Collation setting Change

It would be a challenging situation to be in for a SQL DBA when there is a mismatch of collation setting in SQL Instance or SQL Database. There would be couple of scenarios for both the case.

A- Collation setting change of SQL Instance

1- Collation change of SQL Instance without User Database
2- Collation change of SQL Instance with User Database

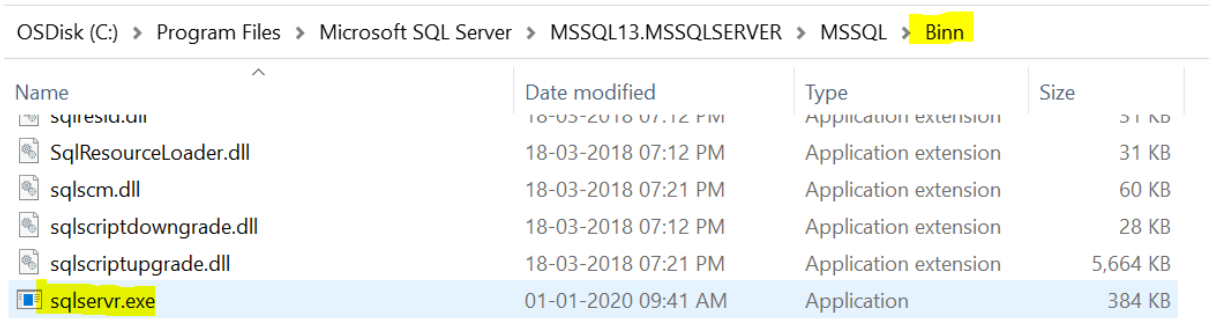
Scenario 1 (Without User DB)

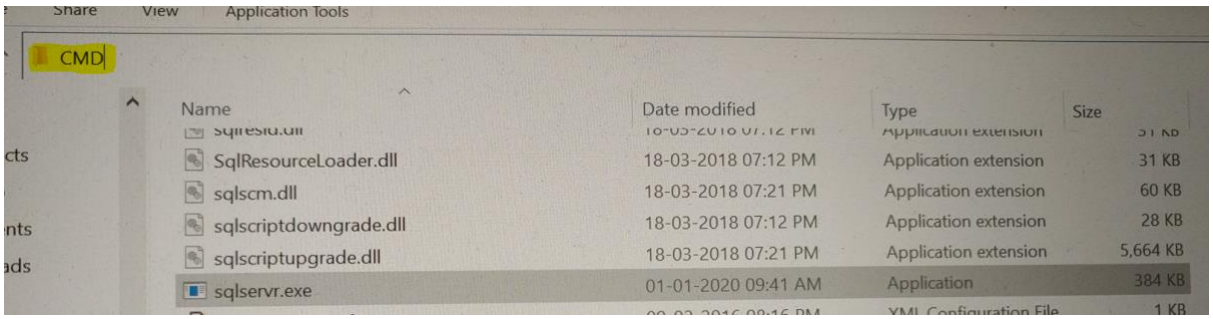
Requirement of a fresh installation for SQL Side by Side upgrade / Migration and found collation mismatch during Post installation validation check.

Step 1- Stop all the SQL services related to Default Instance (MSSQLSERVER) or Named Instance (eg: NEW2019)

Step 2- Open a CMD prompt & Navigate to the SQL Server Binn directory

This can be done easily by First Navigating to the Binn folder and then type **CMD** in the folder Path section and Enter. It will open Command prompt as





Step 3- Apply new Server Collation

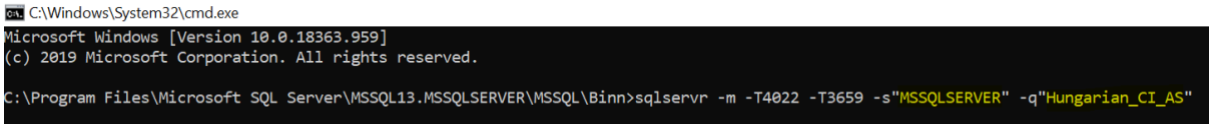
Execute the Below mention script

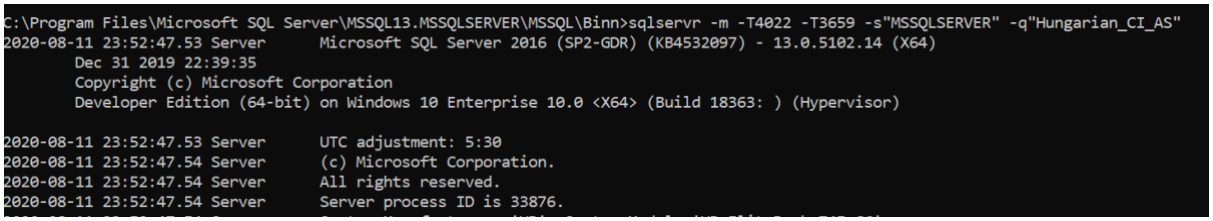
```
sqlservr -m -T4022 -T3659 -s"MSSQLSERVER" -q"Hungarian_CI_AS"
```

Change the MSSQLSERVER with Named Instance as required.

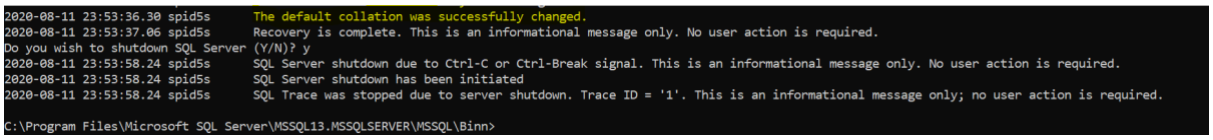
e.g: sqlservr -m -T4022 -T3659 -s"NEW2019" -q"Hungarian_CI_AS"

where NEW2019 is the Named Instance Name & Hungarian_CI_AS is the New collation.



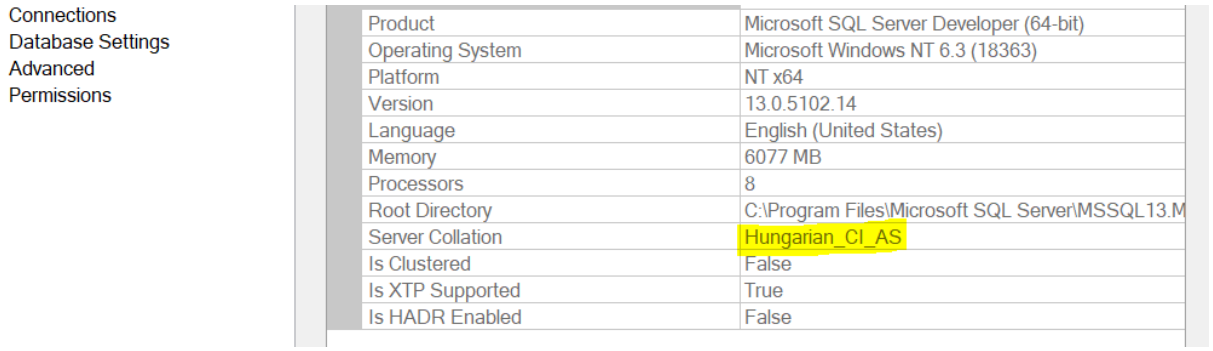


No User action required till you see the Successful message.



Step 4- Restart the SQL Instance related services.

Step 5- Verify SQL server Collation.



<u>Scenario 2</u> (With User DB)

Requirement of an installation for SQL Side by Side upgrade / Migration and found collation mismatch post the user databases are migration. Issue got highlighted only once application user have started facing issues as Databases & SQL Server are in different Collation. Databases are in correct collation as those are migrated from old server.

Step 1- Stop all Application services

Step 2- Take Backup of All Databases for rollback just in case it requires

Step 3- Detach all the User databases

Step 4- Follow (step 1 to Step 5) from Scenario 1

Step 5- Re attach all the user databases

B- Collation setting change of SQL Database

1- Collation change of Database Without Data
2- Collation change of Database with Data

Scenario 1 (Without Data)

Just created a new database with wrong collation setting and with no data

Execute the below t SQL script, Replace the [DatabaseName] with the actual database name

Step 1- Change the Collation of Database

```
USE master;
GO

-- Set to single-user mode
ALTER DATABASE [DatabaseName]
SET SINGLE_USER WITH ROLLBACK IMMEDIATE
GO

-- change collation
ALTER DATABASE [DatabaseName]
COLLATE Hungarian_CI_AS;
GO

-- Set to multi-user mode
ALTER DATABASE [DatabaseName]
SET MULTI_USER WITH ROLLBACK IMMEDIATE;
GO

--Verify the collation setting.
SELECT name, collation_name
FROM sys.databases
WHERE name = N'[DatabaseName]';
GO
```

Step 2- Change the collation of table Column

After Step 1, May be few table columns with different collation (columns of types char, varchar, text, nchar, nvarchar, and ntext) .It is advisable before making any changes collations, check whether the changes will generate any problems to the applications or not .

To list all table columns with their current collations, use the t SQL query mentioned below.

```
SELECT
 t.name,
 c.name,
 c.collation_name
FROM sys.columns c INNER JOIN sys.tables t
   ON t.object_id = c.object_id
WHERE c.object_id
    IN (SELECT object_id
       FROM sys.objects
       WHERE type = 'U')
    AND c.collation_name != 'NULL'
ORDER BY t.name, c.name
```

To alter a column collation to a new one, you may run the following query:

```
ALTER TABLE [<Table>] ALTER COLUMN [<Column>] <ColumnType>
 COLLATE <NewCollation>
```

In case of an error, make sure that the referencing constraints and indexes are dropped and recreate those after the collation has been changed.

Scenario 2 (Without Data)

Step 1-Export Data

Export Data Using bulk copy program utility (bcp) using the following command: for all the tables.

```
bcp [DatabaseName].dbo.[TableName] out TableName.txt -c -T
```

where -c performs the operation using a character data type and -T specifies that the bcp utility connects to SQL Server with a trusted connection using integrated security.

Step 2- Change Database Collation

```
-- Alter database collation
ALTER DATABASE [DatabaseName] COLLATE Hungarian_CI_AS;
```

Step 3- Drop and Recreate the tables

Step 4- Import previously exported data to all tables newly created

```
bcp [DatabaseName].dbo.[TableName] in Tablename.txt -c -T
```

NOTE: Please TEST the steps in Lower Environment before Implementing on production