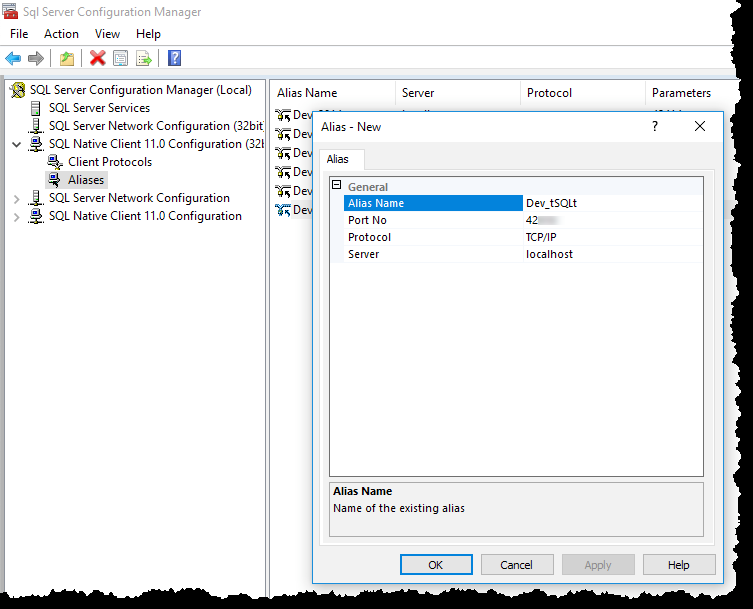
# Install the tSQLt build

## Clone Git Repository to local directory

* For manual download: <https://github.com/tSQLt-org/tSQLt>
* Git
  1. Create or locate directory for repository
  2. Execute (CMD.EXE)
     1. CD [that directory]
     2. git clone <https://github.com/tSQLt-org/tSQLt.git>
  3. This creates a tSQLt directory in that place. Later references to .\tSQLt\ in this document refer to that location

## Setup instance alias (optional)

1. Enable TCP connections on instance (see Google)
2. Create Dev\_tSQLt Alias  
   

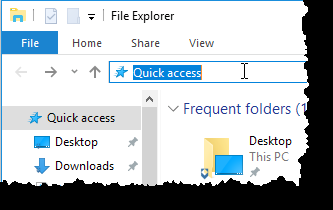
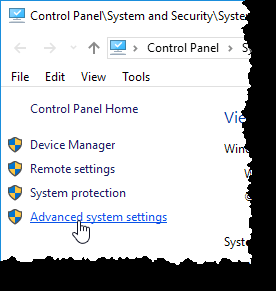
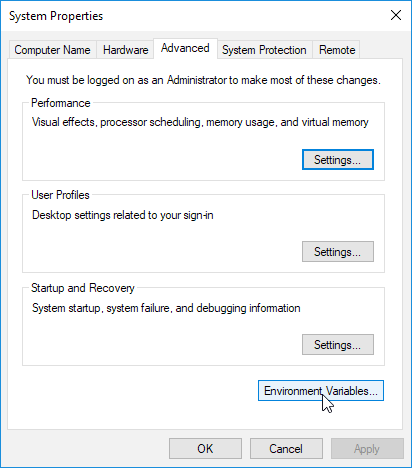
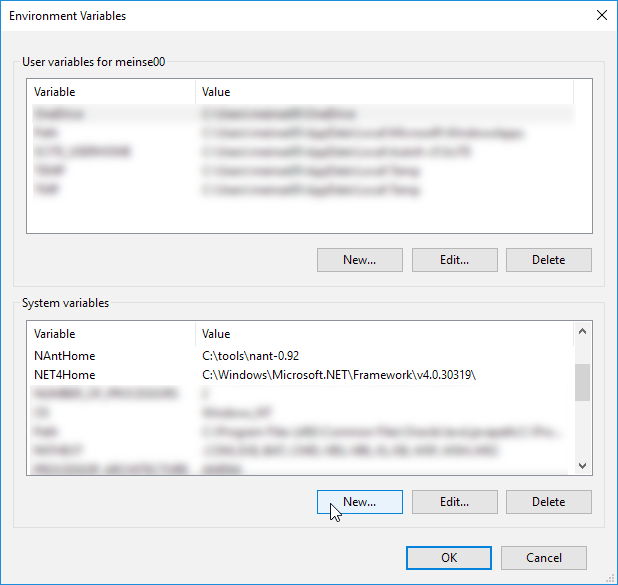
## Create Logins

1. Execute (CMD.EXE):
   1. CD .\tSQLt\Build
   2. Setup.bat Dev\_tSQLt

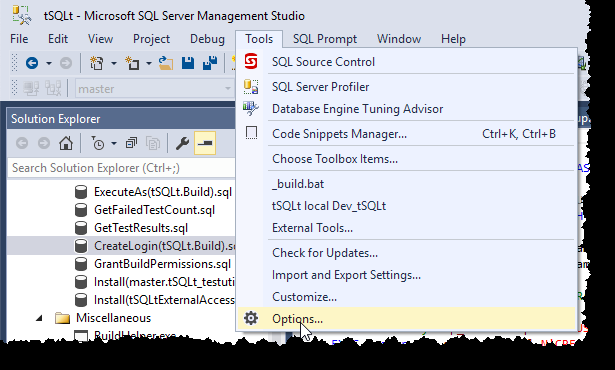
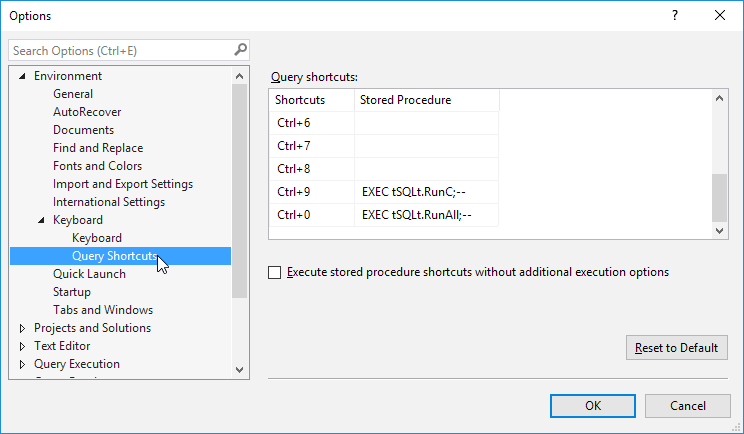
## Install NANT

1. Download NANT from <http://nant.sourceforge.net/>
   1. (likely at <https://sourceforge.net/projects/nant/files/nant/0.92/>)
2. Unzip to folder on C: (not under C:\Users)
3. You might have to follow these steps before unzipping: <https://stackoverflow.com/questions/8605122/how-do-i-resolve-configuration-errors-with-nant-0-91>

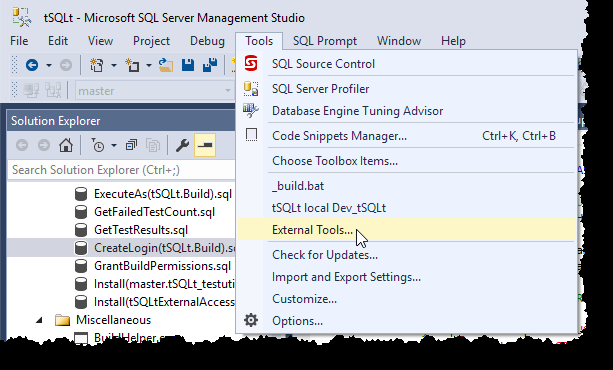
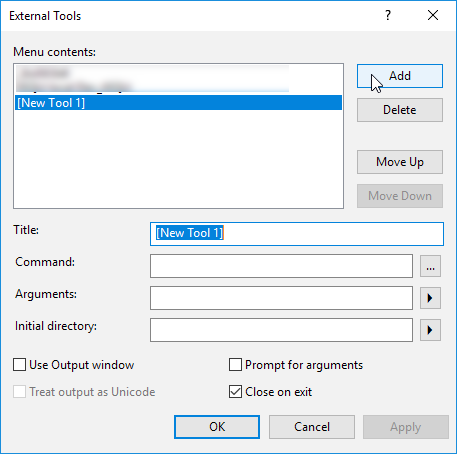
## Set Environment Variables

1. Open File Explorer window
2. Click on the Path Bar (behind the current path, not on)  
   
3. Type in Control Panel\System and Security\System
4. Hit <Return>
5. Select [Advanced system settings]  
   
6. Click on [Environment Variables…] button  
   
7. Create 2 new System variables ( NAntHome & NET4Home )  
   
   1. Set the paths according to where NANT and .NET 4.0 are installed!
8. Restart SQL Server Management Studio (if it was running).

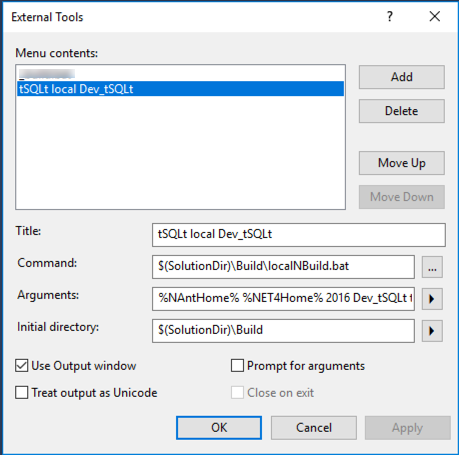
## Create SSMS Tool & Shortcut

1. Open SSMS
2. Menu / Tools / Options  
   
3. Query Shortcuts  
   
4. Enter these shortcuts (Exactly as typed here!)

|  |  |
| --- | --- |
| Ctrl+9 | EXEC tSQLt.RunC;-- |
| Ctrl-0 | EXEC tSQLt.RunAll;-- |

1. Menu / Tools / External Tools  
   
2. Click [ADD] button  
   
3. Type in the information below:  
   (The last three items in the Arguments box are: SQL Server version, Instance, Database. The version is important, because tSQLt has different tests for different versions.)

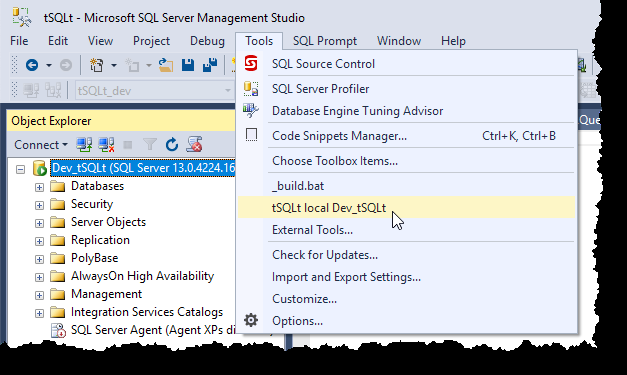
|  |  |
| --- | --- |
| Title | tSQLt local Dev\_tSQLt |
| Command | $(SolutionDir)\Build\localNBuild.bat |
| Arguments | %NAntHome% %Net4Home% 2016 Dev\_tSQLt tSQLt\_dev |
| Initial directory | $(SolutionDir)\Build |

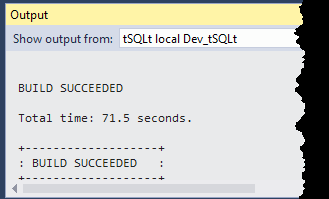
1. Change the check boxes to this:  
   

## Prepare the build

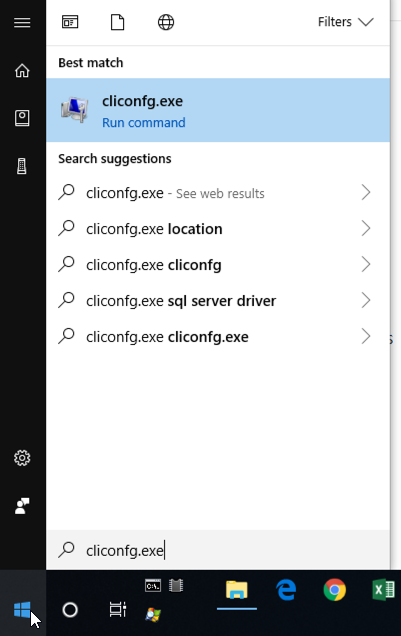
1. Open SSMS
2. Open the tSQLt solution in .\tSQLt \tSQLt.ssmssln
3. Connect Object Explorer to the Dev\_tSQLt instance
4. Enable CLR (you can use EnableCLR.sql in the build folder)
5. Make sure you have .NET 2.0 installed
   1. needed for compatibility with SQL Server 2008
   2. In newer windows versions you’ll have to install 3.5 which will automatically install 2.0

## Run the build

1. Menu / Tools / tSQLt local Dev\_tSQLt  
   
2. Confirm Build Succeeded:

Note - You’ll get several of these messages throughout. The final one is the important one.  


## FLUP

* SQL Server Alias with other tool  
  
* Deal with SigningKey (two keys with switch batch script)
* Check if newest msbuild.exe suffices.
* Provide instructions on SigningKey
* Fix “statement” in build file
* Enable CLR

|  |
| --- |
| * sp\_configure 'Advanced Options', 1 * RECONFIGURE * sp\_configure 'CLR Enabled', 1 * RECONFIGURE   There’s also a file in the build folder ( |

* Fix “publickeytoken” in EAKE tests