Git version: 1.4 Easier XML setup

Benchmarks full run on Core version 2.26

# All Scripts started at 2/11/2016 06:34:03

1. Running generate UAR runs in **18 minutes 55 seconds**
   1. Removing old messages 1 minute 9 seconds
   2. Importing models 5 minutes 17 seconds
   3. Analysing models 59 seconds
   4. Generating R, S, and Y messages 10 minutes 40 seconds
   5. Generating UAR file 43 seconds
2. Running loaducdata runs in **2 minutes 3 seconds**
3. Running DBupdate for all tables runs in
   1. Getting latest components 45 seconds
   2. Getting latest models 24 seconds
   3. Getting latest include procs 7 seconds
   4. Getting latest global procs 19 seconds
   5. Importing models 5 minutes 2 seconds
   6. Analysing models 2 minutes 13 seconds
   7. Generating R, S, and Y messages 16 minutes 30 seconds
   8. Importing components 30 minutes 2 seconds
   9. Compiling services
   10. Compiling forms (happens in parallel with services compiling)
4. Total elapsed time **2 hours 30 minutes**

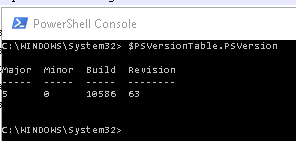
(This includes generate uar and loaducdata completeing in parallel)

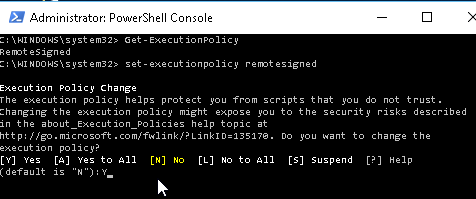
Be aware This is the worst case - picking just the Table(s) you have changed is much faster.

# All Scripts ended at 2/11/2016 08:51:47

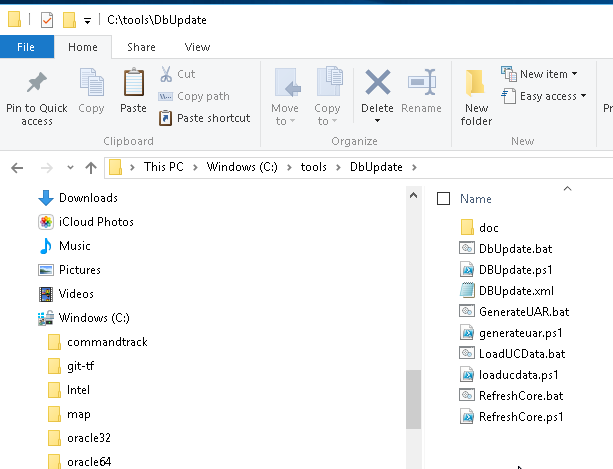
Computer Install/Setup

1. Install recent Power shell (windows includes this but please check)
   1. To open power shell console
      1. Go to start menu and type power shell
      2. If it is not there then install it
   2. To see your power shell version number
      1. Open a power shell console and do this



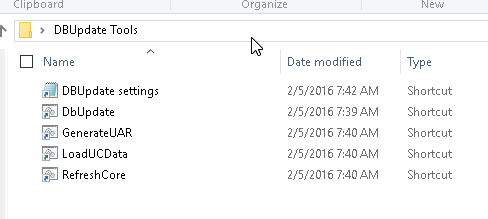
* + 1. My Windows 10 box is version 5
  1. Now Get power shell to allow scripts to run on your box
     1. Open a **Admin** power shell console and do this
     2. 
     3. Yours will probably be set to restricted initially (You want remote signed)
     4. You just need to do this just one time at setup
  2. Ok, now you can run the scripts

1. Install SQL server 2014 together with all the tools
   1. Make sure to check the power shell support (not loaded down by the default install)
   2. If you have older SQL server on your box let do the upgrade to get the new stuff
      1. SQL server will look on your system when doing this and you might need to install service packs first
      2. Keep relaunching until your SQL server is updated
2. Install TFS Power Tools 2012 or later
   1. The tool we want is named tf.exe
      1. This tool allows TFS get and checkout and check-in support
      2. Once it is installed you will need to update its location in the script setup file
3. Install 7-ZIP tool
   1. You can just point to the 7za.exe Command has already put in place
   2. I copied this exe local to my box and pointed it location out in dbupdate.xml
4. Place Power shell scripts on your local drive



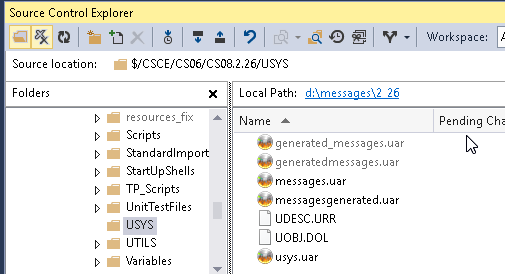
* 1. TFS location of scripts is $/CSCE/IDF/Projects/Powershell\_DBUpdate
  2. Location of this document is $/CSCE/IDF/Projects/Powershell\_DBUpdate/doc
  3. Location of the C# supporting tool loaducdata.exe is $/CSCE/IDF/Projects/Powershell\_DBUpdate/cs
     1. Compile this tool in Release mode
     2. Move the EXE file local if you wish

1. Place batch file shortcuts on your desktop
   1. Right mouse on the bat file in your power shell directory
   2. Go to your desktop and do a past shortcut now you can launch the tools with a mouse click

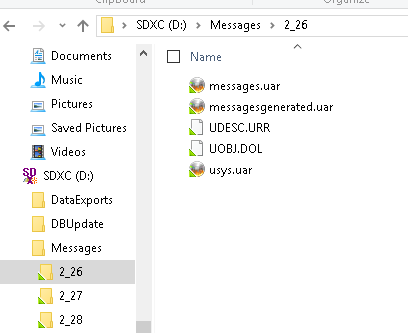


* 1. **DBUpdate settings** is a shortcut to quickly edit the settings file normally only the version number should be changed (once everything is set up)
  2. **DBUpdate** is a shortcut to Launch all of the scripts to run concurrently
  3. **Generate UAR** is a shortcut to recreate the messages UAR file
  4. **LoadUCData** is a shortcut to create the LoadUCData.sql file
  5. **RefreshCore** is a shortcut to recompile core (it also launches the loaducdata script). Make sure to tell the tool which table(s) you want to do. Doing everything takes a while (will try to speed this up in the future)

1. Point your TFS messages USYS workspace on your local drive (cannot be a networked drive)

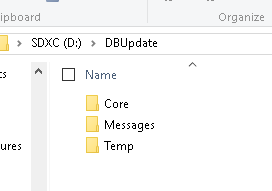


* 1. This is the TFS location for the messagesgenerated.uar file. Notice I have overridden the standard H: drive location to my local SSD drive
  2. This needs done for each version
  3. Doing this speeds stuff up and allows TFS power tools to be able to check in the file
  4. There were problems because it’s a real big file and it’s a network drive

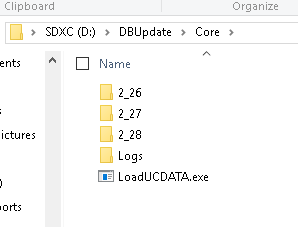


* 1. Notice I have 3 regions set up right now

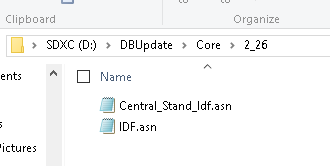
1. Set up your local directory structure



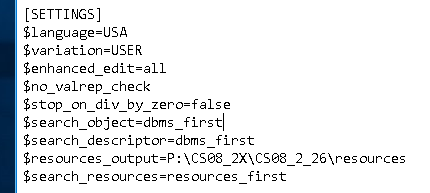
* 1. Set up the Core Directory (used for RefreshCore)

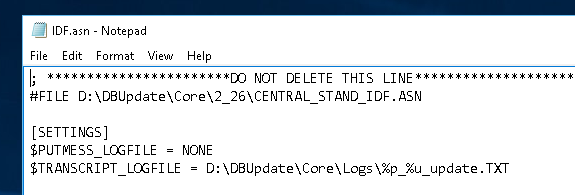


* + 1. Place the LoadUCData.exe tool at the root of core
    2. Create sub directories in the format “2\_26” (major version “\_” minor version)



* + - 1. Copy IDF.ASN and CENTRAL\_STAND\_IDF.ASN from P:\CS08\_2X\CS08\_2\_26\USERS\MAIN
      2. Edit central\_stand\_idf.asn so there are no relative resource file paths



* + - 1. Edit IDF.ASN
         1. 
         2. Edit #FILE to be your local ASN file
         3. Set $putmess = NONE (must be capitalized and no quotes. This avoids the white pop-up windows)
         4. Set up $transcript to point to the Logs dir. Also notice the %p\_%u. This will tack on the PID number and the Username of who ran the script
    1. Create a Logs sub directory (stores the uniface log files)
  1. Set up the Messages Directory (used for GenerateUAR)
     1. Put a LOGS sub directory
     2. Create a resources subdirectory and inside of that a msg subdirectory
     3. Add tool 7za.exe to the messages directory (zip up tool)
     4. Copy the IDF.ASN file from P:\CS08\_2X\MessagesGenerated\USERS\MAIN
     5. Edit the IDF.ASN file
        1. In settings have this
           1. $PUTMESS\_LOGFILE = NONE
           2. $TRANSCRIPT\_LOGFILE = D:\DBUpdate\Messages\Logs\%p\_%u\_update.TXT
  2. Set up the temporary files dir (used in loaducdata and generate uar)

1. Update your dbupdate.xml file with your version and your local directory structure settings
   1. See the tables describing these settings shown below
   2. You will see an example of how my dbupdate.xml is set up
   3. Ricky may want to provide his version 3 setting file.
2. Move custom dbupdate.xml to your local directory structure (if necessary)
   1. Copy dbupdate.xml from the power shell scripts location to the local Core ASN file location
      1. Dbupdate.xml will load from the standard location to find the version you want to run. It will then look in your local directory for the setting file
      2. If it is found there this will override the standard settings
   2. Example: When we come up under a newer Uniface version
   3. Example: you want a non-standard log location or to use a different Zip tool
3. Ok, now the tool should be set up.

Global XML Settings

|  |  |  |
| --- | --- | --- |
| **Tag** | **Meaning** | **Example** |
| CoreVersion |  | 2.26 |
| HDriveRoot |  | H:\unicomp\CSCE\ |
| PDriveRoot |  | P:\CS08\_ |
| UnifaceIDFPath | Current Uniface idf | T:\UNIFACE\U9605\X505\common\BIN\idf.exe |
| TFSToolPath | TFS Power tools | C:\Program Files (x86)\Microsoft Visual Studio 12.0\Common7\IDE\tf.exe |
| TFSIncludeFolder |  | IncludeProcs |
| TFSIncludeExtension |  | ipx |
| TFSGlobalFolder |  | GlobalProcs |
| TFSGlobalExtension |  | prx |
| TFSModelFolder |  | Models |
| TFSModelExtension |  | xml |
| TFSComponentFolder |  | Components |
| TFSComponentExtension |  | cmx |
| ASNCoreRoot | Local Core Dir | D:\DBUpdate\Core\ |
| INICoreName |  | idf96.ini |
| INICoreLocation | P Drive Core INI file Location | USERS\MAIN\idf96.ini |
| TempFileLocation |  | D:\DBUpdate\Temp\ |

RefreshCore Settings

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

LoadUCData Settings

|  |  |  |
| --- | --- | --- |
| **Tag** | **Meaning** | **Example** |
| ASNFileName | ASN filename | idf.asn |
| Tool | C# executable tool that creates loaducdata.sql file | D:\DBUpdate\Core\LoadUCData.exe |
| LoadUCDataFolder | Production folder of loaducdata file | UTILS\LoadUCData.sql |

GenerateUARFile Settings

|  |  |  |
| --- | --- | --- |
| **Tag** | **Meaning** | **Example** |
| TFSWorkspaceRoot | Local TFS workspace | d:\messages\ |
| MessageArgs | Source location of file | USYS/messagesgenerated.uar |
| ASNMessagePath | Local ASN location | D:\DBUpdate\Messages\ |
| INIMessageLocation | Local ini file | MessagesGenerated\USERS\MAIN\idf96.ini |
| ResourcesGenerated | Local place to generated messages | D:\DBUpdate\Messages\resources\msg |
| ZipLocation | Location of Zip tool | D:\DBUpdate\Messages\7za.exe |
| LogPath | Location of the IDF log files | D:\DBUpdate\Messages\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 |

LoadUCData script Walkthrough

1. Loads dbupdate.xml config file
2. Possibly overrides dbupdate config in local core directory
3. Makes sure the ASN Core path exists
4. Makes sure the temporary folder exists
5. Makes sure the loaducdata.exe tool can be found
6. Removes the old loaducdata.old file from the temp directory
7. Copies loaducdata.sql from production into the temp dir as loaducdata.old
8. Runs the tool creating the new loaducdata.sql in the temp dir
9. Deletes loaducdata.sql from production
10. Copies loaducdata.sql from temp dir into production dir

GenerateUARFile script Walkthrough

1. Loads dbupdate.xml config file
2. Possibly overrides dbupdate config in local core directory
3. Make sure that Uniface idf.exe can be found
4. Makes sure that the TFS power tool is found
5. Make sure that the temporary file dir exists
6. Makes sure that the TFS workspace dir can be found
7. Makes sure that the 7-Zip file tool can be found
8. Does a get latest and a checkout of the messagesgenerated.uar file from TFS
9. Warns if it is already locked and will bail out if it is …
10. Removes all the old LOG records for the user who runs the tool
11. Drops the tables in the SQLServer section of the config file (currently uobj, ouobj, usource, and ousource)
12. Gets the latest models from TFS
13. Removes the old messagesgenerated.old file
14. Copies production messagesgenerated.uar to messagesgenerated.old in the temp dir
15. Removes all the old resources\msg files
16. Imports the models
17. Analyses the models
18. Generates all the R, S and Y messages an puts them in the local resources dir
19. Uses the 7-ZIP tool to compress dir resources\msg into messagesgenerated.uar puts this in the temp dir
20. Copies the new messagesgenerated.uar to the TFS workspace directory
21. Compares the new file size to the old file size if the new size is smaller than the old size then undo the checkout with the TFS tool
22. If the size is good then a check-in will happen

RefreshCore script Walkthrough

1. Loads dbupdate.xml config file
2. Possibly overrides dbupdate config in local core directory
3. Make sure that Uniface idf.exe can be found
4. Makes sure that the TFS power tool is found
5. Make sure that the temporary file dir exists
6. Makes sure the ASN Core path exists
7. Prompts the user for which table(s) to process or carriage return to recompile everything
8. Removes the user specific log files from the Logs dir
9. Gets latest components
10. Gets latest models
11. Gets latest include procs
12. Gets latest global procs
13. Imports include procs
14. Imports global procs
15. Import models
16. Analyses models
17. Generates R, S and Y messages
18. Spawns off the loaducdata power shell script
19. Compiles all Global procs
20. If the user pressed a carriage return
    1. Import all components
    2. Compile all services
    3. Compile all forms
21. If the user picked some table(s)
    1. Import only Components that had this table in them
    2. Compile only components that had the table in it

DBUpdate script Walkthrough

1. Launches RefreshCore script (with priv)
2. Launches Generate UAR script (with priv)

Future ideas for improvement

1. Better scanning of files to compile only what is necessary
2. Parallel compiling (Like all EDT\* forms at the same time as all RPT\* forms)
3. Scan logs when done to see model Analyze warning messages and form/service compile errors
4. Scan logs for certain Warnings that we do not want to see
5. Perhaps it can be faster to update a local resources directory when compiling and at the end do a copy / overwrite to the networked P drive resources folder? (may need to experiment)
   1. Add robocopy commands to see if I can speed up compiling
   2. Only refresh the newer files on the P Drive
6. Speed up the scripts as long as the tool stays accurate and consistent