

Copying a Controller SQL database to a different SQL server

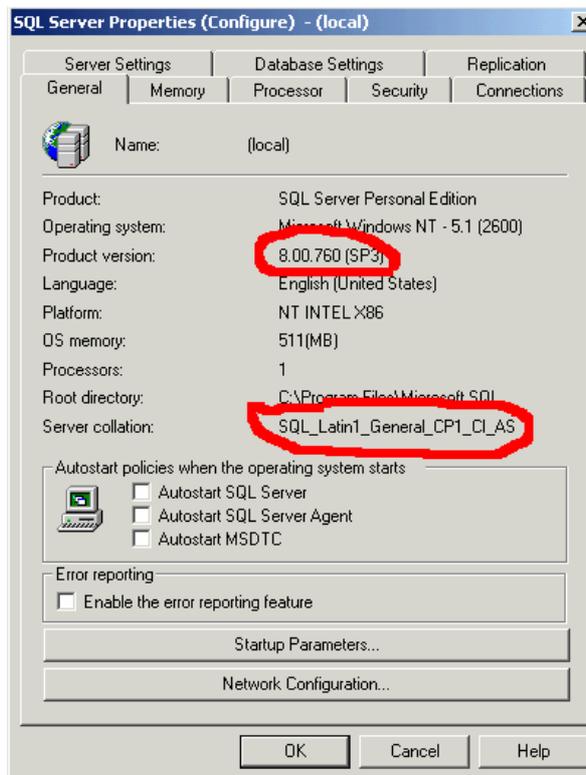
Overview

This document is designed to describe how to create a copy of an existing Controller database, which is currently stored in a SQL 2000 server, then restore it onto a *different* SQL 2000 server.

PLEASE NOTE!

Although Controller can use (virtually) any SQL collation setting, if you transfer databases from one SQL server to another SQL server, you *must* ensure that both SQL servers (i.e. both the “source” and the “destination” SQL servers) have the same “*default collation*” setting.

This setting can be found by launch SQL Enterprise manager, right-clicking on the appropriate server and choosing “properties” (e.g. see below):



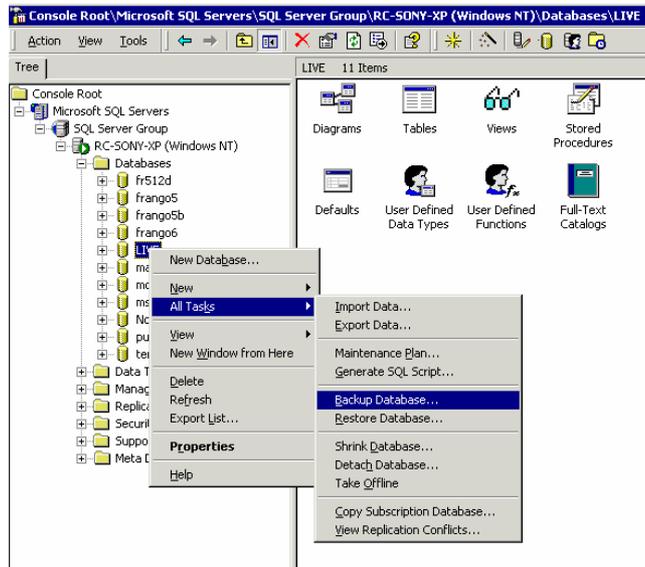
In the above case, the collation setting is set to Cognos’ “preferred” (for ease of support reasons) default setting of “SQL_Latin1_General_CI_AS” (a.k.a. “SQL 1252, dictionary order, case insensitive”).

Details

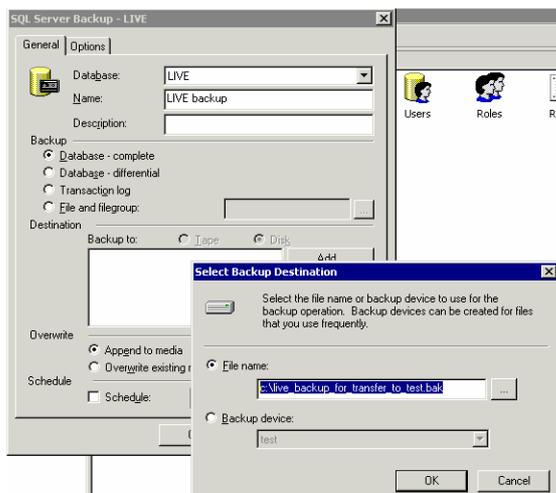
- First of all we will create a SQL backup file on the *source* server, which will contain a copy of the current database.
- Then we will restore this backup “*.BAK” file on the *destination* SQL server.
- Finally, we will ‘synchronise the SQL logins’, which is necessary due to SQL 2000’s fundamental design.

1. Backup of source database

- Ideally, the following should be done when no users are logged onto the Controller system¹
- Launch SQL Enterprise Manager
- Expand the console (click the '+' signs) until you locate the databases
- Right-click on the appropriate Controller database, and choose “all tasks” – “Backup database” (see below)



- Look inside the “Destination” box (approx in middle of the screen). This gives the filename of where the database will be backed up to. If there is already a file/path inside here, then (by default) this file will be appended. *To reduce confusion, it is recommended that you “remove” any existing file details from here, and replace it with a unique file e.g. “C:\backup_controller_22_11_2004.BAK”*
- Click “Add” and type in the filename that you wish to use (see below)



e.g. make a 'complete' backup to the file:
c:\live_backup_for_transfer_to_test.bak

¹ We have not seen a problem caused by creating a backup when users are in the system, but it is a wise precaution nevertheless.

- After clicking OK and OK you should receive the following success message:



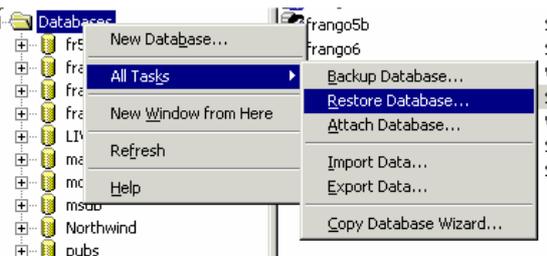
Copy this file to your destination SQL server

NOTE: SQL "*.BAK" files are quite easily compressible. You may therefore find it very useful to ZIP up this database backup file, before transferring it.

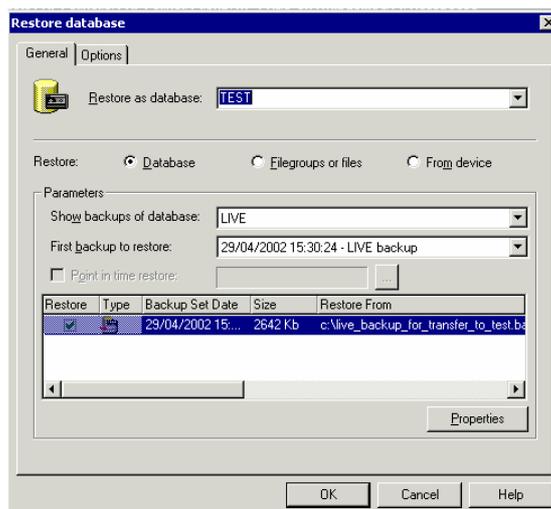
2. Restoration of database onto destination server

First of all, you need to ensure that the correct SQL login ID exists on the *destination* server. This will almost certainly be the case. However, to check:

- launch SQL Enterprise Manager
- Navigate to “security” and then expand to select “logins”. There should be a controller SQL user here (default name is “fastnet”). This user should have “Bulk Insert Administrators” server role ONLY².
- Now right-click on ‘databases’ and select ‘all tasks’ then ‘restore database’:



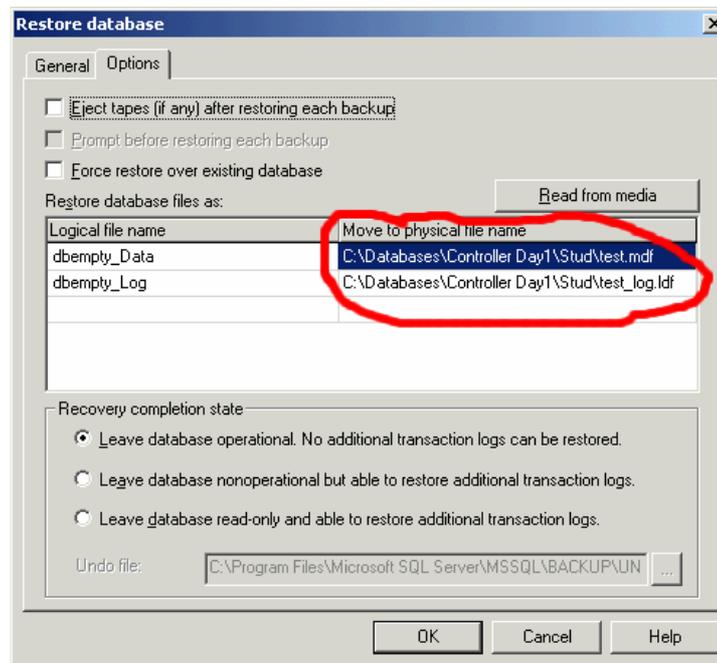
- First of all, decide on the name of the restored database (e.g. “test”) and type this into the top of the screen inside “Restore as database”



- Click “from device”
- Click “Select devices” then “add”
- Browse to the “*.BAK” file that you created in step (1) and add this to the “choose restore devices”
- Click “OK”

² E.g. if has system administrators role, then Controller will not work – get error messages on logon.

- Click on the “options” tab. Here we decide where the data/log files will reside on the destination server. Ideally, for performance reasons, the data (MDF) and log (LDF) files should be on separate disks/spindles/arrays. However, for many situations (e.g. test server) they will be on the same drive. You must choose a location whose folder structures already exist. Therefore, either modify the destination locations, or create some folders to house the MDF/LDF files (see below)

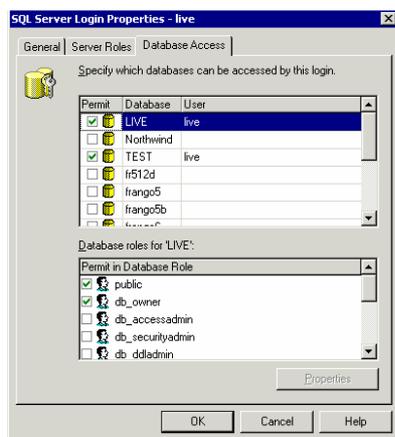


- Click “OK” then “OK” to begin the restoration procedure.

3. Synchronising SQL Logins

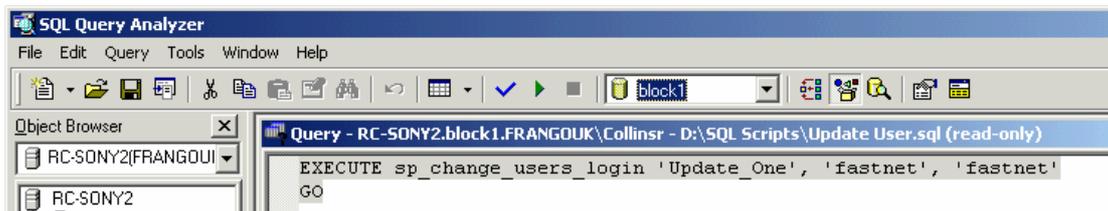
If you have restored a Controller database onto a SQL server which is *not* the same as from where it was backed up, then you need to ‘synchronise’ the SQL logins³. If you restored it to the same server (but with the same or different database name) then you can ignore this step.

- launch SQL Enterprise Manager on the destination server
- Expand “security” then click “logons”
- Double-click on user “fastnet”.
- Click on “database access”. Click on the database that you have just restored, to make it go blue. You should now see something similar to the following:



Ensure that “public” and “db_owner” are the only database roles selected

- Click “OK”. You will receive an error “User fastnet already exists”.
- Acknowledge this with “OK” and then click on the “cross” symbol at the top-right of the window – this is the only way to get out of this screen
- Now launch SQL Query Analyser (“Tools” – “SQL Query Analyser”)
- “File” – “open” and then browse to the file “Update User.SQL”⁴
- Change the database selected to the one that you’ve just restored
 - e.g. in the example below, I’ve selected “block1”



- Click the green “play” button to the left of the database name, to “execute script”

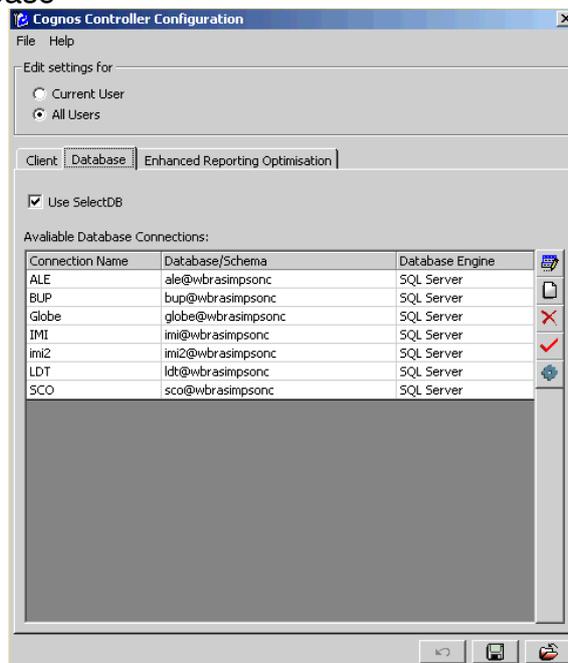
³ This is due to the way that SQL interprets SQL logins in a similar way to how NT interprets userIDs. With NT, each userID has a unique “SID”. By deleting a username in NT (e.g. JSMITH), and then recreating the same userID (“JSMITH”) you are NOT creating the same user (different SIDs!).

⁴ If you do not have this script, then you can create your own file. Just create a new TXT document (e.g. with NOTEPAD), rename it as “Update User.SQL” and paste the following into it:
`EXECUTE sp_change_users_login 'Update_One', 'fastnet', 'fastnet'`
`GO`

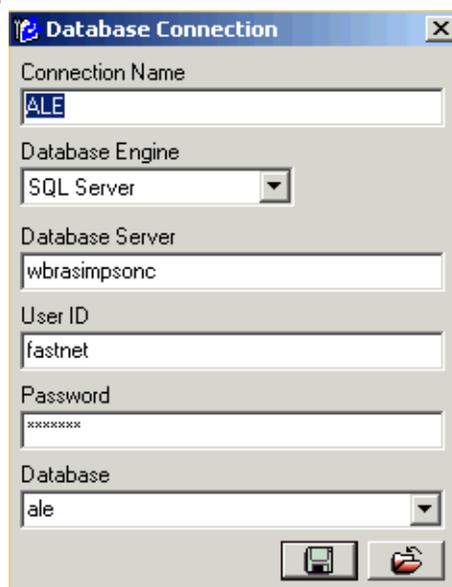
- You will get a message saying “The command(s) completed successfully”.

You are now ready to connect to this new database.

- Logon to the “Application Server”
- Launch START – PROGRAMS – Cognos Controller - Cognos Controller Configuration
- Click “database”



- Create new (or modify the existing) relevant connection(s) so that it/they point to this new database/server
 - your default connection MUST BE called “frango”. This is the one that you would typically modify on a production system.
- If you double-click on an existing connection, a screen similar to the following will appear:



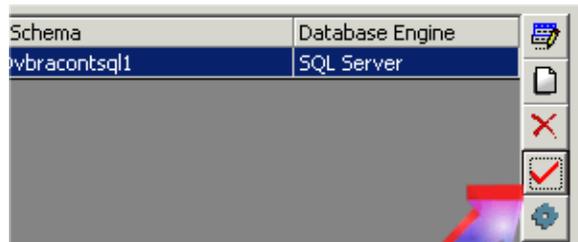
- You can modify the databases server name as appropriate, and then select the appropriate database name, using the drop-down box.

NOTE:

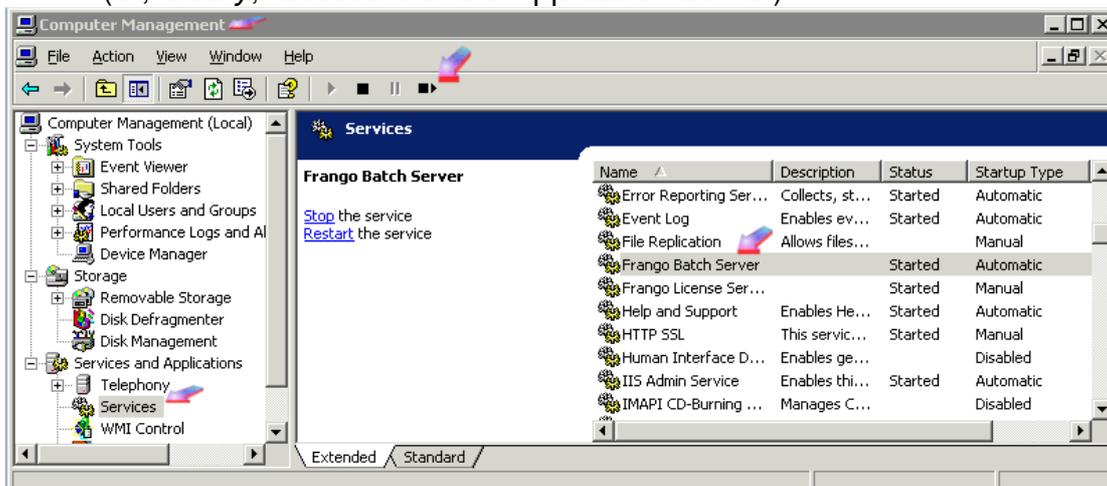
- The “connection name” does *not* have to be the same as the “database name”, but naturally (to save confusion) you should use something logical
- However, issues have been seen where the connection name is not “straightforward”. We therefore recommend:
 - Do not use any “strange” characters inside your connection name (e.g. “%” or “£” or “#” etc.)
 - ONLY use lowercase letters, and not a mixture either
 - e.g. call the connection name “test” but not “TEST” or “Test”
 - Ideally, do not use numbers (e.g. “2” or “6”) either, although this is *normally* not a problem

IMPORTANT:

- Once you’ve created all the relevant entries, you must ensure that the relevant connection(s) are all valid:
 - highlight each connection (in turn), to make it go “blue”
 - click on the red “tick” to check the “test database connection”



- If it asks you to “update ODBC/ADO... settings” then please agree to do so
- Finally, for the application server to allow consolidations to run on this new database, you must restart the service “Frango Batch Server” (or, *ideally*, reboot the entire Application Server):



[Incidentally, you *may* have a need to configure the “Enhanced Reporting Optimisation” tab. However, for most customers, they do not use this. See separate document “Configuring ERO.pdf” – which is part of the install CD - for more information on how to change these settings]

You can now launch the Controller program to test!