SYBASE DBA Commands

Sybase ASE 15.0

Prepared by – Abhisek Vyas

Document Version 1.0

Team, SybaseRays.com

Process Start: How to start Sybase ASE server from RUNFILE:

==========

1. login into unix box

2. cd /opt/sybase/ASE-15_0/install (cd to your sybase intall dir)

3. ./startserver -f RUN_SYBASERAYS_SERVER_1

Note: You will see RUN_<SERVER_NAME> in install dir. In my case server name is SYBASERAYS_SERVER_1 and runfile name is RUN_SYBASERAYS_SERVER_1. run file name used to be create by intall process, when we install sybase/create server.

Process End:

==========

Process Start: How to start Sybase ASE back up server from RUNFILE:

==========

1. login into unix box

2. cd /opt/sybase/ASE-15_0/install (cd to your sybase intall dir)

3. ./startserver -f RUN_SYBASERAYS_SERVER_1_BS

Note: You will see RUN_<SERVER_NAME>_BS in install dir. In my case server name is SYBASERAYS_SERVER_1 and runfile name is RUN_SYBASERAYS_SERVER_1_BS. run file name used to be create by intall process, when we install sybase/create server.

Process End:

==========

Process Start: diskinit

==========

A System Administrator initializes new database devices with the disk init command, which:

- Maps the specified physical disk device or operating system file to a database device name
- Lists the new device in master..sysdevices
- Prepares the device for database storage

Syntax:

disk init

name = "device_name" ,

physname = "physicalname" ,

[vdevno = virtual_device_number ,]

size = number_of_blocks

[, vstart = virtual_address

, cntrltype = controller_number]

[, contiguous]

[, dsync = { true | false }]

[, directio = {true | false}

Example:

disk init name = "test_disk",

physname = "/opt/sybase/data/test_disk.dat", size = "1M"

go

Note: above command should run while you are connected to a Sybase Server.

Note: To see the numbers already in use for vdevno, look in the vdevno column of the report from **sp_helpdevice**, or use the following query to list all the device numbers currently in use:

select vdevno from master..sysdevices

where status & 2=2

Here, "status 2" specifies **physical disk**.

Process End:

==========

Process Start: sp_diskdefault (for Choosing default and nondefault devices)

==========

Syntax: sp_diskdefault logicalname, {defaulton | defaultoff}

Example: sp_diskdefault master, defaultoff

or

sp_diskdefault test_disk, defaulton

To create a pool of default database devices to be used by all Adaptive Server users for creating databases, use sp_diskdefault after the devices are initialized. sp_diskdefault marks these devices in sysdevices as default devices. Whenever users create (or alter) databases without specifying a database device, new disk space is allocated from the pool of default disk space.

Process End:

==========

Process Start: disk resize

==========

The disk resize command allows you to increase the size of your database devices dynamically, rather than initializing a new device.

Syntax:

disk resize

name = "device_name",

size = additional_space

Note: You cannot use disk resize on dump or load devices.

Note: The minimum size for disk resize is 1MB or one allocation unit, whichever is greater.

Process End:

==========

Process Start: create database

==========

Syntax: Create database database_name

[on {default | database_device} [= size]

[, database_device [= size]...]

[log on database_device [= size]

[, database_device [= size]]...]

[with {override | default_location = "pathname"}]

[for {load | proxy_update}]

create database db1

on test_disk = "10M"

go

Note: If you use the default keyword, or if you omit the on clause, Adaptive Server puts the database on one or more of the default database devices specified in master..sysdevices.

Note: If you omit the size parameter in the on clause, Adaptive Server creates the database with a default amount of space. This amount is the larger of the sizes specified by the default database size configuration parameter and the model database.

Always place the log on a separate database device (recommended).

Example:

create database db1

on test_disk = "10M"

log on log1

go

Note: As a general rule, allocate to the log 10 to 25 percent of the space that you allocate to the database. for example database size is 100MB then log size should be between 10MB to 25MB.

For log device:

alter database db1 log on log1

go

Moving the transaction log to another device:

If you did not use the log on clause to create database, follow the instructions in this section to use sp_logdevice to move your transaction log to another database device.

Syntax: sp_logdevice database_name, devname

Note: You should set database in "single user" mode before executing above command otherwise you will receive below error:

Error: System Administrator (SA) must set database 'db1' to single-user mode with

sp_dboption before using 'sp_logdevice

Before executing sp_logdevice

```
4> sp_helpdb db1
5> go
 name    db_size   owner     dbid  created       durability   lobcomplvl  inrowlen status
 -------- ---------------------------- ---------- --------
 db1     11.0 MB   sa           6   Dec 27, 2012   full           0   NULL       single user, mixed log and data

 device_     fragments size   usage                created          free kbytes
 ---------------- ----         -----------------          ------------------  ---------------
 test_disk   10.0 MB   data and log only      Dec 27 2012  8:13AM  6856
 log1        1.0 MB    log only               Dec 27 2012  8:49AM  not applicable

 -------------------------------------
 log only free kbytes = 7876
```

Set in a single user mode:

```
------------------------------
use master
go
sp_dboption "db1", "single user", true
go
---------------------------
After executing sp_logdevice
4> sp_helpdb db1
5> go
 name    db_size   owner     dbid  created       durability   lobcomplvl  inrowlen status
 -------- ---------------------------- ---------- --------
 db1     11.0 MB   sa       6   Dec 27, 2012   full       0   NULL       single user, mixed log and data

 device_fragments size   usage           created          free kbytes
 ---------------- ----   -----------------  -------------------  ---------------
 test_disk   10.0 MB   data only       Dec 27 2012  8:13AM  6856
 log1        1.0 MB    log only        Dec 27 2012  8:49AM  not applicable
-------------------------------------
 log only free kbytes = 1020
```

Process End:

==========

Process Start: drop database

==========

Syntax:

drop database database_name [, database_name]...

1> drop database db1

2> go

Use drop database to remove a database from Adaptive Server, thus deleting the database and all the objects in it. This command:

- Frees the storage space allocated for the database
- Deletes references to the database from the system tables in the master database

Note: Only the Database Owner can drop a database.

Note: You must be in the master database to drop a database.

Note: You cannot drop a database that is open for reading or writing by a user

Note: After you drop a database, dump the master database to ensure recovery in case master is damaged.

Process End:

==========

Process Start: Changing database ownership

==========

A System Administrator might want to create the user databases and give ownership of them to another user. sp_changedbowner changes the ownership of a database. The procedure must be executed by the System Administrator in the database where the ownership is to be changed.

Syntax: sp_changedbowner loginame [, true]

Example:

sp_changedbowner abhi

Above example makes the user “abhi” the owner of the current database and drops the aliases of users who could act as the former “dbo.”

Note: You cannot change ownership of the master database. It is always owned by the “sa” login.

Process End:

==========

Process Start: dump a user db.

==========

The dump database command makes a copy of the entire database, including both the data and the transaction log. dump database does not truncate the log

Example:

1> dump database db1 to "/dev/SYBASERAYS_SERVER_1/dumpfiles/dump1"

2> go

Messages after running above command:

Backup Server session id is: 5. Use this value when executing the

'sp_volchanged' system stored procedure after fulfilling any volume change

request from the Backup Server.

Backup Server: 4.41.1.1: Creating new disk file

/dev/SYBASERAYS_SERVER_1/dumpfiles/dump1.

Backup Server: 6.28.1.1: Dumpfile name 'db11236209A27 ' section number 1

mounted on disk file '/dev/SYBASERAYS_SERVER_1/dumpfiles/dump1'

Backup Server: 4.188.1.1: Database db1: 752 kilobytes (34%) DUMPED.

Backup Server: 4.188.1.1: Database db1: 934 kilobytes (100%) DUMPED.

Backup Server: 3.43.1.1: Dump phase number 1 completed.

Backup Server: 3.43.1.1: Dump phase number 2 completed.

Backup Server: 3.43.1.1: Dump phase number 3 completed.

Backup Server: 4.188.1.1: Database db1: 944 kilobytes (100%) DUMPED.

Backup Server: 3.42.1.1: DUMP is complete (database db1).

Note: Make sure backup serve is up and running otherwise above command will fail.

Process End:

==========

Process Start: load a user db.

==========

Example:

3> load database db1 from "/dev/SYBASERAYS_SERVER_1/dumpfiles/dump1"

4> go

Backup Server session id is: 12. Use this value when executing the

'sp_volchanged' system stored procedure after fulfilling any volume change

request from the Backup Server.

Backup Server: 6.28.1.1: Dumpfile name 'db11236209A27 ' section number 1

mounted on disk file '/dev/SYBASERAYS_SERVER_1/dumpfiles/dump1'

Backup Server: 4.188.1.1: Database db1: 3780 kilobytes (33%) LOADED.

Backup Server: 4.188.1.1: Database db1: 11270 kilobytes (100%) LOADED.

Backup Server: 4.188.1.1: Database db1: 11280 kilobytes (100%) LOADED.

Backup Server: 3.42.1.1: LOAD is complete (database db1).

Started estimating recovery log boundaries for database 'db1'.

Database 'db1', checkpoint=(925, 50), first=(925, 50), last=(925, 50).

Completed estimating recovery log boundaries for database 'db1'.

Started ANALYSIS pass for database 'db1'.

Completed ANALYSIS pass for database 'db1'.

Started REDO pass for database 'db1'. The total number of log records to process

is 1.

Completed REDO pass for database 'db1'.

Use the ONLINE DATABASE command to bring this database online; ASE will not

bring it online automatically.

Process End:

==========

Process Start: Making a database online after load .

==========

3> online database db1

4> go

Started estimating recovery log boundaries for database 'db1'.

Database 'db1', checkpoint=(925, 50), first=(925, 50), last=(925, 50).

Completed estimating recovery log boundaries for database 'db1'.

Started ANALYSIS pass for database 'db1'.

Completed ANALYSIS pass for database 'db1'.

Recovery of database 'db1' will undo incomplete nested top actions.

Database 'db1' is now online.

Process End:

==========

Process Start: disk mirroring

==========

Disk mirroring can provide nonstop recovery in the event of media failure. The disk mirror command causes an Adaptive Server database device to be duplicated, that is, all writes to the device are copied to a separate physical device. If one device fails, the other contains an up-to-date copy of all transactions.

Sybase recommends to mirror following devices:

- All default databases
- User databases
- Transaction log (you should always put their transaction logs on a separate database device)

Disk Mirror Syntax:

disk mirror

name = "device_name" ,

mirror = "physicalname"

[, writes = { serial | noserial }]

Example:

1> disk mirror

2> name = "log1", mirror = "/opt/sybase/data/log1_mirror.dat"

3> go

Creating the physical file for the mirror...

Starting Dynamic Mirroring of 512 pages for logical device 'log1'. 256 pages mirrored...

The remaining 256 pages are currently unallocated and will be mirrored as they are allocated.

Note: Disk mirroring is not designed to detect or prevent database corruption. Some of the scenarios described can cause corruption, so you should regularly run consistency checks such as dbcc checkalloc and dbcc checkdb

on all databases.

The disk mirror, disk unmirror, and disk remirror commands control disk mirroring. All the commands can be issued while the devices are in use, so you can start or stop database device mirroring while databases are being used.

Note: The disk mirror, disk unmirror, and disk remirror commands alter the sysdevices table in the master database. After issuing any of these commands, dump the master database to ensure recovery in case master is damaged.

Note: enable disk mirroring before executing disk mirror commands by executing below command.

1> sp_configure 'disable disk mirroring', 0

2> go

Note: Above command will not take effect dynamically, so you need to restart server.

Note: You cannot mirror a dump device.

Process End:

==========

Process Start: Adding loings to ASE server .

==========

Use sp_addlogin to add a new login name to Adaptive Server. You do not use it to give the user permission to access user databases. Use sp_adduser for that purpose. Only the System Security Officer can execute sp_addlogin.

Syntax:

sp_addlogin loginame, passwd [, defdb] [, deflanguage [, fullname]]]

Example:

6> sp_addlogin vyasab, Sunday01, NULL, NULL, "Abhisek Vyas"

7> go

Password correctly set.

Account unlocked.

New login created.

Note: Passwords must be at least 6 character(s)

You can use below command to define defualt database while adding login.

1> sp_addlogin user2, password, db1, NULL, "user2"

2> go

Note: A System Administrator can change anyone's default database with sp_modifylogin. Other users can change only their own default database.

After specifying the default database, add the user to the default database with sp_adduser so that he or she can log in directly to the default database. Also, SA can define defualt database with sp_moifylogin after creating login.

Alternatively, you can specify a parameter name, in which case you do not have to specify all the parameters. For example:

1> sp_addlogin user3, password, @defdb = db1

2> go

Note: When you execute sp_addlogin, Adaptive Server adds a row to master.dbo.syslogins, assigns a unique user ID (suid) for the new user, and fills in other information. When a user logs in, Adaptive Server looks in syslogins for the name and password provided by the user. The password column is encrypted with a one-way algorithm so it is not human-readable.

Process End:

==========

Process Start: Adding groups to ASE Server

==========

Every user is a member of the group "public" and can also be a member of one other group. (Users remain in "public," even when they belong to another group.)

syntax: sp_addgroup grpname

example:

1> sp_addgroup group1

2> go

New group added.

Note: The System Administrator can assign or reassign users to groups with sp_changegroup.

Process End:

==========

Process Start: Adding users to database .

==========

The Database Owner or a System Administrator can use sp_adduser to add a user to a specific database. The user must already have an Adaptive Server login.

syntax: sp_adduser loginame [, name_in_db [, grpname]]

Example:

6> use db1

7> go

1> sp_adduser vyasab

2> go

New user added.

Note: If no name_in_db parameter is given, the name inside the database is the same as loginame.

Example:

3> sp_adduser user2, user2_db, group1

4> go

New user added.

Note: sp_adduser adds a row to the sysusers system table in the current database.

Adding a "guest" user to a database: Creating a user named "guest" in a database enables any user with an Adaptive Server account to access the database as a guest user. If a user issues the use database_name command, and his or her name is not found in the database's sysusers or sysalternates table, Adaptive Server looks for a guest user. If there is one, the user is allowed to access the database, with the permissions of the guest user.

Syntax: sp_adduser guest

Process End:

==========

Process Start: To create a database that is to be owned by another user by SA:

==========

1. Issue the create database command in the master database.

2. Switch to the new database with the use command.

3. Execute sp_changedbowner.

Sybase suggests that you transfer ownership before the user has been added to the database, and before the user has begun creating objects in the database.

The new owner must already have a login name on Adaptive Server, but cannot be a user of the database, or have an alias in the database. You may have to use

sp_dropuser or sp_dropalias before you can change a database's ownership, and you may have to drop objects before you can drop the user.

Issue sp_changedbowner in the database whose ownership is to be changed.

Syntax:

sp_changedbowner loginame [, true]

Example: sp_changedbowner abhi

This example makes “abhi” the owner of the current database and drops aliases of users who could act as the old “dbo:”

Include the true parameter to transfer aliases and their permissions to the new “dbo.”

Note: You cannot change the ownership of the master database and should not change the ownership of any other system databases.

==========

Process Start: Granting roles

==========

To grant roles to users or other roles, use:

Syntax: grant role role_granted [{, role_granted}...]to grantee [{, grantee}...]

where:

- role_granted – is the role being granted. You can specify any number of roles to be granted.
- grantee – is the name of the user or role. You can specify any number of grantees.

Example:

1> grant role oper_role to vyasab

2> go

You can grant one role to another role as well.

Example:

1> grant role sso_role to oper_role

2> go

Use revoke role to revoke roles from users and other roles:

Syntax: revoke role role_name [{, role_name}...]from grantee [{, grantee}...]

Process End:

==========

Process Start: Plan dbccdb database

==========

- Run sp_plan_dbccdb in the master database to obtain recommendations for database size, devices, workspace sizes, cache size, and the number of worker processes for the target database (for example targer database is db1, which is a user database).

```
3> use master
4> go
1> sp_plan_dbccdb db1
2> go
```

Recommended size for dbccdb database is 20MB (data = 18MB, log = 2MB).

No suitable devices for dbccdb in master..sysdevices.
Recommended values for workspace size, cache size and process count are:

```
dbname              scan ws  text ws  cache  comp mem  process count
----------          --------- ------- ------- ------ ---------- ----
db1                  192K    96K     1280K  0K       2
```

♦ Use disk init to initialize disk devices for dbccdb data and log

---

```
1> disk init
2> name = "db1_dbccdb_data",
3> physname = "/opt/sybase/data/db1_dbccdb_data.dat"
4> ,size = "18M" -- Recommended data size for dbccdb database (sp_plan_dbccdb db1)
5> go
```

```
3> disk init
4> name = "db1_dbccdb_log",
5> physname = "/opt/sybase/data/db1_dbccdb_log.dat"
6> , size = "2M" -- Recommended log size for dbccdb database (sp_plan_dbccdb db1)
7> go
```

♦ Create dbccdb database on data disk device ("db1_dbccdb_data" created earlier)

```
1> create database dbccdb
2> on db1_dbccdb_data
3> log on db1_dbccdb_log
4> go
CREATE DATABASE: allocating 1536 logical pages (6.0 megabytes) on disk
'db1_dbccdb_data' (1536 logical pages requested).
CREATE DATABASE: allocating 512 logical pages (2.0 megabytes) on disk
```

'db1_dbccdb_log' (1536 logical pages requested).

Database 'dbccdb' is now online.

♦ Create the tables for dbccdb and initialize the dbcc_types table:

[root@localhost install]# isql -S SYBASERAYS_SERVER_1 -U vyasab -P Sunday01 -i /opt/sybase/ASE-15_0/scripts/installdbccdb

(return status = 0)

(1 row affected)

(1 row affected)

Msg 911, Level 11, State 2:

Server 'SYBASERAYS_SERVER_1', Line 24:

Attempt to locate entry in sysdatabases for database 'dbccalt' by name failed - no entry found under that name. Make sure that name is entered properly.

(1 row affected)

Database option 'select into/bulkcopy/pllsort' turned ON for database 'dbccdb'.

Running CHECKPOINT on database 'dbccdb' for option 'select into/bulkcopy/pllsort' to take effect.

Creating dbcc_types table

Object name has been changed.

Warning: Changing an object or column name could break existing stored

♦ Create and initialize the scan and text workspaces

---

use dbccdb

go

sp_dbcc_createws dbccdb, scanseg, scan_pubs2, scan, "64K"

sp_dbccvreatews dbccdb, textseg, text_pubs2, text, "64K"

Process End:

==========

Process Start: Create server using srvbuilders

==========

[root@localhost bin]# cd $SYBASE/$SYBASE_ASE/bin

[root@localhost bin]#

./srvbuildres -r /opt/sybase/ASE-15_0/init/sample_resource_files/srvbuild.adaptive_server_2.rs

Warning: You have selected '4k' as the logical page size for the Adaptive

Server. If you plan to load dump from another database, make sure this logical page size matches the size of the source database. The default logical page size in previous Adaptive Server versions was 2KB.

Building Adaptive Server 'SYBASERAYS_SERVER_2':

Writing entry into directory services...

Directory services entry complete.

Building master device...

Master device complete.

Writing RUN_SYBASERAYS_SERVER_2 file...

RUN_SYBASERAYS_SERVER_2 file complete.

Starting server...

Server started.

Set SA password...

SA password is set.

Building sysprocs device and sybsystemprocs database...

sysprocs device and sybsystemprocs database created.

Running installmaster script to install system stored procedures...

installmaster: 10% complete.

installmaster: 20% complete.

installmaster: 30% complete.

installmaster: 40% complete.

installmaster: 50% complete.

installmaster: 60% complete.

installmaster: 70% complete.

installmaster: 80% complete.

installmaster: 90% complete.

installmaster: 100% complete.

installmaster script complete.

Creating two-phase commit database...

Two phase commit database complete.

Extending tempdb database ...

Extending tempdb database complete.

Installing common character sets (Code Page 437, Code Page 850, ISO Latin-1,

Macintosh and HP Roman-8)...

Character sets installed.

Setting server name in Adaptive Server...

Server name added.

Setting optimization goal...

Setting optimization goal complete.

Server 'SYBASERAYS_SERVER_2' was successfully created.

FYI: Below are values used for /opt/sybase/ASE-15_0/init/sample_resource_files/srvbuild.adaptive_server_2.rs

sybinit.release_directory: USE_DEFAULT

sybinit.product: sqlsrv

sqlsrv.server_name: SYBASERAYS_SERVER_2

sqlsrv.sa_password: sybaserays

sqlsrv.new_config: yes

sqlsrv.do_add_server: yes

sqlsrv.network_protocol_list: tcp

sqlsrv.network_hostname_list: localhost

sqlsrv.network_port_list: 5000

sqlsrv.application_type: MIXED

sqlsrv.server_page_size: 4k

sqlsrv.force_buildmaster: no

sqlsrv.master_device_physical_name: /opt/sybase/data/master_2.dat

sqlsrv.master_device_size: 73

sqlsrv.master_database_size: 26

sqlsrv.errorlog: /opt/sybase/ASE-15_0/install/SYBASERAYS_SERVER_2.log

sqlsrv.do_upgrade: no

sqlsrv.sybsystemprocs_device_physical_name: /opt/sybase/data/sysprocs_2.dat

sqlsrv.sybsystemprocs_device_size: 172

sqlsrv.sybsystemprocs_database_size: 172

sqlsrv.sybsystemdb_device_physical_name: /opt/sybase/data/sybsysdb_2.dat

sqlsrv.sybsystemdb_device_size:6

sqlsrv.sybsystemdb_database_size: 6

sqlsrv.tempdb_device_physical_name: /opt/sybase/data/tempdbdev_2.dat

sqlsrv.tempdb_device_size: USE_DEFAULT

sqlsrv.tempdb_database_size: USE_DEFAULT

sqlsrv.default_backup_server: SYBASERAYS_SERVER_2_BS

#sqlsrv.addl_cmdline_parameters:

sqlsrv.do_configure_pci: no

#sqlsrv.sybpcidb_device_physical_name: PUT_THE_PATH_OF_YOUR_SYBPCIDB_DATA_DEVICE_HERE

#sqlsrv.sybpcidb_device_size: USE_DEFAULT

#sqlsrv.sybpcidb_database_size: USE_DEFAULT

If sqlsrv.do_optimize_config is set to yes, both sqlsrv.avail_physical_memory and sqlsrv.avail_cpu_num need to be set.

sqlsrv.do_optimize_config: no

sqlsrv.avail_physical_memory: USE_DEFUALT

sqlsrv.avail_cpu_num: USE_DEFUALT

Backupserver:

[root@localhost ASE-15_0]# cd bin

[root@localhost bin]# cd $SYBASE/$SYBASE_ASE/bin

[root@localhost bin]#

./srvbuildres -r /opt/sybase/ASE-15_0/init/sample_resource_files/srvbuild.backup_server_2.rs

Building Backup Server 'SYBASERAYS_SERVER_2_BS':

Writing entry into directory services...

Directory services entry complete.

Writing RUN_SYBASERAYS_SERVER_2_BS file...

RUN_SYBASERAYS_SERVER_2_BS file complete.

Starting server...

Server started.

Server 'SYBASERAYS_SERVER_2_BS' was successfully created.

FYI: /opt/sybase/ASE-15_0/init/sample_resource_files/srvbuild.backup_server_2.rs

srvbuild.release_directory: /opt/sybase

srvbuild.product: bsrv

srvbuild.server_name: SYBASERAYS_SERVER_2_BS

srvbuild.new_config: yes

srvbuild.do_add_backup_server: yes

srvbuild.do_upgrade: no

srvbuild.network_protocol_list: tcp

srvbuild.network_hostname_list: localhost

srvbuild.network_port_list: 5001

srvbuild.language: USE_DEFAULT

srvbuild.character_set: USE_DEFAULT

srvbuild.tape_config_file: USE_DEFAULT

srvbuild.errorlog: /opt/sybase/ASE-15_0/install/SYBASERAYS_SERVER_2_BS.log

srvbuild.addl_cmdline_parameters:

srvbuild.related_sqlsrvr: SYBASERAYS_SERVER_2

srvbuild.sa_login: sa

srvbuild.sa_password: sybaserays

Process End:

==========