pages from ELC may increase spinlock contention on proc cache global pool

When and Why

- Use to clean and defragment proc cache without recycling ASE
- Try this if you receive a 701 error
- Pages in ELC may not be used due to multi-page allocs
- free_unused can be used when changing optimization goals/settings

dbcc proc_cache

```
dbcc proc_cache(free_unused)
go
Procedures in cache before free unused:   15
Procedures in cache after free unused:     0
```

```
dbcc proc_cache(flush_elc)
go
erlocal_count before flush: eng0:   1571
erlocal_count after  flush: eng0:      0
```

dbcc proc_cacherm

Syntax

- dbcc proc_cacherm(type, dbname, objname)

Version

- Added prior to 11.9

Usage

- type is V,P,T,R,D,C,F, or S (must be uppercase)
- corresponds to View, Proc, Trigger, Rule, Default, Cursor, SQLJ Function, SQL function
- Must have sybase_ts_role to execute
  - Cannot be granted

dbcc proc_cacherm

Potential Issues

- Objects removed from cache may need to be read from disk and recompiled

When and Why

- Remove objects from procedure cache
- May help with some 3702 errors
  - 3702: Cannot drop the <objtype> <objname> because it is currently in use
- Use monCachedProcedures to get a list of objects currently in proc cache

dbcc proc_cacherm

```
select
  case ObjectType
     when 'stored procedure'
     then 'dbcc proc_cacherm(P,' + DBName + ',' + ObjectName + ')'
     when 'trigger procedure'
     then 'dbcc proc_cacherm(T,' + DBName + ',' + ObjectName + ')'
     when 'view'
     then 'dbcc proc_cacherm(V,' + DBName + ',' + ObjectName + ')'
     when 'default value spec'
     then 'dbcc proc_cacherm(D,' + DBName + ',' + ObjectName + ')'
     when 'rule'
     then 'dbcc proc_cacherm(R,' + DBName + ',' + ObjectName + ')'
  end
from master..monCachedProcedures
go
```

dbcc showrecovery

Syntax

- dbcc showrecovery[(dbid | dbname)]

Version

- Added in 12.5.3 under CR 353375, enhanced in 12.5.4 ESD7 under CR 370410

Usage

- If dbid or dbname is specified, the database must be in recovery
  - boot time, LOAD DATABASE, LOAD TRAN, ONLINE DATABASE, or MOUNT DATABASE
- If omitted, information is shown for all databases in recovery
- Must have sa_role to execute
  - Cannot be granted

dbcc showrecovery

Potential Issues

- None

When and Why

- Database recovery occasionally takes a long time
  - One customer scenario was over 30 hours
- Use this command to display information about the recovery process and help determine why it is running long

dbcc showrecovery

```
dbcc showrecovery
go
No databases are being recovered.
```

```
dbcc showrecovery
go
 RECOVERY STATISTICS FOR DATABASE 6

 General:
   Spid: 12
   Command: LOAD DATABASE
   Type: LOAD DATABASE
   Pass: Initialize
 …    (lots of output)
```

dbcc serverlimits

Syntax

- dbcc serverlimits[(‘all’)]

Version

- Added in 12.5 under CR 214917, enhanced in 15.0 under CR 360050 for long idents

Usage

- Some people familiar with the command, but not the ‘all’ option
- If ‘all’ specified, additional output is shown with more internal information
- Anyone can execute, however, output goes to stdout where ASE was started
  - Use traceflag 3604 to display on screen – requires sa_role

dbcc serverlimits

Potential Issues

- None

When and Why

- Identify various server limits, such as
  - max columns in a table
  - max indexes on a table
  - max length of table, column, proc names, etc in 15.x
- Limits based on 2K, 4K, 8K, 16K pagesize, such as
  - max size of row
  - max rows per page

dbcc serverlimits

Use isql –w120 option

dbcc serverlimits(‘all’)

go

…

```
  Max number of variable length columns in a system table
   (MAX_VAR_SYSTABS)                  : 81
  Max number of times a statement in stored procedure can be recompiled
   (MAX_NUMBER_OF_RECOMPILATIONS): 10
  Max length of a Sybase version string
   (SYBASE_MAX_VERSION_LEN)       : 100
```

…

dbcc tune(des_bind)

Syntax

- dbcc tune(des_bind, dbid, objid | objname)

Version

- Added in 11.9.1 under CR 139739

Usage

- Must specify dbid, not dbname
- Marks the DES as hot, meaning it won’t be scavenged and is guaranteed to be in memory
- Does not survive ASE recycle
- Must have sa_role to execute
  - Can be granted – access to all “tune” options

dbcc tune(des_bind)

Potential Issues

- A DES is used and cannot be scavenged
- Object cannot be dropped until unbound

When and Why

- Relieve pressure on the DES spinlock
  - Check sp_sysmon “Object Manager Spinlock Contention”
- Code can be skipped since we are guaranteed to have an active DES for this object
- object can be table, proc, default, rule, view, trigger

dbcc tune(des_bind)

```
dbcc tune(des_bind,4,abc_dol)
go

dbcc cacheremove(test,abc_dol)
go
Some one else is using this object. DES not Removed

drop table abc_dol
go
Msg 3702, Level 16, State 1:
Server 'acid1502', Line 1:
Cannot drop the table 'abc_dol' because it is currently in use.
```

dbcc tune(hotdes)

Syntax

- dbcc tune(hotdes, dbid)

Version

- Added in 11.9.1 under CR 139739

Usage

- Must specify dbid, not dbname
- Must have sa_role to execute
  - Can be granted – access to all “tune” options

dbcc tune(hotdes)

Potential Issues

- None

When and Why

- Displays DES information for objects currently marked as “hot” within the specified database

dbcc tune(hotdes)

```
dbcc tune(hotdes,4)
go

DES chain (bind state) for all objects in dbid 4:
Current database (4) test:

DES at 0x20158a8e0
ddbid=4
        OBJECT structure at: 0x20158a90c
        objname=abc_dol dol_rowno=14 dol_vattcnt=7 dol_status: 0x04:
   (0x0004(DOL_ROW_POSSUNCMT))

        dobjstat.objid=976003477  dobjstat.objuserstat=0x0
        dobjstat.objsysstat=0x43 3 (O_USER) (0x0040 (O_LOGGED))
        dobjstat.objsysstat2=0x18200 (0x00010000 (O2_ELEVEN90_OBJECT),
   0x00008000
(O2_LOCK_DATAROWS), 0x00000200 (O2_NOVARCOL))
…
```

dbcc tune(des_unbind)

Syntax

- dbcc tune(des_unbind, dbid, objid | objname)

Version

- Added in 11.9.1 under CR 139739

Usage

- Must specify dbid, not dbname
- Removes the ‘hot’ status from the DES and allows the descriptor to be freed (see cacheremove) / reused or the object to be dropped
- Must have sa_role to execute
  - Can be granted – access to all “tune” options

dbcc tune(des_unbind)

Potential Issues

- DES for this object can be reused

When and Why

- Remove “hot” status from an object previously marked hot with des_bind
- Must be done before dropping object

dbcc tune(des_unbind)

```
dbcc tune(des_unbind,4,abc_dol)
go

dbcc cacheremove(test,abc_dol)
go

drop table abc_dol
go
```

dbcc istraceon

Syntax

- dbcc istraceon(<traceflag>)

Version

- Added prior to 11.9

Usage

- Used to determine if a traceflag is on
- Can be done in batch code
- New built-in, switchprop, added in 12.5.4 ESD4 and 15.0.2 (0 is off)
- Anyone can use

dbcc istraceon

Potential Issues

- @@error variable is set (0 = on, -1 = off)

When and Why

- Use in a script to see if a particular traceflag is set

dbcc istraceon

```
dbcc traceoff(3604)
go

dbcc istraceon(3604)
select case @@error
  when 0 then 'traceflag is on‘
  when -1 then 'traceflag is off‘
end
go
```

```
 ----------------
 traceflag is off
```

dbcc dbreboot

Syntax

- dbcc dbreboot(<cmd>,<db1> [,dbn])

Version

- Added in 12.5.4 ESD1 and 15.0.2 under CR 368529

Usage

- <cmd> can be report, reboot, reboot_norecovery, shutdown, shutdown_load, restart, restart_norecovery
- Can specify up to 16 databases

dbcc dbreboot

Potential Issues

- Running tasks will be terminated

When and Why

- Database may be incorrectly marked as in use
- spid received 3475 error, or 12583
  3475: There is no space available in SYSLOGS to log a record for which space has been reserved.
  12583: Unable to log a record due to internal error %d.
  - Both errors will retry at 1 minute intervals
  - If need to add more space, use “reboot_norecovery” option
- Task failed during termination processing – ‘phantom spids’

dbcc dbreboot

Options

- report – Shows a report on the specified database(s).
- reboot – Database is shutdown and restarted with recovery.
- reboot_norecovery – Database is shutdown and restarted and left in a “not recovered” state. Can be used to add more space.
- shutdown – Database is shutdown and left in an unusable state.
- shutdown_load – Database is shutdown and left in a state that allows LOAD DATABASE to be done.
- restart – Restart and recover a database that has been shutdown.
- restart_norecovery – Restart database and left in a “not recovered” state. Good when there are problems with recovery.

dbcc dbreboot

```
drop database tb2
go
Msg 3702, Level 16, State 4:
Server 'acid1502', Line 1:
Cannot drop the database 'tb2' because it is currently in use.
dbcc dbreboot(report,tb2)
go
---------- Active Processes and Transactions in Database 'tb2' ----------

  Spid Program     Transaction     Status          CPU     I/O    Blk  Error

---------- Operation on Database 'tb2' Completed Successfully ----------
```

dbcc dbreboot

```
dbcc dbreboot(reboot,tb2)
go
---------- Shutting Down Database 'tb2' ----------
---------- Re-starting Database 'tb2' With Recovery ----------
Recovering database 'tb2'.
Started estimating recovery log boundaries for database 'tb2'.
Database 'tb2', checkpoint=(1051, 9), first=(1051, 8), last=(1051, 9).
Completed estimating recovery log boundaries for database 'tb2'.
Started ANALYSIS pass for database 'tb2'.
Completed ANALYSIS pass for database 'tb2'.
Started REDO pass for database 'tb2'. The total number of log records to process is 2.
Completed REDO pass for database 'tb2'.
Recovery of database 'tb2' will undo incomplete nested top actions.
Started UNDO pass for database 'tb2'. The total number of log records to process is 2.
Undo pass of recovery has processed 1 incomplete transactions.
Completed UNDO pass for database 'tb2'.
Started filling free space info for database 'tb2'.
Completed filling free space info for database 'tb2'.
Started cleaning up the default data cache for database 'tb2'.
Completed cleaning up the default data cache for database 'tb2'.
Recovery complete.
Database 'tb2' is now online.
---------- Operation on Database 'tb2' Completed Successfully ----------
drop database tb2
go
```

QUESTIONS?