# Add a Database to a SQL Server AG Using T-SQL

<https://www.sqlnethub.com/blog/how-to-add-database-to-sql-server-availability-group-using-tsql/>

In some cases, while trying to add a database to a SQL Server availability group using the wizard in SQL Server Management Studio, you might get some warnings or errors. There is a chance that these warnings/errors won’t let you continue and add the database.

For example, a possible warning could be the below:

This database is encrypted by database master key, you need to provide valid password when adding it to the availability group.

So, let’s consider the following scenario: You have two or more AG-enabled database servers, on which there is one (ore more) availability groups. On the current primary replica node, you have a database which you want to add to the existing availability group. However, the wizard does not let you do this because of some conditions such the above warning message.

Don’t worry because there is another way to achieve this. It is a little bit more “manual” as it involves T-SQL, but it is very simple.

The procedure to add a database to a SQL Server availability group, when it is not possible to add it using the wizard is:

### **Step 1. On the primary node, add the database to the availability group**

-- On Primary Node

USE MASTER;

GO

ALTER AVAILABILITY GROUP [AGNAME] ADD DATABASE [DBNAME];

GO

### **Step 2: Perform a full backup of the database on the primary node/replica**

You can easily do this using the [SSMS database backup wizard](https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/create-a-full-database-backup-sql-server?view=sql-server-2017) or with a T-SQL script.

Here’s a simple example below:

BACKUP DATABASE [DBName] TO DISK = N'Path\_to\_Store\_Backup\_File\BackupFileName.bak';

GO

### **Step 3: Restore the full database backup on the secondary node along with specifying the “WITH NORECOVERY” option**

The [NORECOVERY](https://docs.microsoft.com/en-us/sql/t-sql/statements/restore-statements-transact-sql?view=sql-server-2017) option specifies that a rollback of the database will not take place, thus allowing to restore even more records.

***Note:****You must be very careful when restoring databases. For the context of this article, first you need to make sure that the same database does not exist on the secondary node, prior to run the above-mentioned restore operation.*

You can restore the database using the [SSMS restore database wizard](https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/restore-a-database-backup-using-ssms?view=sql-server-2017), along with specifying the NORECOVERY option or use a T-SQL statement.

Example:

RESTORE DATABASE [DBName]

FROM DISK = N'Path\_Database\_Backup\_File\_Stored\BackupFileName.bak'

WITH NORECOVERY;

GO

### **Step 4: Perform a transaction log backup of the database on the primary node/replica**

The next step, is to go back again on the primary node/replica and perform a transaction log backup of the database.

Again, you can do this using the SSMS backup database wizard or using a T-SQL statement.

**Example**:

BACKUP LOG [DBName] TO DISK = N'Path\_to\_Store\_Log\_Backup\_File\LogBackupFileName.bak';

GO

### **Step 5: Restore the log backup on the secondary node along with specifying the option “WITH NORECOVERY”**

The next step, is to restore the log backup on the secondary node, on the previously restored database (see Step 3), again with the “NORECOVERY” option.

If you prefer doing this using T-SQL instead of the SSMS restore database wizard, below you can find an example of how the T-SQL script looks like.

Example:

RESTORE LOG [DBName] FROM DISK = N'Path\_Log\_Backup\_File\_Stored\LogBackupFileName.bak' WITH NORECOVERY;

GO

### **Step 6: On the secondary node, add the database to the availability group by altering it**

The final step, is on the secondary node to add the database to the availability group by using the ALTER command.

Here’s an example of the T-SQL script:

USE MASTER;

GO

ALTER DATABASE [DBNAME] SET HADR AVAILABILITY GROUP = [AGNAME];

GO

### **Step 7: Confirm that all AG replicas are synchronized**

It goes without saying that after all the above, you need to verify that all database replicas in the availability group are synchronized. You can do this via SSMS, by right clicking on the availability group under “Always On High Availability” and selecting “Show Dashboard”. There, you can see if everything is OK. If it is all **green**, and you see the wording “no data loss” then all is good 🙂