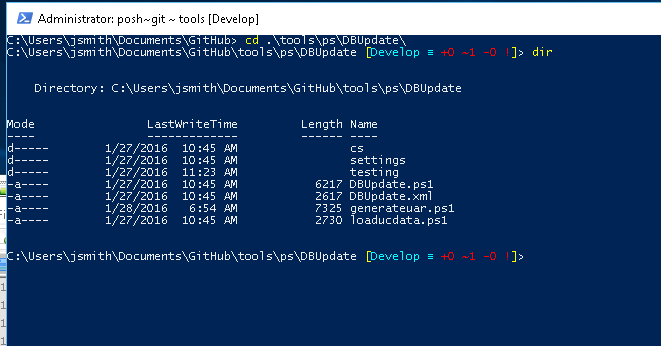
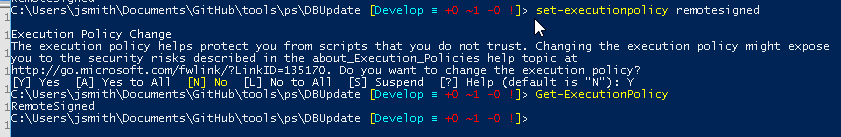
Setup

The scripts and supporting projects and documentation is located in TFS here: $/CSCE/IDF/Projects/Powershell\_DBUpdate



Place the Powershell scripts on your local system. Run powershell console as an Admin. Now change the execution policy like below (this allows scripts to un on your box)



You used to need always launch the PowerShell console ad an admin. Because the scripts not copy idf.exe to a local temp file location you no longer need to run the scripts as an admin

There is a settings file named DBUpdate.XML, a generate UAR script (which can run standalone) a loaducdata script (which can run standalone). And the everything script called DBUpdate which will run everything.

Some benchmarks:

1. Running generate UAR runs in about 21 minutes
2. Running loaducdata takes about 4 minutes
3. Running DBupdate for table IMST takes about 34 minutes (This includes generate uar and loaducdata completeing in parallel

Network IO is slower so I have created a settings file in XML so some of the work can be done on your local box. This settings file also unhardcodes the scripts. Below is a table of all the settings and what they mean. In the future we can probably parse some of the settings cased on other setting to decrease the number of setting file tags.

Global Settings

|  |  |  |
| --- | --- | --- |
| **Tag** | **Meaning** | **Example** |
| UnifaceIDFPath | Uniface location of idf.exe | T:\UNIFACE\U9605\X505\common\BIN\idf.exe |
| TFSToolPath | Location of TFS power tools (need to install) | C:\Program Files (x86)\Microsoft Visual Studio 12.0\Common7\IDE\tf.exe |
| TFSIncludePath | Location of our include procs | H:\unicomp\CSCE\CS06\CS08.2.27\IncludeProcs |
| IncludeArgs | Source location for include procs | $/CSCE/CS06/CS08.2.27/USYS/IncludeProcs |
| ImportIncludes | Import includes switches | XML:H:\Unicomp\CSCE\CS06\CS08.2.27\IncludeProcs\\*.ipx |
| TFSModelPath | Location of our models | H:\unicomp\CSCE\CS06\CS08.2.27\Models |
| ModelArgs | Source location of models | $/CSCE/CS06/CS08.2.27/USYS/Models |
| ImportModels | Import model switches | XML:H:\Unicomp\CSCE\CS06\CS08.2.27\Models\\*.xml |
| TFSComponentPath | Location of components | H:\unicomp\CSCE\CS06\CS08.2.27\Components |
| ComponentArgs | Source location of models | $/CSCE/CS06/CS08.2.27/USYS/Components |
| ImportComponent | Import component switches | XML:H:\Unicomp\CSCE\CS06\CS08.2.27\Components |
| ASNCorePath | Local location of Core ASN file (I have this local) | D:\DBUpdate\Devo\_v2\ |
| INICorePath | Location of INI file (can be main) | /ini=P:\CS08\_2X\CS08\_2\_27\USERS\MAIN\idf96.ini |
| TempFileLocation | Local temp file dir (idf.exe gets copied here) | D:\DBUpdate\Temp\ |

DBUpdate Settings

|  |  |  |
| --- | --- | --- |
| **Tag** | **Meaning** | **Example** |
| LogPath | Location of the IDF log files | D:\DBUpdate\Devo\_v2\Logs\ |
| ModelPrompt | Prompt which shows at the top of the main script. To pick the tables to compile | DBUpdate script - Table(s) ordr,schl or (cr = all[slow]) |

LoadUCData Settings

|  |  |  |
| --- | --- | --- |
| **Tag** | **Meaning** | **Example** |
| ASNFileName | ASN filename | idf.asn |
| Tool | C# executable to create loaducdata file | D:\DBUpdate\Devo\_v2\LoadUCData.exe |
| LoadUCDataFile | Production location of loaducdata file | P:\CS08\_2X\CS08\_2\_27\UTILS\LoadUCData.sql |

GenerateUARFile Settings

|  |  |  |
| --- | --- | --- |
| **Tag** | **Meaning** | **Example** |
| TFSWorkspace | Local TFS workspace | d:\messages |
| MessageArgs | Source location of file | $/CSCE/CS06/CS08.2.27/USYS/messagesgenerated.uar |
| ASNMessagePath | Local ASN location | D:\DBUpdate\MessagesGenerated\_v2\ |
| INIGenerated | Local ini file | /ini=D:\DBUpdate\MessagesGenerated\_v2\idf96.ini |
| ResourcesGenerated | Local place to generated messages | D:\DBUpdate\MessagesGenerated\_v2\resources\msg |
| ZipLocation | Location of Zip tool | D:\DBUpdate\MessagesGenerated\_v2\7za.exe |
| LogPath | Location of the IDF log files | D:\DBUpdate\MessagesGenerated\_v2\Logs\ |
| SQLServer | SqlServer settings for drop table | (See belowTable) |

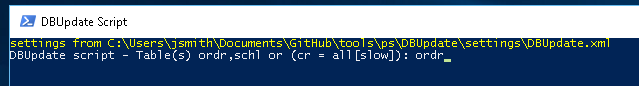
SQLServer Settings

|  |  |  |
| --- | --- | --- |
| **Tag** | **Meaning** | **Example** |
| Server |  | AL-SQL2K8R2-S01 |
| Database |  | Messages2xGenerated |
| User |  | ital |
| Password |  | ital |
| DropTableList | Tables to be dropped from the DB before regenerating the models and messages | uobj,ouobj,usource,ousource |

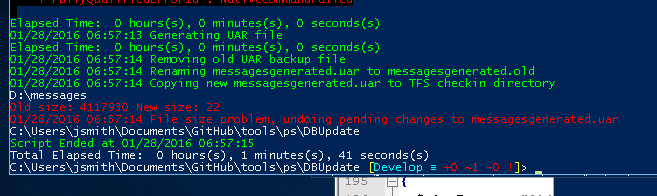
Users Settings

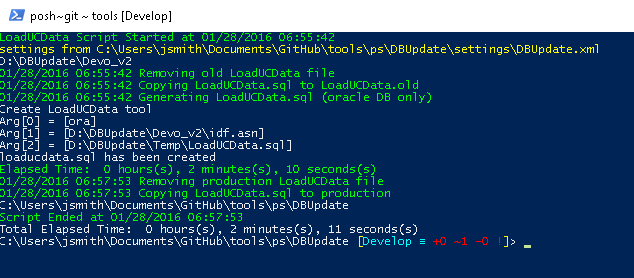
|  |  |  |
| --- | --- | --- |
| **Tag** | **Meaning** | **Example** |
| SettingsDirectory | Scripts will first load dbupdate.xml from root, if there is a user setting override this is where the script will look next. Allows Default and user settings | C:\Users\jsmith\Documents\GitHub\tools\ps\DBUpdate\settings\ |

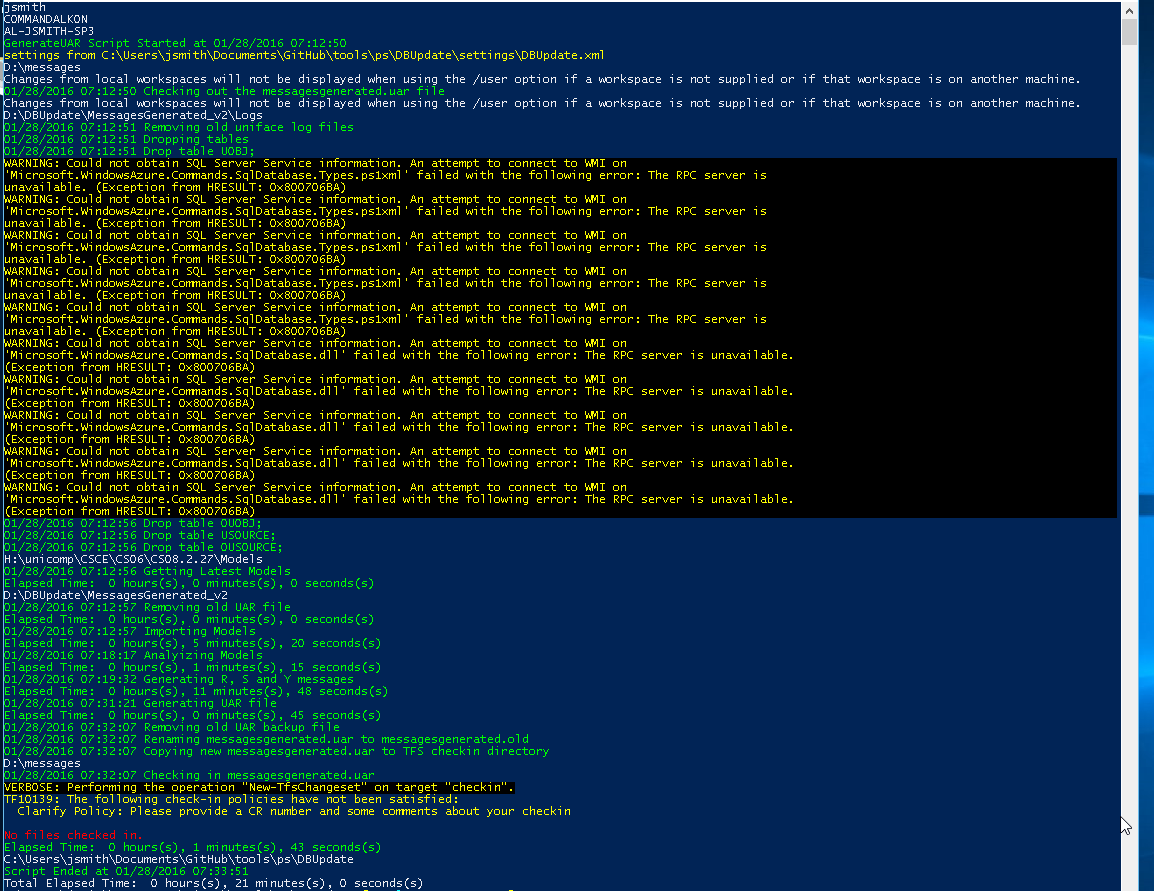
To run: CD to the scripts dir and run them you can do more than one table at once separate by commas



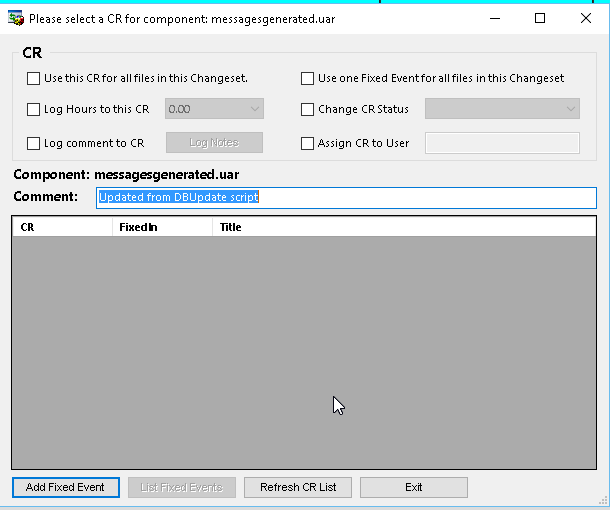
If the generate uar is smaller than the one already checked in you will get a warning and it won’t check in also it will get unlocked. (figure out what is strange before continuing)







The yellow warnings was caused if you have Azure powershell loaded on your box and the warnings cannot be suppressed the tables still drop fine. I fixed it by uninstalling Azure power shell scripts. Or you can just ignore the warnings everything is still good.



Jeff’s code will be triggered on checkin

Example setup

Below are some pics of my local setup. I suggest making changes to DBUpdate.xml for either v2 or v3 specific settings or then each user can have an override directory where his setting can be kept.

