**CREATE DATABASE USING POWERSHELL SCRIPT COLLECTION**

The CREATE DATABASE script set is a collection of PowerShell scripts created by perform the actions to create an clean database, this script set was develop thinking is necessary create a new Confirmations Manager Database in a new clients environments.

The script set collection could be used to perform any database creation, the script collection check some SQL Server like File Stream and Service Broker settings because this settings are require in the Confirmations Manager Database but these settings can be Disable after the process if this settings are not required.

**Requirements:**

1. Windows Server 2008 R2 or higher / Windows 7 or higher
2. PowerShell 2.0 or higher
3. SQL SERVER 2008 R2 or higher

If you are running the script set outside of a SQL Server machine you may need to install SQL Server Data ToolKit, this toolkit is necessary for import the SQLPS program and the SQLServer.Management.SMO libraries for the PowerShell scripts actions.

1. SQL Server Data Tools

If you are running the scripts from a SQL Server machine you can omit it.

**How to Use:**

The CREATE\_DATABASE script collection is a collection of many PowerShell scripts used to create a new database.

The CREATE DATABASE have a principal script named CREATE\_DATABASE.ps1 this script calls all PS scripts requires to create the database perhaps you can use the others scripts separately, this point is not recommend but if is necessary uses it on this way one-by-one script you could perform the action that you needed.

After you open your PowerShell prompt you need move the prompt till the path where you have saved the scripts.

For start to create the SQL database you need validate if the ID that you are using have sysadmin roles for create the DB.

This script set only read files .SQL scripts that you have create as .sql file from CREATE\_SCRIPTS\_BASELINE.ps1 if you try to use another file extension like .txt or another you will get errors.

The PS scripts tool create an error file log in the root of the path within this file you will find information relate with the error that not allow you complete your process with success. Below you can see a example of the error log

Example:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

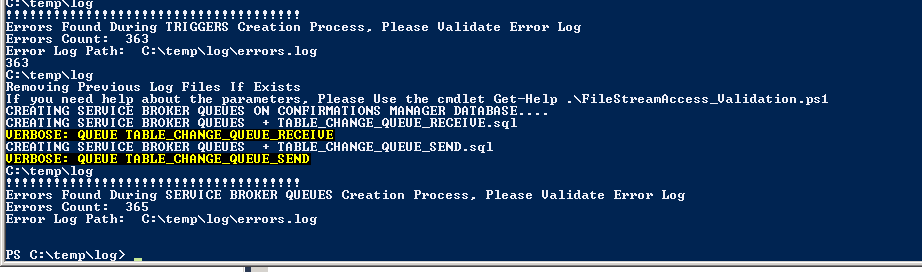
ERROR SCRIPTING

Error on Script : ConfirmMgr.F\_ASSOCIATED\_TRADE\_IDS.sql

Exception Message : Cannot find the user 'AFFINF', because it does not exist or you do not have permission.

Target Object :

Category Info : InvalidOperation: (:) [Invoke-Sqlcmd], SqlPowerShellSqlExecutionException

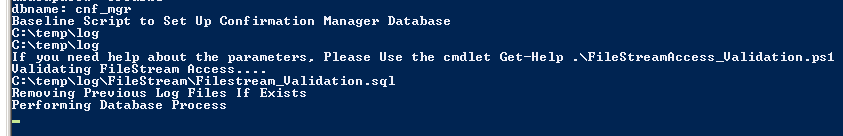
You will get a notification message like this

***Execution Steps:***

The CREATE\_DATABASE have an order of execution this order is stablished based in the steps requires to create the Confirmations Manager Database.

1. FileStream validation - FileStreamAccess\_Validation.ps1
2. Database Creation - CreateDataBase.ps1
3. Schema Creation - CREATE\_SCHEMA.ps1
4. Tables Creation - CREATE\_TABLES.ps1
5. Users Creation - CREATE\_USERS.ps1
6. DB Roles Creation - CREATE\_ROLES.ps1
7. Sequences Creation - CREATE\_SEQUENCES.ps1
8. User Defined Functions(UDF) Creation - CREATE\_UDF.ps1
9. User Defined Table Type(UDTT) Creation - CREATE\_UDTT.ps1
10. Views Creation - CREATE\_VIEWS.ps1
11. Stored Procedure Creation - CREATE\_SPROCEDURES.ps1
12. Default Constraints Creation - CREATE\_DFCONSTRAINT.ps1
13. Check Constraint Creation - CREATE\_CHKCONSTRAINT.ps1
14. Primary Keys Creation - CREATE\_PRIMARYKEYS.ps1
15. Foreign Keys Creation - CREATE\_FOREIGNKEYS.ps1
16. Triggers Creation - CREATE\_TRIGGERS.ps1
17. Services Broker Queues Creation - CREATE\_SBQUEUES\_SERVICES.ps1

The CREATE\_DATABASE script start calling one by one the scripts listed above in each step the process is showing on screen the overall progress also if some script set get issues the process append in the error log file the error description and shown the number of errors founded in the process.



File Stream Validation: Perform the actions to validate if the File Stream Access is active in the SQL Server where the DB will be created. The File Stream level access set up is **Full Access Enabled**



**Database Creation:** This script performs the actions for create the database that you want to build.

In this step you need take special care because you need provide in the SQL script the MDF, LDF and File stream path files.

Also you need the Database name that you want to create match the name like the database name provided in the PowerShell parameters because if the Database is created with a different name you will get a problem when the database objects try are created.

**Schema Creation:** In this step the database creation perform the action to create the schemas that you provide in the SQL script files.



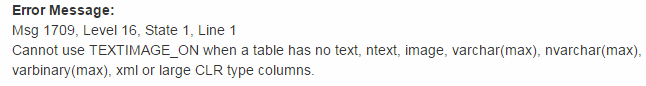
**Tables Creation:** when the database is created and the schemas are ready this step start to create the tables required. In this step the Primary keys are not created because some issue could be raised if one table depends on other tables because it the process will be affected and aborted.

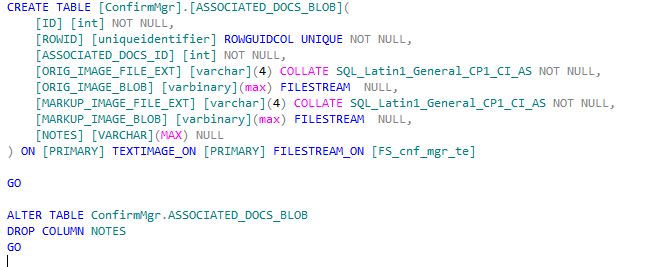
Preventing this issue the Create\_Baseline\_Database script generated the primary keys in separate files and not affects the table’s creation.

Possible issues may be with the tables that have File Stream columns because if the table not have a normal column like VARCHAR(max) then you could get an error related with TEXTIMAGE\_ON for solve this issue you should add a VARCHAR(max) and after the table creating you should add an ALTER table statement for DROP the column

Example:

In this example I have added a column named NOTES VARCHAR(max) this add is only to have possibility to create the table because as you can look the table have a FileStream column





Other point to be care is the File Stream that you are using, do you need check in your database creation and table’s creation the File Stream group name is the same.

Database Creation

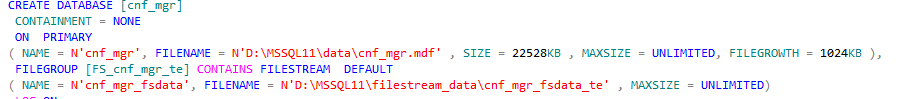
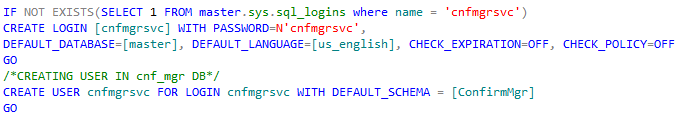


Table Creation



**User Creation:** in this step you could provide in the user script all database users that you need created also you will provide the SQL or Windows ID that are necessary create.



**Database Roles (DB roles) Creation:** when the Schemas, Database and users are ready the process creates the Roles requires to the users then assign the users to the Roles.

**Sequences Creation:** this step only require the Schemas are present in the Database not have others dependences for create the sequences.

**User Defined Functions (UDF) Creation**: the Functions could be a different order if have dependences of others objects like Stored Procedure or Views. Possible issues: Objects dependencies

**User Defined Table Type (UDTT) Creation:** This actions create the Table Type requires for the services broker and some procedure within the Confirmations Manager Database.

**Stored Procedure Creation:** Refer to the Stored Procedure Creation; their order could be different if the stored procedure depends on other objects. Possible Issue: Objects dependencies.

**Default Constraint Creation:** this step creates all Default values requires for the column tables and set dependencies of Sequences and others. Possible Issues: Objects dependencies like Sequences.

**Check Constraints Creation:** Refers to the Check constraint requires in the database tables.

**Primary Keys Creation:** Refer to the creation of the Primary Keys requires in the tables this step could be get issues if tables columns are not present.

**Foreign Keys Creation:** Perform the actions to create the foreign keys in the tables, this step require the tables and primary keys exists before you try to create this objects.

**Triggers Creation:** Refer to the Triggers need in the tables, may require others database objects are present.

**Service Broker Queues Creation:** In this step is performed the creation of the service broker objects requires for the Confirmations Manager Database.

***Folders structure***

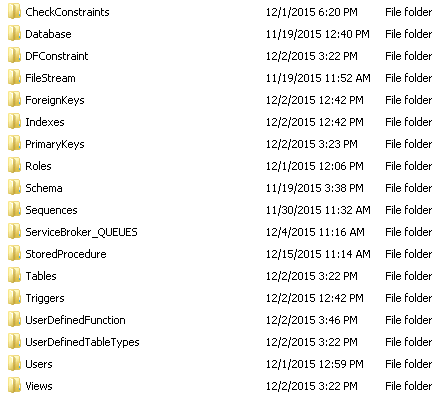
The CREATE\_DATABASE read a pre-stablished folder structure this folder require have the name defined in the script tool however the script tool CREATE\_SCRIPTS\_BASELINE build the folders structure when the database objects are scripted from a previous database.

If the script folders exists before you don’t need change or rename those folders.

The script tool runs across the folders structure as you can see in the section Execution Steps then the tool read each .sql script file to perform the actions.

|  |  |
| --- | --- |
| Objects | Folder Name |
| Check Constraint | CheckConstraints |
| Default Constraints | DFConstraints |
| Database Files and Creation | Database |
| SQL Server File Stream Configuration | FileStream |
| Service Broker Queues | ServiceBroker\_QUEUES |
| Database Schema | Schema |
| Foreign Keys | ForeignKeys |
| Indexes | Indexes |
| User Database Roles | Roles |
| Sequences | Sequences |
| Stored Procedures | StoredProcedure |
| Tables | Tables |
| Triggers | Triggers |
| User Defined Functions (UDF) | UserDefinedFunction |
| User Defined Table Types (UDFF) Service Broker part | UserDefinedtableTypes |
| Database Users (SQL Login too) | User |
| Views | Views |
|  |  |

More folder could be add in the future and only is necessary in the script tool add the folders path and scripts to perform the actions



***Logs***

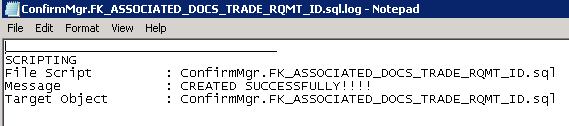
The script tool has 2 kinds of logs:

1. **Objects creation log:** in this log is saved the File Script name, Message and Target. Each object processed have a separate log and saved in the log folder under object root path

For Example:

All objects processed for created the Foreign Key DB objects have their log under the path ..\ForeignKeys\log in this path only are saved the log of the objects completed with success.

When the process finish you will be able to see a log like this.

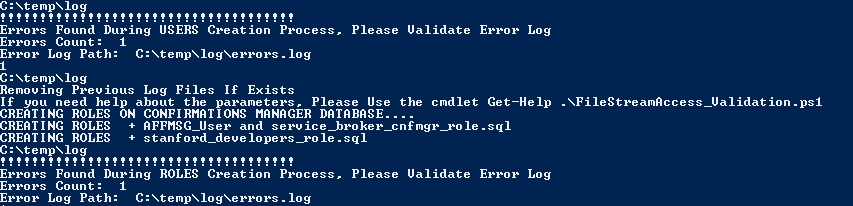


1. **Error log:** the error log are saved in an unique file, this file is saved in the root path. For a best and easy issues identification all errores could be tracked in the principal log in this way you will be able to troubleshoot any situation or identify if your error is related with a previous issue or error.

Also the error log notify in the end of the process how many errors you have during the process then you can perform the requires actions to fix it.

For example:

The process gets a error when perform the actions to create an user, the process not abort and continue performing the others actions but shows the Error and the Number of errors.



Because the process get an error in this moment the Script tool created a new error file in the root path, this error file start to append all new errors that raise during the creation process.

Now when you read the error log you will be able to determine if this error affect the good Database work or not, if this error affect the Database function you could try to fix it in the Database.

