

Database Backup/Import/Export Guide V1.1

Oracle 10g/11g and MSSQL Server 2014

**Table of Contents**

Introduction………………………………………………………………………….………………3

[Oracle – Database Backup 3](#_bookmark6)

[Oracle – Database Export 3](#_bookmark6)

[Oracle - Database Restore 4](#_bookmark6)

[SQL Server – Database Backup 6](#_bookmark7)

[SQL Server – Database Restore 1](#_bookmark7)1

|  |  |
| --- | --- |
| Version | Changes |
| v1 thru 3.1 | Creation of Document |
| V1.0 | Version 1 of Oracle MSSQL database backup/export/import |
|  |  |
| V1.1 | Revise document and update Screenshots to MSSQL 2014 |
|  |  |

# 

# Introduction

# This document is intended to provide general guidance for database actions in context of working with Teamcenter. Always consult your local Database Administrator before attempting any steps.

It is recommended to backup and restore during production downtime where there are no active connections to the db or users logged into a Teamcenter client. Active and/or idle sessions can cause locks in the database. These will need to be removed by running the utility ‘Clearlocks –assert\_all\_dead’ from a tc cmd shell before beginning the process.

# Oracle Database Backup

Set the directory parameter for exp\_dir:

Log into the OS as the Oracle installing user and run the two commands below from SQLPlus:

**SQL> CREATE OR REPLACE DIRECTORY exp\_dir AS ‘C:\some\path’;**

**SQL> GRANT READ, WRITE ON DIRECTORY exp\_dir TO infodba;**

**NOTE:** If your db user isn't infodba then replace infodba with that account name here as well as in all the syntax for expdp and impdp referenced here

**NOTE**: In the syntax, replace %SID% with the literal string for your database SID~default is tc

# Oracle Database Export

# Prior to exporting the db, it is recommended to perform the following to extract the current database sting:

# SQL> SELECT count(\*) FROM PPOM\_OBJECT;

# Store this value or image for reference to compare future results to ensure no modifications to the db occurred between the steps.

At this point, exit the SQLplus utility but stay in the shell or open a new one. The next command will be to export the database to a flat file, called a ‘dump’ file. There are two different methods available to accomplish this.

1. EXP

Specify all valid parameters and their values from the command line using the following syntax

**exp infodba/infodba@%SID% full=Y file=PATH/filename.dmp log=PATH/filename.log**

1. Data Pump

If using Oracle 11g or higher, you also have the option of ‘Data Pump’. This feature uses a Direct path method of unloading, making a single stream of data unload approximately 2 times faster than the original Export because Direct path API has been updated to be more efficient.

A good reference for Oracle Data Pump can be found:

<http://www.oracle.com/technetwork/database/enterprise-edition/datapump-faq-082259.html#what_is>

Using EXPDP for Data Pump:

**expdp infodba/infodba@%SID% full=Y directory=exp\_dir dumpfile=%SID%.dmp logfile=%SID%\_dump.log compression=all**

# Post exporting the db, again, perform the following to extract the current database sting:

# SQL> SELECT count(\*) FROM PPOM\_OBJECT;

# Store this value or image for reference to compare future results to ensure no modifications

Creation of the dump file is complete at this point.

## Oracle Database - Restore

1. Drop the current Teamcenter database:

Navigate to the %ORACLE\_HOME%\assistants\dbca\templates folder and you should find a file called tc\_unpopulate\_db.sql. You can call this from a regular command prompt:

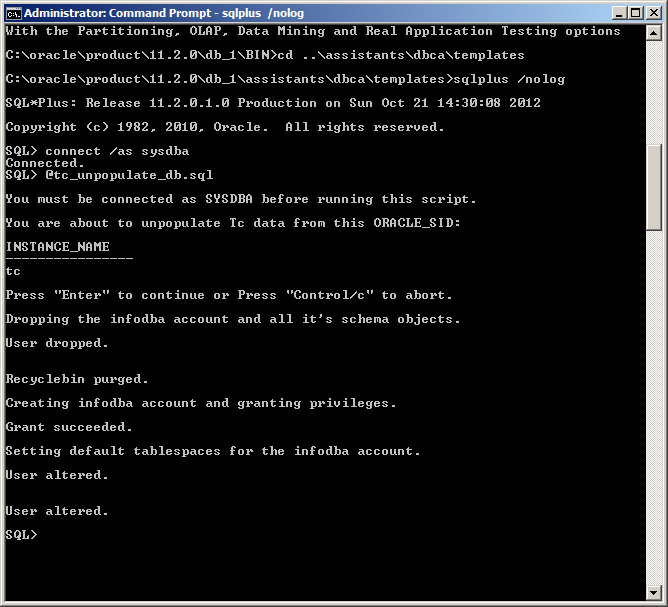
set ORACLE\_HOME=\* set ORACLE\_SID=\*

cd %ORACLE\_HOME%\assistants\dbca\templates sqlplus /nolog

connect /as sysdba

@tc\_unpopulate\_db.s

For example:



1. Oracle Database Import

Use the IMP method if EXP was used to export

**imp infodba/infodba@%SID% full=Y file=PATH/filename.dmp log=PATH/filename.log**

Use IMPDP for Data Pump if EXPDP was used to export

**impdp infodba/infodba@%SID% full=Y directory=exp\_dir dumpfile=%SID%.dmp logfile=%SID%\_dump.log compression=all**

**NOTE:** If you use expdp then you must use impdp. If you use exp then you must use imp. Don’t mix and match the utilities.

**After Import:**

After the dump of the database, perform the following and hang onto the results for comparison:

* **SELECT count(\*) FROM PPOM\_OBJECT;**

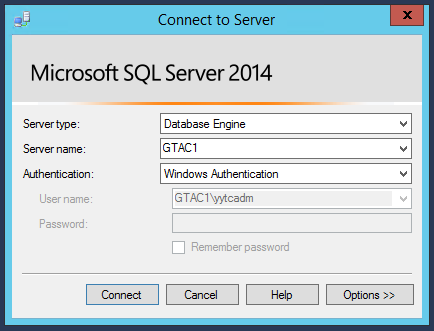
Compare the results from the three select statements. If the value returned for this is the same for all three then continue forward. If not, then re-export and run through the steps again making sure that there are no active connections to the database.

## 

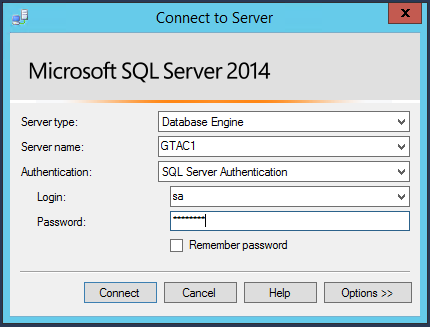
## SQL Server – Database Backup

To begin, login to the SQL Server Management Studio with the local Windows installing account or with the ‘sa’ account

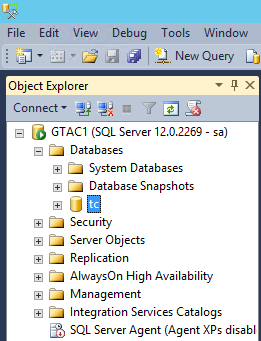
**Login as Windows user**



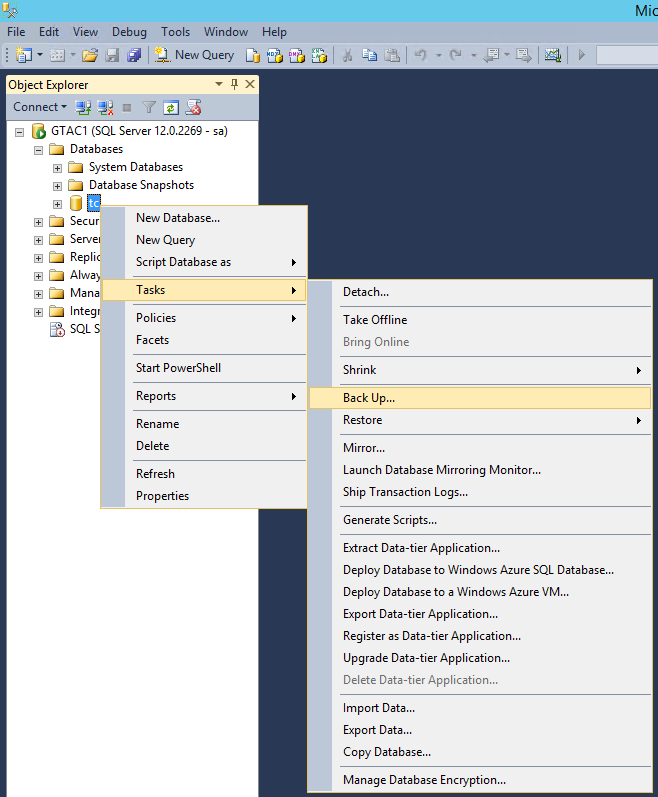
**Or login as ‘sa’**

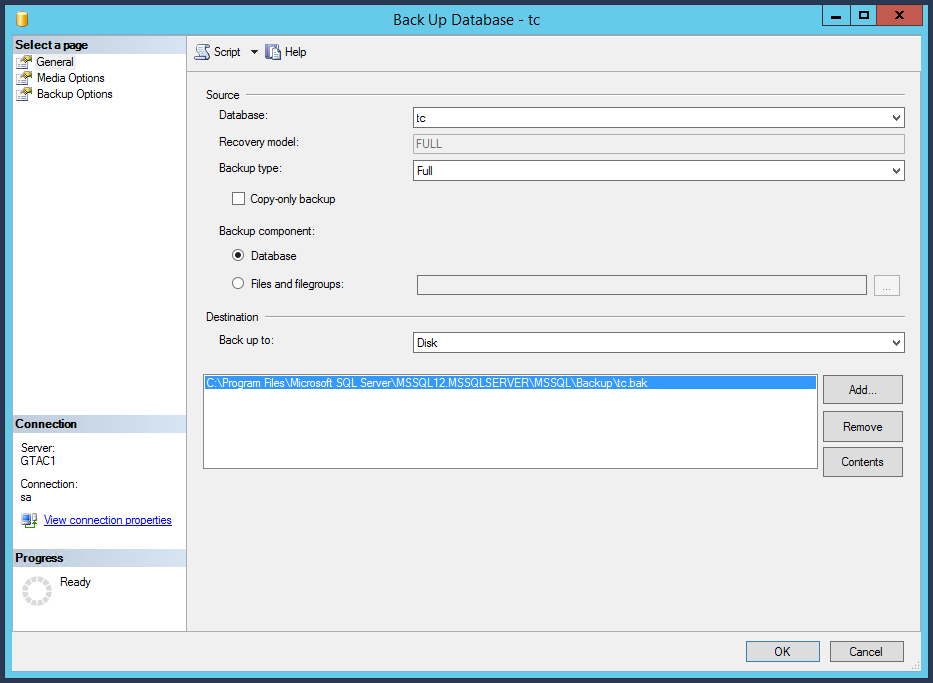


* **find the database in the interface. In this example, the database is ‘tc’**



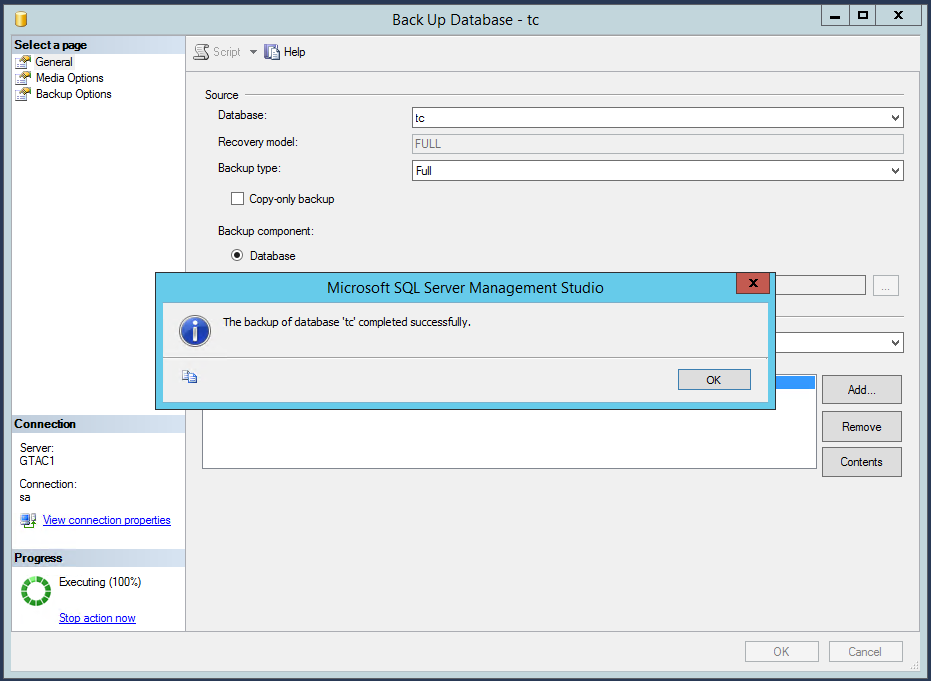
* **right click on the database. And select Tasks 🡪 Back Up…**





* **Verify database options and the “Destination” path. Note the ‘Backup type’ is ‘Full’ to simulate either a full clone or disaster recovery scenario as part of this example.**

Once complete, the dialog reports:

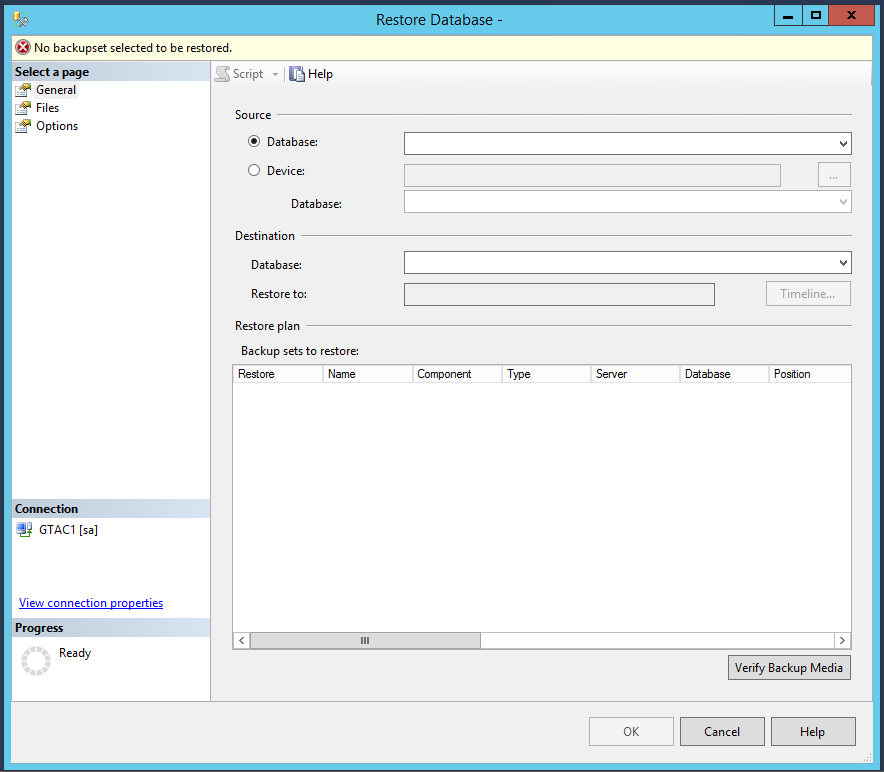


* **The file tc.bak has now been created in the specified Destination. This provides the database component to perform a clone or disaster recovery.**

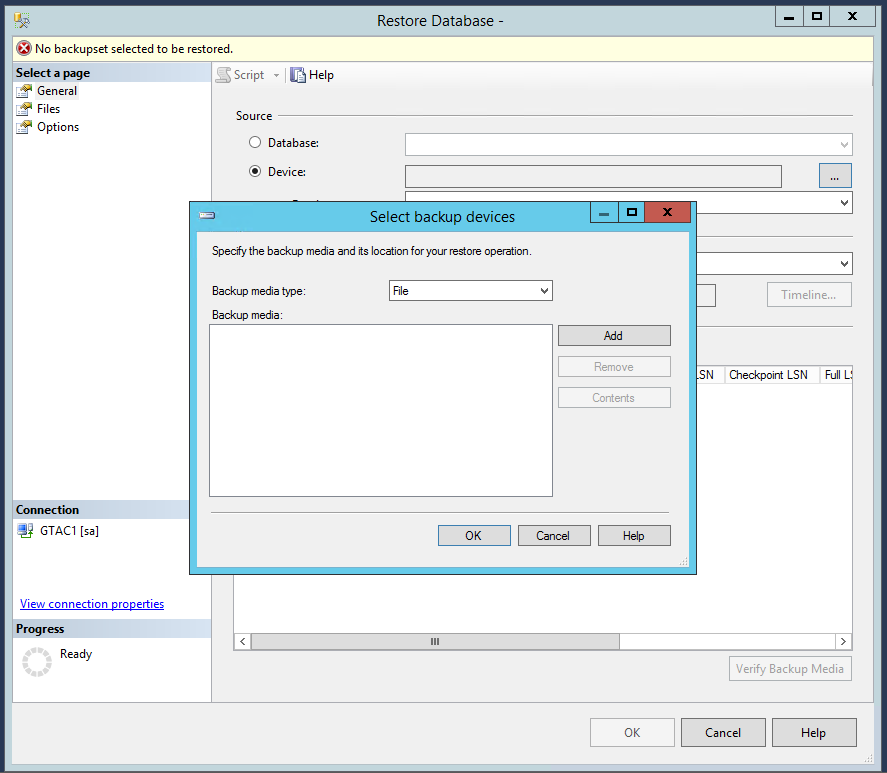
## SQL Server – Database Restore

To perform the restore of the database, make the backup (.bak) file available to the new SQL Server file system. It is also recommended the names of the databases are the same as this makes the process much cleaner. Then, from the SQL Server Management Studio:

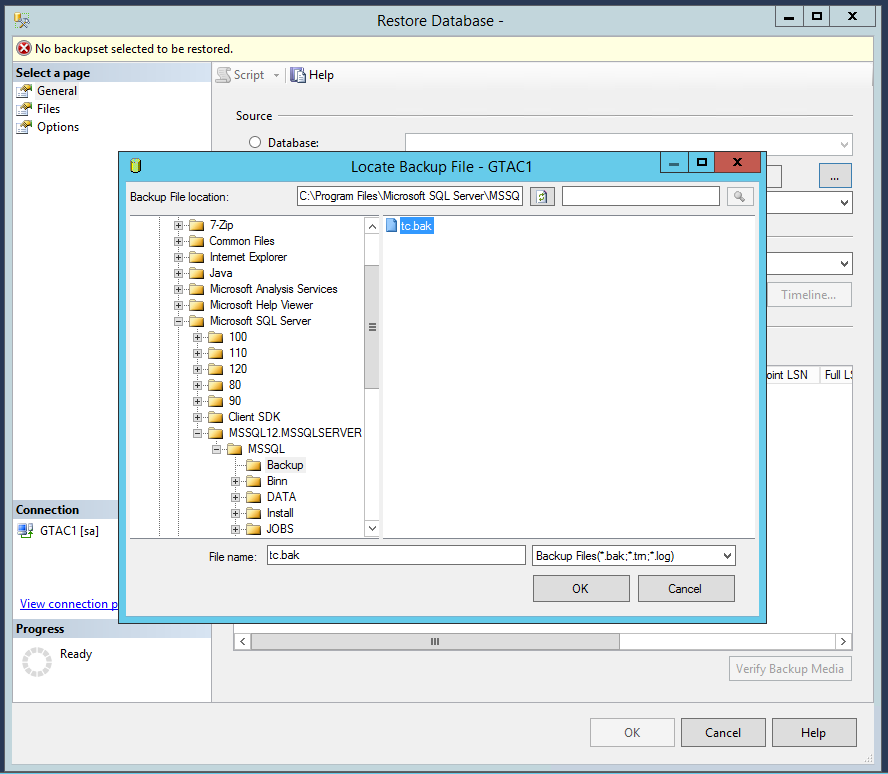
## 

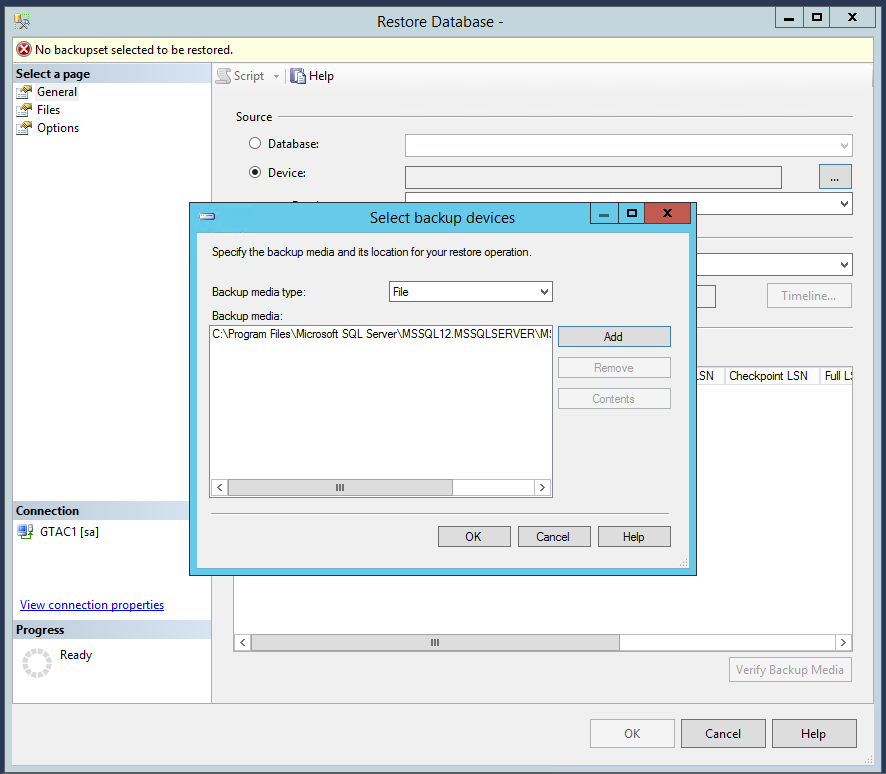


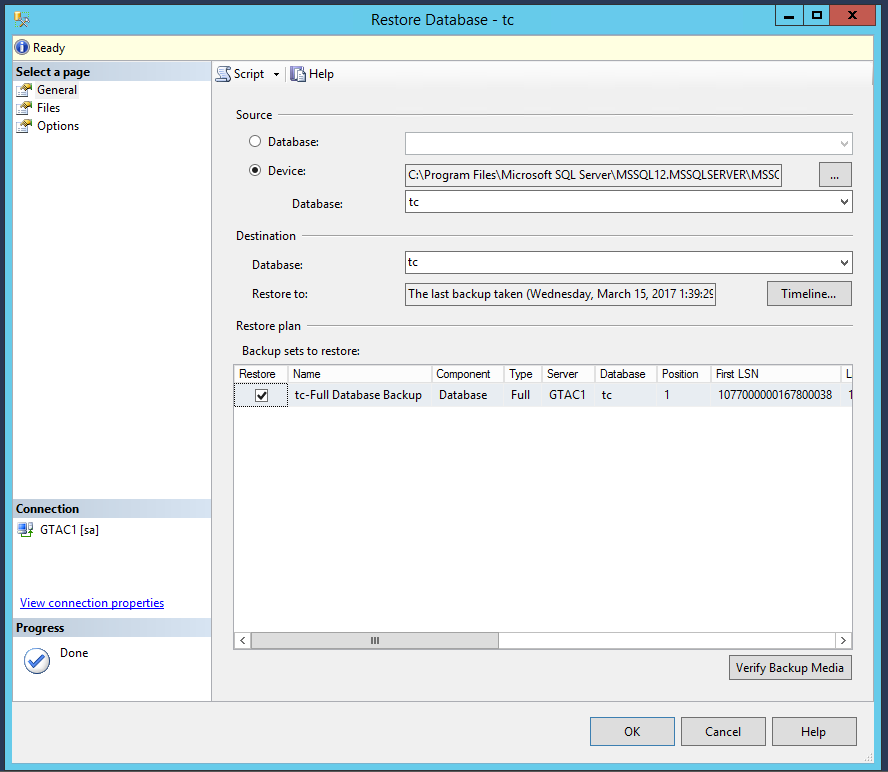
* **By default, “Database” is selected above. To point to the \*.bak file, select “Device” and “…”**



* **“Add” and browse to where you saved the backup.**







* **Verify the settings are correct and ‘OK’**

