



tSQLt

Unit Test for SQL Databases

Arguments

- 1.- Testing will improve
- 2.- Deployment frequency will go up
- 3.- Improve response to business needs
- 4.- More time for experimentation
- 5.- Stable releases

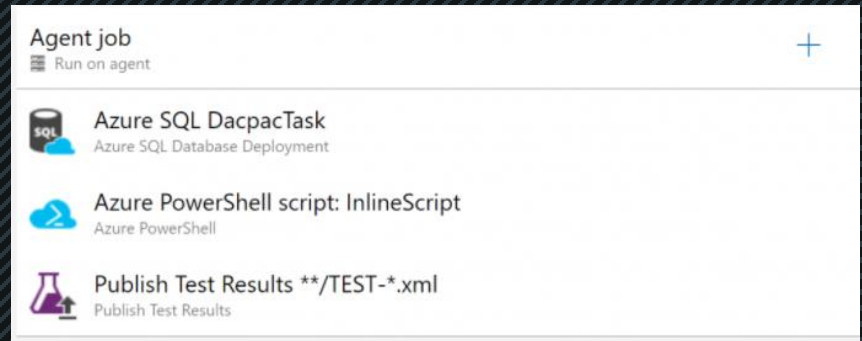
What basic test are being generated

- Object Existence
- Database Collation
- Function Parameters
- Procedure Parameters
- Table Columns
- Table Constraints
- Index Columns
- View Columns

Demo

DepOps

1. Create a project
2. Setup build pipeline
3. Run tests



```
1 $connectionString = "<ADD YOUR CONNECTION STRING HERE>"
2 $sqlCommand = @"
3 BEGIN TRY
4 EXEC tsqLl.RunAll
5 END TRY
6 BEGIN CATCH
7 END CATCH;
8 EXEC tsqLl.XmlResultFormatter
9 "@
10
11 $connection = new-object system.data.SqlClient.SqlConnection($connectionString)
12 $command = new-object system.data.sqlclient.sqlcommand($sqlCommand,$connection)
13 $connection.Open()
14
15 $adapter = New-Object System.Data.sqlclient.sqlDataAdapter $command
16 $dataset = New-Object System.Data.DataSet
17 $adapter.Fill($dataSet) | Out-Null
18
19 $connection.Close()
20 $dataSet.Tables[0].Rows[0].ItemArray[0] | Out-File "$(System.DefaultWorkingDirectory)/drop/sql-database/TEST-$(DatabaseName).xml"
```

<https://nikolaithomassen.com/2019/02/26/azure-sql-unit-testing-with-tsqlt-using-azure-devops/>

Demo

Thank you

Links

1. tSQLt
<https://tsqlt.org/>
2. PSModuleDevelopment – Create Project using Template
<https://psframework.org/>
3. SSDT Template
<https://github.com/sanderstad/SSDT-With-tSQLt-Template>
4. PStSQLtTestGenerator
<https://github.com/sanderstad/PStSQLtTestGenerator>
5. Demo Code
<https://github.com/antonio-bravo/CoP>