

ASE Sybase Notes

Handy tips for the busy DBA

Last updated: 05/04/2018

DBXperts Ltd

Garrett Devine

www.dbxperts.co.uk



Secure Remote Database Administration

Table of Contents

Document Revision 1.2 3

Introduction & Disclaimer 3

Mounting & Unmounting databases..... 3

Archive Databases 3

 Create Archive Databases 4

 WATCH-OUT 4

 Syntax 4

HANDY SQL..... 4

 Search for unique keys..... 4

 Finding a which table contains a given column name 5

 How to a get a list of triggers on the table 5

 Listing Zombie processes..... 5

 Identifying bottlenecks in Sybase..... 5

Licensing..... 6

Using multiple & sa tempdbs..... 6

Moving or reducing tempdb in Sybase..... 7

Mixed data and log devices removal 8

Moving disk devices 9

Copy statistics from one database to another..... 10

Fix Syslogs free space count errors in sybsecurity..... 10

Fix sp_helpdb/sp_helpsegment is returning negative numbers 11

Sybase Database ASE Server Startup & Stop script for linux RHEL for run level 3, 5: 11

Document Revision 1.2

Introduction & Disclaimer

The notes contained in this document are intended as a fast find guide to using Sybase Adaptive Server Enterprise and have been built up over my time using ASE in the real world. It is not intended to be a complete exploration of all of ASE server's abilities, nor do I claim that all the notes are without error. If you find errors or would like to submit your own top tip for the next edition of this guide, then please email us at info@ddsafe.co.uk

Mounting & Unmounting databases

Syntax:

```
mount database all from <manifest_file> [with listonly]
```

Example:

First, make sure mirroring of devices is disabled

```
sp_configure 'disable disk mirroring'
```

```
go
```

If not, set to '1' and reboot ASE

The devices must be wholly used by the database in question. The simplest devices are those found on filesystem, not RAW devices in Unix. To unmount a database to a 'manifest file'

```
USE master
```

```
go
```

```
unmount database pubs2 to 'c:\sybase_15\pubs2.umount.manifest'
```

```
go
```

```
sp_helpdb --check DB has disappeared.
```

```
go
```

```
mount database all from 'c:\sybase_15\pubs2.umount.manifest'
```

```
go
```

```
online database prim_db
```

```
go
```

If you want to move the devices or database to another location, use the following 'mount' syntax.

```
mount database all from 'c:\sybase_15\pubs2.umount.manifest'
```

```
using 'C:\sybase_new_devs\pubs2_data01.dat'='pubs2_data01',
```

```
.....any other devices.....
```

```
go
```

Note:

It is good practice to issue a "quiesce database" on the database, prior to unmounting it. This Suspends and resumes updates to a specified list of databases.

Can only mount/unmount if the set of databases that occupy those devices are isolated, self-contained on those devices.

Can unmount several databases at the same time to a single manifest file.

Archive Databases

Databases must be dumped using the new syntax...

```
dump databases dbname to '/path/dbname_1.cmp'
```

```
stripe on '/path/dbname_2.cmp'
```

```
with compression='1'
```

```
go
```

```
3
```

www.ddsafe.co.uk

Create Archive Databases

First you need to create a small scratch database which will hold meta data for the archive database. The only interesting thing with it is that after you have built it you need to set the dboption **scratch database** to true for it. It needs to be quite large. The example below is based on a 220Gb DB.

```
create database scratch_db on datadev1 = 5000
log on logdev1 = 2000
go

exec sp_dboption "scratch_db", "scratch database", "true"
exec sp_dboption "scratch_db", "trunc log on chkpt", "true"
go
```

WATCH-OUT

If you are loading from a compressed dump then you need to run
 sp_configuration 'compression memory size', 64

You may also need to increase your procedure cache. The scratch database needs to be around 5GB data/2GB Log for the recovery of a 220Gb database. Make sure you change the ownership of the database to match the database to be restored prior to starting the load. This cannot be altered later and may be important for user access. One way is to give the DB owner sa_role, create the DB, then revoke the sa_role. The progress of the load is not shown in the backuplog. You can only tell if it is working by checking the physical_io from sysprocesses. The ASE errorlog will show you that the dump files have been initialised as devices when you start the load.

Syntax

The syntax for creating the archive database is (as DB owner):

```
create archive database archive_db on datadev1=1000
with scratch_database = scratch_db
go
USE archive_db
go
EXEC sp_changedbowner 'dbname'
go
```

You can then load the dump into the archive database with

```
Load database archive_db from "/dumps/dbname.dmp"
Go
Online database archive_db
go
```

The database is now ready for browsing and you can also run dbcc checks etc against it.

HANDY SQL

Search for unique keys

```
--list tables with no unique index
select convert(varchar(30), o.name)
from sysobjects o
where o.type="U"
and o.id not in (select i.id
from sysobjects o, sysindexes i
where o.type="U"
and o.id=i.id
and i.status & 2 = 2 --2 if unique index
)
```

--list tables and primary keys

4

www.ddsafe.co.uk

```
select convert(varchar(30), object_name(i.id)), convert(varchar(30),i.name)
from sysobjects o, sysindexes i
where o.type="U"
and o.id=i.id
and i.status & 2 = 2
```

Finding a which table contains a given column name

```
select o.name
from sysobjects o, syscolumns c
where o.id = c.id
and c.name = <column name>
```

How to a get a list of triggers on the table

```
select "table name" = name
,"insert trigger" = convert(varchar, object_name(instrig))
,"update trigger" = convert(varchar, object_name(updtrig))
,"delete trigger" = convert(varchar, object_name(deltrig))
from sysobjects
where type = "U"
--and name = @table_name
order by name
```

Listing Zombie processes

```
select spid from syslocks where spid not in (select spid from sysprocesses)
```

Identifying bottlenecks in Sybase

Capture showplan & query stats.

```
set tracefile '/tmp/filename' for SPID
go
```

```
set showplan on
set statistics time on
set statistics plancost on
set option show_abstract_plan on
go
```

With the plancost info we should be able to identify any optimiser errors by comparing the estimated rows and IO with the actual figures. Most likely, there will not be any and it will just be a case of ensuring the high IO / time sections of the SQL are using the correct indexes. If they are not (and update all stats does not fix it) then we add an index hint to force the issue (or tweak the abstract plan).

Licensing

From version 12.5 ASE we need to use SYAM licensing. This is a bit of a pain in the back side. You can install 2-3 network licences. These are used by the servers to 'check-out' a license. Here we will look at installing stand alone licences.

Download the correct license from the Sybase site.

For SBE edition, you need to create these licenses for each server. It uses the mac address of the PC to create these.

Check currently installed licence type

```
sp_lmconfig 'edition', 'SE' --Small Business Edition
sp_lmconfig 'license type', 'DH' -- Development license
```

Set the variable SYBASE_SAM_CPUINFO in both user and system environment variables
SYBASE_SAM_CPUINFO=CPUID

Using multiple & sa tempdbs

Bind 'sa' to a tempdb

```
-----
CREATE TEMPORARY DATABASE tempdbsa
ON tempdb03=50,
tempdb03=50
go
sp_tempdb bind, lg, sa, DB, tempdbsa
go
```

Explanation:

```
sp_tempdb bind, objtype, objname, bindtype, bindobj
objtype=object type (login_name (or LG), application_name (or AP))
bindtype=bind type (group (or GR), database (or DB))
```

Testing:

To check what you have bound the logins to the database

```
sp_tempdb 'show', 'all'
```

To check if login is using the tempdb, login with new session to server

```
sp_tempdb 'who', tempdbsa
```

Round-robin tempdb

```
=====
having created multiple tempdb's, add them to the 'default' group
sp_tempdb 'add', 'tempdb_2', 'default'
go
sp_tempdb 'add', 'tempdb_3', 'default'
go
sp_tempdb 'add', 'tempdb_4', 'default'
go
```

Moving or reducing tempdb in Sybase

1. restart ASE server in single user mode (using -m switch)

2. backup master database and BCP out tables:-

```
master..sysusages
master..sysdevices
master..sysdatabases
master..syslogins
master..sysconfigures
master..syscharsets
```

3. Do the work!

```
1> use master
```

```
2> go
```

```
1> sp_configure "allow updates", 1
```

```
2> go
```

```
1> begin transaction
```

```
2> go
```

```
1> select * from sysusages
```

```
2> where dbid = db_id('tempdb')
```

```
3> go
```

--Set the first 2MB of tempdb back to data and log in case they were separated:

```
1> update sysusages
```

```
2> set segmap = 7 where dbid = db_id('tempdb')
```

```
3> and lstart = 0
```

```
4> go
```

--Should look a bit like this.....

dbid	segmap	lstart	size	vstart	pad	unreservedpgs	crdate
2	7	0	1024	8196	NULL	423	Dec 28 2006 3:43PM
2	7	1024	512000	33554432	NULL	510000	Feb 12 2007 9:57AM
2	7	513024	512000	50331648	NULL	510000	Feb 12 2007 9:57AM

--Delete all other rows belonging to tempdb from sysusages.

--The number of rows affected should be one less than the number of rows affected by the previous select command.

```
1> delete sysusages where dbid = db_id('tempdb')
```

```
2> and lstart != 0
```

```
3> go
```

--Verify that tempdb has one entry that looks like this:

```
1> select * from sysusages
```

```
2> where dbid = db_id('tempdb')
```

dbid	segmap	lstart	size	vstart
2	7	0	1024	2564

7

www.ddsafe.co.uk

```
--If you see a problem
1> rollback transaction
2> go
--else
1> commit transaction
2> go
1> sp_configure "allow updates", 0
2> go
```

```
--immediately issue a checkpoint and shut down Adaptive Server:
--You must shut down Adaptive Server before altering the size of tempdb again.
--Extend database as usual.
```

```
Note on removing multiple tempdb's
sp_tempdb 'show', 'all'
go
sp_tempdb 'remove', 'tempdb_1', 'default'
go
```

Then restart ASE.

```
drop database tempdb_1
go
```

Mixed data and log devices removal

```
1.
dump tran mxgfxo_dev_02_db with truncate_only
go
2.
dump database mxgfxo_dev_02_db to
'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_prefix_1.cmp'
stripe on 'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_prefix_2.cmp'
stripe on 'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_prefix_3.cmp'
stripe on 'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_prefix_4.cmp'
stripe on 'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_prefix_5.cmp'
stripe on 'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_prefix_6.cmp'
go
Get DDL of database using DBAArtisan
3. backup systables
~/bin/bcp_systables.ksh ACC51_ASE
4. select * from sysusages where dbid = db_id('mxgfxo_dev_02_db')
5. Check if at least one of the segmap=4 and is not mixed with any of the data device of this database. This
can be varified with the following command
select a.segmap, b.name
from sysusages a, sysdevices b
where dbid = db_id('mxgfxo_dev_02_db')
and a.vstart between b.low and b.high
go
Otherwise alter database <dbname> log on <new_log_device> = <size> Check sysusages again to make
sure segmap = 4 for this
6. set segmap to 'data' for mixed devices
sp_configure 'allow updates', 1
go
8
www.ddsafe.co.uk
```



```

update sysusages set segmap = 3 where dbid = 14 and segmap = 7
go
sp_configure 'allow updates', 0
go
7. Shutdown/restart server.
8. create a dummy table and insert 1000 rows. This will make new page allocation for syslogs move to the
new logonly device.
create table dummy_table ( xxx char(255) )
go
insert dummy_table select "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
go 1000
9. Dump tran with truncate_only.
dump tran <dbname> with truncate_only
go
10. Shutdown and restart server and insert 1000 rows to create few logs. insert dummy_table select
"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
go 1000
11. dump tran <dbname> to "device_name"
go
dump database mxgfxo_dev_02_db to
'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_fixed_1.cmp'
stripe on 'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_fixed_2.cmp'
stripe on 'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_fixed_3.cmp'
stripe on 'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_fixed_4.cmp'
stripe on 'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_fixed_5.cmp'
stripe on 'compress::1::/home/sybase/dump01/ACC51_ASE/load/mxgfxo_dev_02_db_fixed_6.cmp'
go
Get DDL of database using DBAArtisan
12. Now drop database
drop database mxgfxo_dev_02_db
go
13. Recreate database using correct segments. Modify the extracted DDL from step 11.
14. load database using stripes from step 11.
15. Finally, check new database
use master
go
sp_helpdb mxgfxo_dev_02_db
go
use mxgfxo_dev_02_db
go
sp_helplog

!!! That's it !!!!

```

Moving disk devices

This example also renames the device (should you want to)

```

sp_helpdevice data_01
go
disk mirror name="data_01", mirror="c:\sybase_devs\data_01.dat"
go
disk unmirror name="data_01", side="primary", mode="remove"
go

```

The old device file will still exist on the server, so once you have dropped it from Sybase (sp_dropdevice), don't forget to go back and remove it from the file system.

Copy statistics from one database to another

1) Extract the statistics for the database with the following example command

```
optdiag binary statistics -Sserver1 dbname_copy2 -Usa -Ppassword -odbname_copy2.opt
```

2) Edit the output file and change the database name found in the header to match the database you will be applying the statistics to. Look for:

```
Specified database: "new_dbname"
```

You may also need to edit the servername - Sybooks does not mention needing to change this.

3) Inject the statistics as per the following example:

```
optdiag binary statistics -Sserver2 -Usa -Ppassword -idbname_copy2.opt
```

Fix Syslogs free space count errors in sybsecurity

```
-- sp__segmentmonitor is reporting free space incorrectly
-- This uses sum(curunreservedpgs(dbid,lstart,unreservedpgs))
-- dbcc tablealloc(syslogs,full,fix) can only be run in DB in single user mode!
```

```
--
sp_configure 'allow updates', 1
```

```
go
```

```
--save away this value (should be 0)
```

```
select status from master..sysdatabases where name = "sybsecurity"
```

```
go
```

```
begin tran
```

```
update master..sysdatabases set status=4096 where name = "sybsecurity"
```

```
go
```

```
select status from master..sysdatabases where name = "sybsecurity"
```

```
go
```

```
commit --if expected value
```

```
shutdown with nowait --not 100% sure if we need 'nowait'
```

```
go
```

```
--Once server restart
```

```
sp_configure 'auditing', 0
```

```
go
```

```
use sybsecurity
```

```
go
```

```
dbcc tablealloc(syslogs,full,fix)
```

```
go
```

```
*****
```

```
TABLE: syslogs          OBJID = 8
```

```
PARTITION ID=8 FIRST=51262 ROOT=51284 SORT=0
```

```
Syslogs free space count has been successfully recalculated. It has been corrected to 50977 pages.
```

```
Data level: indid 0, partition 8. 23 Data pages allocated and 3 Extents allocated.
```

```
TOTAL # of extents = 3
```

```
Alloc page 51200 (# of extent=4 used pages=26 ref pages=23)
```

```
Total (# of extent=4 used pages=26 ref pages=23) in this database
```

```
10
```

```
www.ddsafe.co.uk
```

DBCC execution completed. If DBCC printed error messages, contact a user with System Administrator (SA) role.

```
--put database out of single user mode
begin tran
update master..sysdatabases set status=<use saved value from above> where name = "sybsecurity"
go
select status from master..sysdatabases where name = "sybsecurity"
go
commit
go
sp_configure 'allow updates',0
go
```

Fix sp_helpdb/sp_helpsegment is returning negative numbers

We sometimes see ASE return negative numbers for sp_helpdb. One solution given by Sybase is to restart the server. Hmm... not always possible. An alternative is to use the dbcc command 'usedextents'. Issue the following:

```
dbcc gam(dbname, 0, 0, "fix")
dbcc usedextents(dbname, 0, 1, 1)
```

and the problem should disappear.

(top tip, try running "dbcc tablealloc(syslogs, full, fix)" first before you run the DBCC GAM() command. For some reason it seem to get the DBCC GAM to run much faster)

Sybase Database ASE Server Startup & Stop script for linux RHEL for run level 3, 5:

1. Copy the below script.
2. vi sybase
3. edit to change the servers name & path
4. cp sybase /etc/init.d/sybase
5. chmod 755 /etc/init.d/sybase
6. Now find the syslog number by
ls /etc/rc3.d/S* & ls /etc/rc5.d/S*
7. If 87 is free then create soft link for both run level 3 & 5 by
ln -s /etc/init.d/sybase /etc/rc3.d/S87sybase
ln -s /etc/init.d/sybase /etc/rc5.d/S87sybase

We can also use to start & stop the server manually

```
./sybase stop
./sybase start
```

```
#####  
#!/bin/sh  
#-----  
# Sybase process start / stop  
#  
# Copy to /etc/init.d  
#-----  
### BEGIN INIT INFO  
# chkconfig: 2345 24 94  
# Provides:          sybase  
# Required-Start:    $remote_fs $syslog  
# Required-Stop:     $remote_fs $syslog  
# Default-Start:     2 3 4 5  
# Default-Stop:      0 1 6  
# Short-Description: Start and stop Sybase ASE & BS at boot time and  
shutdown/reboot  
# Description:       Enable service provided by sybase.  
#  
# Customise the following to suit environment  
#     $SYBASE  
#     $user & $pwd. Our example reads this from a file  
#     $server  
### END INIT INFO  
  
SYBASE=/opt/sybase16  
export SYBASE  
  
LD_LIBRARY_PATH=/opt/sybase16/lib  
export LD_LIBRARY_PATH  
  
. /opt/sybase16/SYBASE.sh  
  
logdir=/opt/sybase16/logs  
if [ ! -d $logdir ];then mkdir $logdir;fi  
  
case "$1" in  
  'stop')  
#-----  
# Sybase shutdown  
#-----  
  
logfile=$logdir/k87sybase.log  
ifile=/opt/sybase16/interfaces  
server=UAT2  
  
user=$(egrep "^$server," /opt/dba/scripts/servers.txt.sa | cut -d, -f2)  
pwd=$(egrep "^$server," /opt/dba/scripts/servers.txt.sa | cut -d, -f3)  
ISQLCMD="isql -b -U${user} -P${pwd} -S${server} -w1000"  
  
echo "`date` Shutting down Sybase server ..." >> $logfile  
  
#Execute shutdown (BS then ASE)  
$ISQLCMD <<EOF>>$logfile  
checkpoint  
go  
shutdown SYB_BACKUP  
go  
shutdown  
go  
EOF  
  
sleep 3  
  
echo "`date` Shutdown complete " >> $logfile  
  
12  
www.ddsafe.co.uk
```

```

;;

'start')

#-----
# Sybase startup
#-----
tstamp=`date '+%y%m%d%H%M'`
echo "$tstamp: Starting Sybase ASE" > /opt/sybase16/logs/Sybase_Startup_Log.txt
# initialize variables
logfile=$logdir/S87sybase.log
dsfile=$logdir/ds-boot.log ; touch $dsfile ; chmod 766 $dsfile
server=UAT2
dscmd="/opt/sybase16/ASE-16_0/bin/startserver -f /opt/sybase16/ASE-
16_0/install/RUN_${server}"
bsfile=$logdir/bs-boot.log
bscmd="/opt/sybase16/ASE-16_0/bin/startserver -f /opt/sybase16/ASE-
16_0/install/RUN_${server}_BS"
tstamp=`date '+%y%m%d%H%M'`

echo "Startup of Sybase server on `date`" >>$logfile

# clean archive dir, 180 days old
/usr/bin/find $logdir -mtime +180 -exec rm {} \;

# move log file to archive directory
mv $SYBASE/ASE-16_0/install/${server}_BS.log $logdir/${server}_BS.$tstamp.log
mv $SYBASE/ASE-16_0/install/${server}.log $logdir/${server}.$tstamp.log

sleep 1

# create new logfiles and open up perms so other users can view
touch $SYBASE/ASE-16_0/install/${server}_BS.log
touch $SYBASE/ASE-16_0/install/${server}.log
chmod 777 $SYBASE/ASE-16_0/install/${server}_BS.log
chmod 777 $SYBASE/ASE-16_0/install/${server}.log

sleep 1

# -----
# Start data server
# -----

su - sybase -c "$dscmd > $dsfile"

sleep 3

# -----
# Start backup server
# -----

su - sybase -c "$bscmd > $bsfile" >/dev/null 2>&1

echo "Startup of Sybase completed on `date`" >>$logfile

;;

*)
    echo "Usage: $0 { start | stop }"
    exit 1
;;
esac

exit 0

```