IBM
Spectrum
Protect

SQL Backup & Restore guide

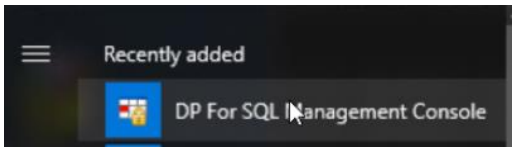
Written by: Bjarke.Ravn-Jonsen@team.blue

Date: 06/02 2020

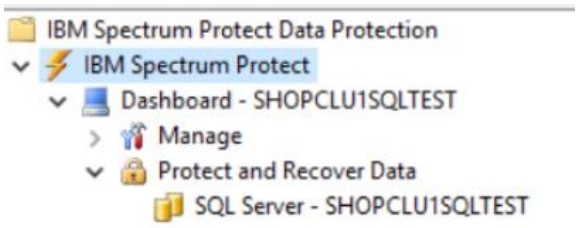
Contents

GUI Backup (from Flashcopy Manager / DP for SQL)

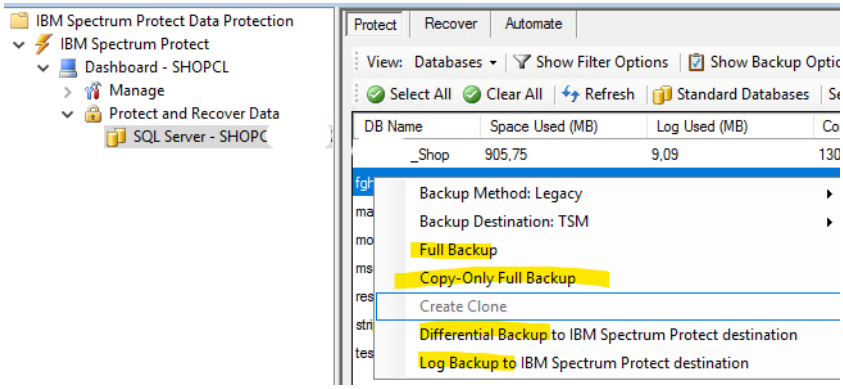
Launch, "DP for SQL Management Console" from the start menu



Expand down, and locate the SQL instance:



Select the database, and right click it:



Select one of the options:

Full Backup

Copy-Only Full Backup

Or *Differential or log backup*

Log backup will only be available if the database is running with a full recovery model.

Command line backup

If needed backup can also be performed the command line. The default location of the client is:

C:\program files\tivoli\tsm\tdpsql

The syntax is fairly simple:

C:\Program Files\tivoli\tsm\TDPSql>tdpsqlc backup <name of the database> <type of backup>

So, it could be:

C:\Program Files\tivoli\tsm\TDPSql>tdpsqlc backup master full

The name of the database could also be replaced by a "*" if all databases should included:

*tdpsqlc backup * log*

If only a few, they should be listed with a comma between:

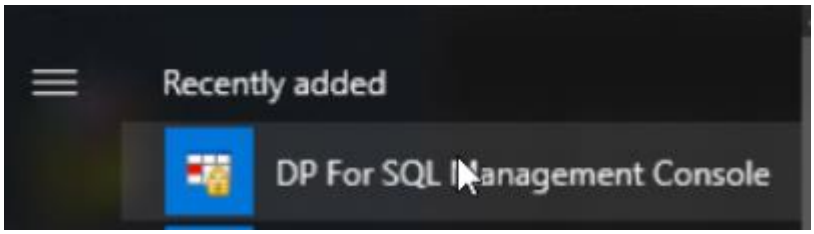
tdpsqlc backup dataBase1,database2,database3 full

If more than one SQL instance is installed, it can be targeted with: /sqlserver=

*tdpsqlc backup * log /sqlserver=Sqlserver\SQL2017*

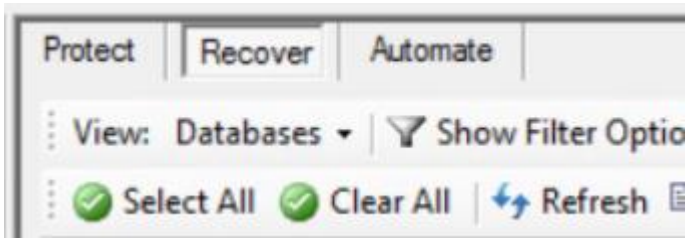
GUI restore (from Flashcopy Manager / DP for SQL)

Launch, "DP for SQL Management Console" from the start menu



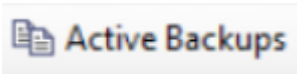
Expand down, and locate the SQL instance:

Chose the tab, "recover":



The default view, will list the active backup of all databases. This will include one FULL backup, and any log and differential backups.

If an older backup is needed for restore, select the button "Active Backups"



and it will now list all backup, including older.

A restore can, (generally speaking) be performed in two ways: overwrite the existing database, or restore to a new name.

Both methods can be of active data, or older inactive data.

Restore and overwrite:

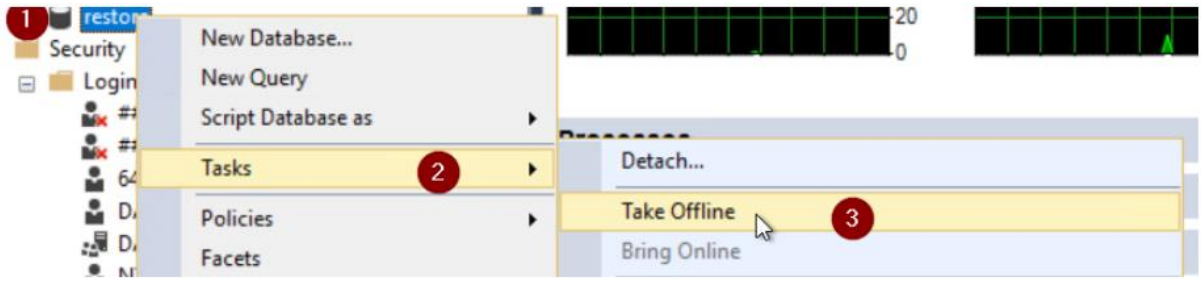
Start by selecting the button "Show Restore Options"

and make sure the "Replace" option is set to yes:

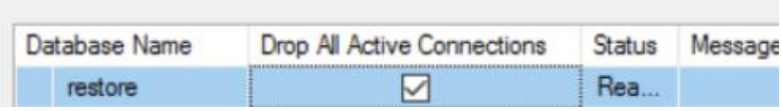
The "Run Database Recovery" option must also be set to yes, in order for the restored database to go online.

In some case, it's necessary to set the database offline before the restore can proceed, often due to active connections.

If so, right click on the database in the Management Studio, select Tasks → Take Offline:



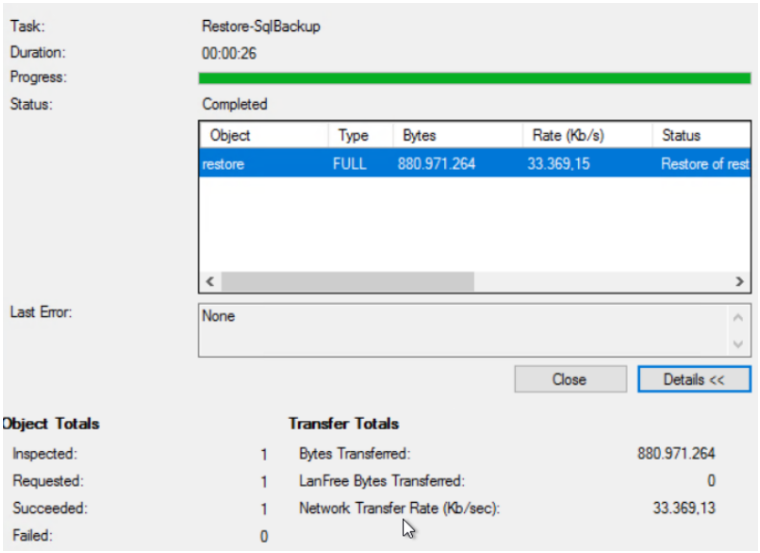
And make sure the "drop all active connections" is set:



Now right click on the object in the GUI, and select restore.

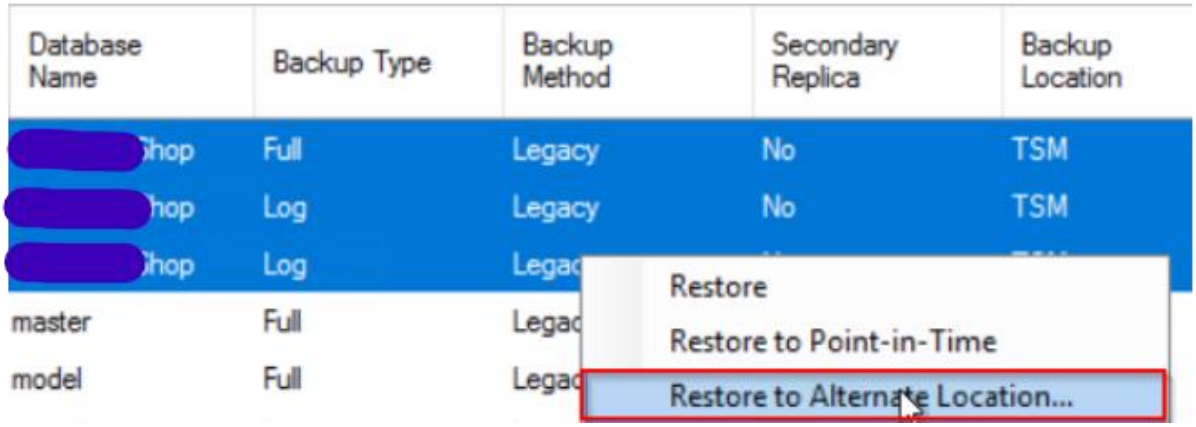
A new task can now be seen in the

And when it's done, it will show up as completed:

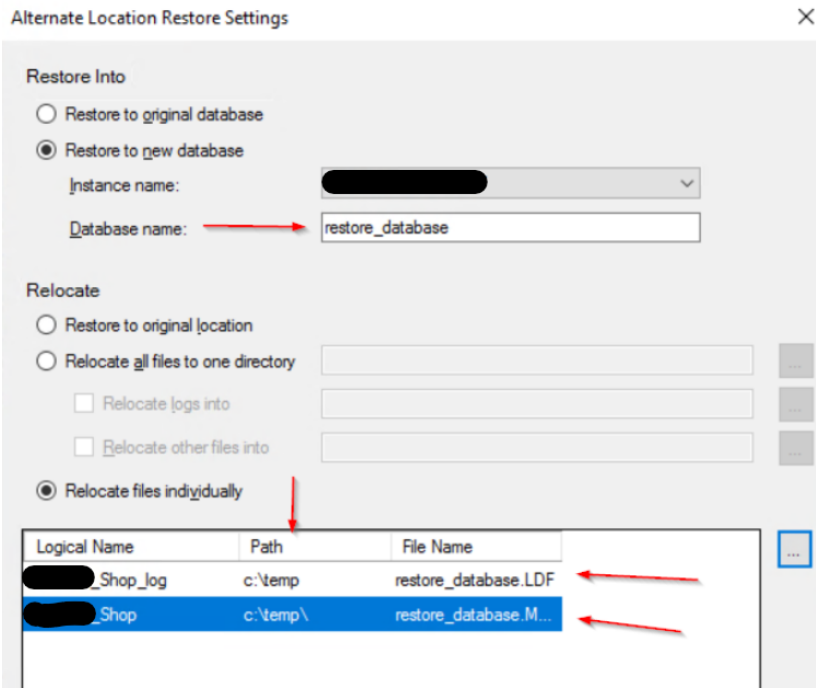


Restore and relocate

Right click the database and select "restore to Alternate Location":



Fill in with a "Database name" this should be a new name that is not being used on the SQL instances.

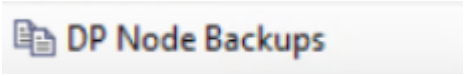


A new path can be set (but not required) and new filenames -these are required!

Other helpful options

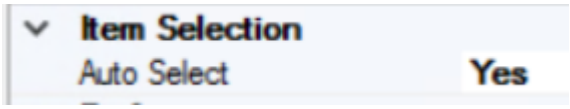
If backup has been performed on a SQL Always On setup, the AO databases will not be listed in the default view.

Click "DP Node Backups"



To view the databases.

It can be helpful to have "Auto Select" set to Yes, in the Restore Option section:



This will make sure to select all the needed objects, in case of a restore from several logs and a full.

Command line restore

In some cases, it can be helpful to use the command line client "tdpsqlc.exe" instead of the Flashcopy Manager GUI.

It could be due to many databases causing the GUI to stall, or if the GUI has not been installed correctly (often in older setup, missing PowerShell and so.

The default path for the client is: *C:\Program Files\Tivoli\tsm\tdpsql*

Query the backup

Before the restore can be initiated, you must know what objects to retrieve.

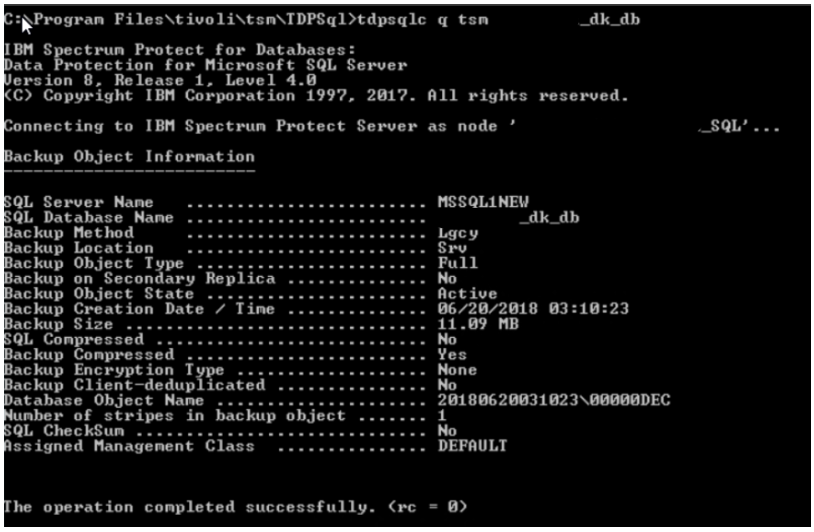
To get a list of all (active) objects run:

*tdpsqlc q tsm **

the list can be reduced by inserting the database name:

tdpsqlc q tsm <name_of_db> ***(Notice, the name is case sensitive!!)***

It will return something like:



By default, only the active version is listed. To get all use */all* this will also include log objects and diff (if present)

tdpsqlc q tsm <name of db> /all

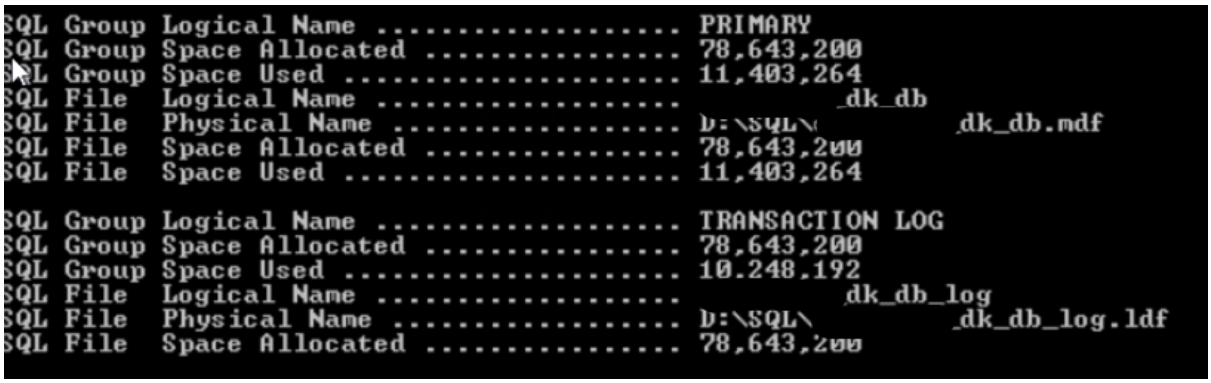
In many cases it will make sense to pipe the list to a fill, so it can be searched:

tdpsqlc q tsm <name of db> /all > c:\temp\sql_backup.txt

In some cases, it can be helpful to know the location (when the backup was made) of the files and names. Use:

tdpsqlc q tsm <name of db> full /fileinfo

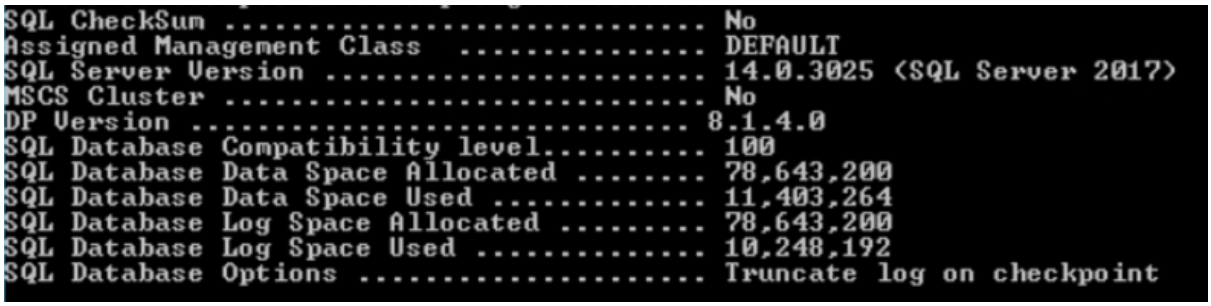
It will provide a list like:



In a case of master database recovery, it will be necessary to know the SQL version the backup was made from.

Use */compat* to get the information:

tdpsqlc q tsm <databasename> full /compat



Restore and overwrite

As a rule of thumb, all sql database restores must be initiated with a full backup and then one or more log backups (if needed)

Logs cannot be restored alone, however a full can be restored without logs.

In our case the file restored was made from 05-06-2018 (notice: this is day/month/year, when viewed in the file restore gui!)

The SQL output will be like:

Backup Creation Date / Time ..... 06/08/2018 so this is Month/day/year!)

Open the text file that the query was output' to and find a full object with the correct data:

```
Backup Object Information
-------------------------

SQL Server Name   ..... (
SQL Database Name ..... SHOP
Backup Method     ..... Lgcy
Backup Location   ..... Srv
Backup Object Type ..... Full
Backup on Secondary Replica ..... No
Backup Object State ..... Inactive
Backup Creation Date / Time ..... 06/05/2018 03:53:19
Backup Size ..... 232.14 MB
SQL Compressed ..... No
Backup Compressed ..... Yes
Backup Encryption Type ..... None
Backup Client-deduplicated ..... Yes
Database Object Name ..... 20180605035319\000028BC
Number of stripes in backup object ..... 1
SQL CheckSum ..... No
Assigned Management Class  ..... DEFAULT
```

1

It’s important to notice the "Database Object Name"

As this object is from around 03:50 in the morning, we want to apply logs to get further ahead:

log 1:

```
Backup Object Information
-------------------------

SQL Server Name   ..................... SHOP
SQL Database Name ..................... SHOP
Backup Method     ..................... Lgcy
Backup Location   ..................... Srv
Backup Object Type .................... Log
Backup on Secondary Replica ........... No
Backup Object State ................... Inactive
Backup Creation Date / Time ........... 06/05/2018 07:37:52
Backup Size ........................... 2.45 MB
SQL Compressed ........................ No
Backup Compressed ..................... Yes
Backup Encryption Type ................ None
Backup Client-deduplicated ............ Yes
Database Object Name .................. 20180605073752\000029DC
Number of stripes in backup object ..... 1
SQL CheckSum .......................... No
Assigned Management Class  ............ DEFAULT
```

Log2:

```
Backup Object Information
-------------------------

SQL Server Name   ..................... SHOP
SQL Database Name ..................... SHOP
Backup Method     ..................... Lgcy
Backup Location   ..................... Srv
Backup Object Type .................... Log
Backup on Secondary Replica ........... No
Backup Object State ................... Inactive
Backup Creation Date / Time ........... 06/05/2018 09:36:56
Backup Size ........................... 719.50 KB
SQL Compressed ........................ No
Backup Compressed ..................... Yes
Backup Encryption Type ................ None
Backup Client-deduplicated ............ Yes
Database Object Name .................. 20180605093656\00001DC0
Number of stripes in backup object ..... 1
SQL CheckSum .......................... No
Assigned Management Class  ............ DEFAULT
```

We now have found a selection of 3 objects: 1 full and 2 logs that would bring the database to a point in time at 09:36 the 5/6 (day/month)

We now need to construct 3 restore commands, and the rule is:

All but the last, should have /recovery=no, the last, no matter how many, should have /recovery=yes

The *Object name* from the text file must be used in each line and the database name as well.

In our case, this translate into: (run them one at a time)

tdpsqlc restore SHOP_xxxxxx /object=20180605035319\000028BC /replace /recovery=no

tdpsqlc restore SHOP_xxxxxx log= /object=20180605073752\000029DC /recovery=no*

tdpsqlc restore SHOP_xxxxxx log= /object=20180605093656\00001DC0 /recovery=yes*

If only one restored is needed, it will be the full, and it has to have /recovery=yes.

There is not limit on how many logs that can be restored and applied, but only 1 full.

Remember, the command line is case sensitive regarding the name of the database!

Relocation of database and files

In some cases, we don't want to replace the existing database, but rather have a copy.

tdpsql restore <name_of_db> /object=<objectID> /into=<new_database_name> /relocate=<logical_name_db>,<logical_name_db_log> /to=<path_to_file.mdf>,<path_to_log.ldf> /recovery=<yes>/<no>

It could be translated into:

tdpsqlc restore database-Prod full /object=20180530000852\00000CDC /into=Database_RESTORE /relocate=database-prod,database-prod_log /to=c:\temp\database-prod_restore.mdf,c:\temp\database-prod_log_restore.ldf /recovery=yes

Other useful settings could be:

if the backup has been made by more than one instance, locate the correct one with:

/fromsqlserver=<name_of_sql_instance>

if more than one sql instance is present, the target can be set like:

/sqlserver=<name_of_sql_instance>