# Q2.13: How to Shrink a Database

**Warning: This document has not been reviewed. Treat it as alpha-test quality information and report any problems and suggestions to bret@sybase.com**

It has historically been difficult to shrink any database except tempdb (because it is created fresh every boot time). The two methods commonly used have been:

1. Ensure that you have scripts for all your objects (some tools like SA Companion or DB Artisian can create scripts from an existing database), then bcp out your data, drop the database, recreate it smaller, run your scripts, and bcp in your data.
2. Use a third-party tool such as DataTool's SQL Backtrack, which in essence automates the first process.

This technote outlines a third possibility that can work in most cases.

### An Unsupported Method to Shrink a Database

This process is fairly trivial in some cases, such as removing a recently added fragment or trimming a database that has a log fragment as its final allocation, but can also be much more complicated or time consuming than the script and bcp method.

#### General Outline

The general outline of how to do it is:

1. Make a backup of the current database
2. Migrate data from sysusages fragments with high lstart values to fragments with low lstart values.
3. Edit sysusages to remove high lstart fragments that no longer have data allocations.
4. Reboot sql server.

#### Details

1. Dump your database. If anything goes wrong, you will need to recover from this backup!
2. Decide how many megabytes of space you wish to remove from your database.
3. Examine sysusages for the database. You will be shrinking the database by removing the fragments with the highest lstart values. If the current fragments are not of appropriate sizes, you may need to drop the database, recreate it so there are more fragments, and reload the dump.

A trivial case: An example of a time when you can easily shrink a database is if you have just altered it and are sure there has been no activity on the new fragment. In this case, you can directly delete the last row in sysusages for the db (this row was just added by alter db) and reboot the server and it should come up cleanly.

1. Change the segmaps of the fragments you plan to remove to 0. This will prevent future data allocations to these fragments.

Note: If any of the fragments you are using have user defined segments on them, drop those segments before doing this.

sp\_configure "allow updates", 1

go

reconfigure with override

go

update sysusages set segmap = 0

where dbid = and lstart =

go

dbcc dbrepair(, remap)

go

Ensure that there is at least one data (segmap 3) and one log (segmap 4) fragment, or one mixed (segmap 7) fragment.

If the server has been in use for some time, you can shrink it by deleting rows from sysusages for the db, last rows first , after making sure that no objects have any allocations on the usages.

1. Determine which objects are on the fragments you plan to remove.
2. dbcc traceon(3604)
3. go
4. dbcc usedextents( dbid,0,0,1)
5. go

Find the extent with the same value as the lstart of the first fragment you plan to drop. You need to migrate every object appearing from this point on in the output.

1. Migrate these objects onto earlier fragments in the database.

Objids other than 0 or 99 are objects that you must migrate or drop. You can migrate a user table by building a new clustered index on the table (since the segmap was changed, the new allocations will not go on this fragment).

You can migrate some system tables (but not all) using the sp\_fixindex command to rebuild it's clustered index. However, there are a few system tables that cannot have their clustered indexes rebuilt, and if they have any allocations on the usage, you are out of luck.

If the objid is 8, then it is the log. You can migrate the log by ensuring that another usage has a log segment (segmap 4 or 7). Do enough activity on the database to fill an extents worth of log pages, then checkpoint and dump tran.

Once you have moved all the objects, delete the row from sysusages and reboot the server.

Run dbcc checkdb and dbcc checkalloc on the database to be sure you are ok, then dump the database again.