**[AlwaysON database NOT SYNCHRONIZED and RECOVERY PENDING](http://www.sqlservercentral.com/blogs/sql-server-blog-forum/2016/08/06/alwayson-database-not-synchronized-and-recovery-pending/)**

Posted on 6 August 2016

FROM: <http://www.sqlservercentral.com/blogs/sql-server-blog-forum/2016/08/06/alwayson-database-not-synchronized-and-recovery-pending/>

**AlwaysON database NOT SYNCHRONIZED and RECOVERY PENDING**

I have recently come across an issue that, one of my alwaysON secondary replica databases are went into “NOT SYNCHRONIZED and RECOVERY PENDING” state. It is a geo cluster alwaysON with 4+2 nodes configured in both synchronous and asynchronous mode.

The issue is when the database removed from the primary replica, with the secondary disconnection the higher database IDs on the secondary went into “NOT SYNCHRONIZED and RECOVERY PENDING” state, but the lower database IDs are good and synchronous state only.

I just checked with one of the Microsoft MSFT and he said it considers as a bug. “Since, it occurs a DDL change on the Availability Group as the primary, while a secondary replica server is down”.

I just reproduced the same. It is a two node synchronous replica server.

Version: Microsoft SQL Server 2012 (SP1) – 11.0.3000.0

I have not tested it on SQL 2014.

**Testing:**

1. Create five databases.

CREATE DATABASE dba

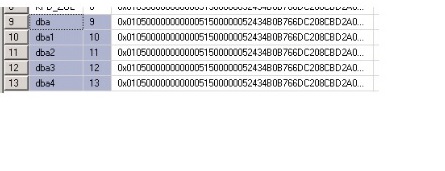
CREATE DATABASE dba1

CREATE DATABASE dba2

CREATE DATABASE dba3

CREATE DATABASE dba4

SELECT \* FROM SYS.SYSDATABASES

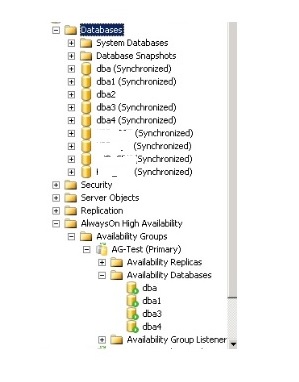
[](http://www.sqlserverblogforum.com/wp-content/uploads/2016/08/AG_Bug/1_DB_Id.jpg)

2. Created a test availability group and added a five databases in synchronous mode.

3. Stop SQL Instance for Secondary Replica.

4. On the Primary Replica issue TSQL to remove a database from availability group.

ALTER AVAILABILITY GROUP [AG-Test] REMOVE DATABASE dba2;

[](http://www.sqlserverblogforum.com/wp-content/uploads/2016/08/AG_Bug/2_DB_removed.jpg)

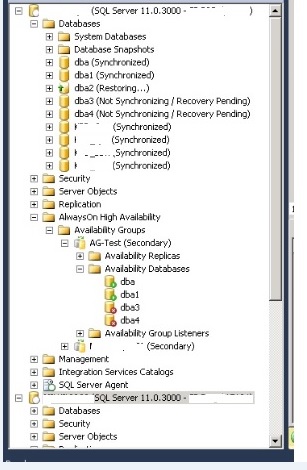
5. Start SQL Instance for the Secondary Replica.

You can see that, databases whose IDs are higher than the one removed dba2 are going to “NOT SYNCHRONIZED / RECOVERY PENDING” mode.

DB ID – 9, 10 for dba,dba1 – Good

DB ID – 11 for dba2 – Removed DB

DB ID – 12,13 for dba3, dba4 went into SYNCHRONIZED / RECOVERY PENDING.

[](http://www.sqlserverblogforum.com/wp-content/uploads/2016/08/AG_Bug/3_Recovery_pending_new.jpg)

SELECT DATABASE\_ID,DB\_NAME(DATABASE\_ID),NAME,REPLICA\_SERVER\_NAME,

SYNCHRONIZATION\_HEALTH\_DESC,SYNCHRONIZATION\_STATE\_DESC,S.\*

FROM SYS.DM\_HADR\_DATABASE\_REPLICA\_STATES S JOIN  SYS.AVAILABILITY\_GROUPS G ON (S.GROUP\_ID=G.GROUP\_ID)

JOIN SYS.AVAILABILITY\_REPLICAS   GR ON (GR.GROUP\_ID=G.GROUP\_ID)

–WHERE SYNCHRONIZATION\_HEALTH\_DESC <>’HEALTHY’

–AND REPLICA\_SERVER\_NAME =”

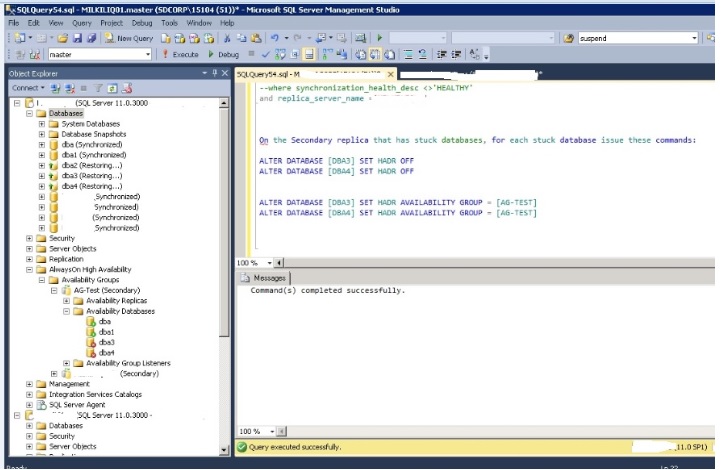
The fix and work around is resume other databases that are stuck in secondary replica.

6. On the Secondary replica that has stuck databases, for each stuck database remove (will make it DB into restoring mode) and add (will make it DB into synchronized mode).

Remove a Secondary Database from an AG –> <https://msdn.microsoft.com/en-us/library/hh231120.aspx>

ALTER DATABASE [DBA3] SET HADR OFF

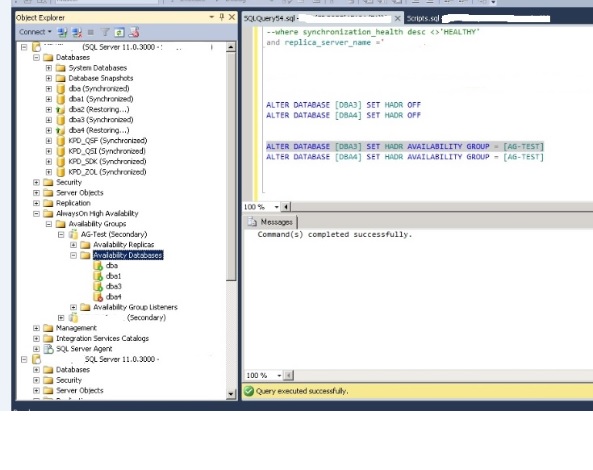
ALTER DATABASE [DBA4] SET HADR OFF

[](http://www.sqlserverblogforum.com/wp-content/uploads/2016/08/AG_Bug/4_HADR_OFF.jpg)

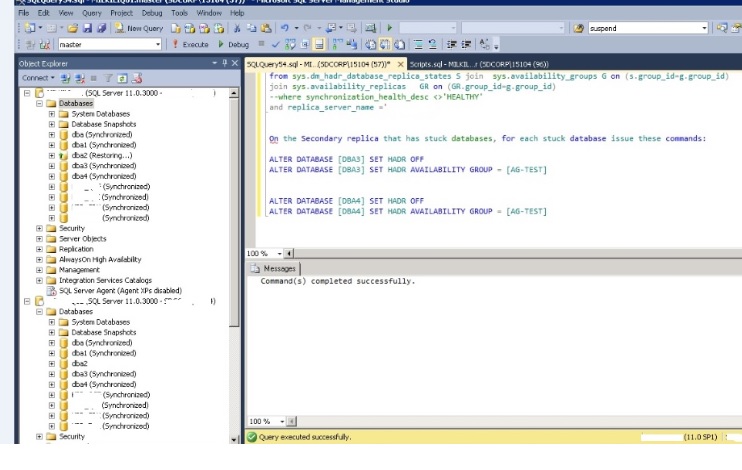
Join a Secondary Database to an AG –> <https://msdn.microsoft.com/en-us/library/ff878535.aspx>

ALTER DATABASE [DBA3] SET HADR AVAILABILITY GROUP = [AG-TEST]

ALTER DATABASE [DBA4] SET HADR AVAILABILITY GROUP = [AG-TEST]

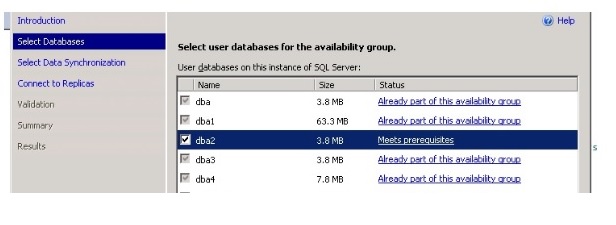
[](http://www.sqlserverblogforum.com/wp-content/uploads/2016/08/AG_Bug/5_HADR_join.jpg)

Repeat for each stuck database.

[](http://www.sqlserverblogforum.com/wp-content/uploads/2016/08/AG_Bug/6_Fixing_DB.jpg)

The database that had been removed <dba2> could be in any number of states, but after it finishes its recovery process, it should return to a RESTORING state.

7. On primary replica, Add the removed database –> primary replica –> right click AG databases–>select database –> join only –> connect the secondary –> finish.

[](http://www.sqlserverblogforum.com/wp-content/uploads/2016/08/AG_Bug/7_Add_DB.jpg)

In case, if you get any following error, check the backup table and restore a log backup with norecovery mode in all the secondary replica. Which are all missed or run during that time.

Msg 1478, Level 16, State 211, Line 1

The mirror database, ‘DBA2’, has insufficient transaction log data to preserve the log backup chain of the principal database.  This may happen if a log backup from the principal database has not been taken or has not been restored on the mirror database.

Added: Thanks [Amit Banerjee](https://twitter.com/banerjeeamit/status/762069339079966720) for your reply. I just reported to the connect.

<https://connect.microsoft.com/SQLServer/feedback/details/3022019>