CIS Update – Part Library - Synchronization with Teamcenter

Technical Design and Solution Implementation Document



Document Detailss

|  |  |
| --- | --- |
| Installation Instruction Document | |
| Project Owner | **Kopp William, Simmonds Chris** |
| Process Title | **CIS Property Update** |
| Document Author | **Sanjeev Beemidi** |
| Document Version | **1.0 – Issued for Approval** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document Edit History | | | | |
| Version | **Description** | **Name** | **Role** | **Date** |
| V1.0 | Initial draft | **Sanjeev Beemidi** | **EDA Integrator** | **04 Sep 2023** |

Contents

[1. Introduction 4](#_Toc144718399)

[1.1. Scope 4](#_Toc144718400)

[1.2. Key Assumptions 4](#_Toc144718401)

[1.3. As-Is 4](#_Toc144718402)

[1.4. Proposed Solution: 4](#_Toc144718403)

[1.5. Prerequisites 4](#_Toc144718404)

[1.6. Package Unpacking 5](#_Toc144718405)

[1.7. CIS Update High Level Architecture: 5](#_Toc144718406)

[1.8. Input File Creation Guideline 6](#_Toc144718407)

[1.9. Preparing Configuration file: 6](#_Toc144718408)

[1.10. CIS Update Automation Design: 7](#_Toc144718409)

[1.11. CIS Process flow Diagram: 9](#_Toc144718410)

[1.12. Installation – Actual Run 10](#_Toc144718411)

[1.13. User Guide - Purpose 13](#_Toc144718412)

[1.14. Scope 13](#_Toc144718413)

[1.15. Pre-Requisites 13](#_Toc144718414)

[1.16. Execute CIS\_Update utility 13](#_Toc144718415)

[1.17. Code snippet of Encrypt Password: 18](#_Toc144718416)

[1.18. Code snippet of CIS Update Utility: 20](#_Toc144718417)

1. **Introduction**

## Scope

This document details the technical architecture and solution implementation of the CIS Part Library Synchronization tool in two aspects. 1) One Time Mass Update and 2) Ongoing Automated Solution. This is intended for developers, administrators, architects and end users. This document don’t claim to have any contents outside the scope the part library synchronization with Teamcenter.

The following sections details the current set up for the CIS Part library updates. As of now, there is no di-rect integration with Tc and hence the synchronization is happening manually.

## Key Assumptions

* The tool shall have access to login into Teamcenter with DBA privileges.
* Ensure A&D module is installed, up and running.
* Ensure Classification and classification admin module is installed, up and running.
* A shared directory between CIS tool and CIS Part system that can be used to copy the input files from CIS Part System
* Input files will be in Proper CSV format.
* Config file will be mapped correctly with all relevant and required information’s.

## As-Is

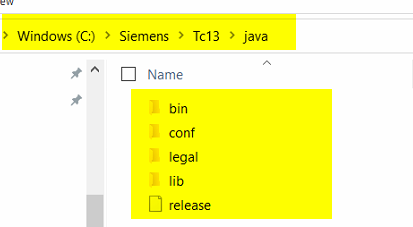
The following sections details the current set up for the CIS Part library updates. As of now, there is no direct integration with Tc and hence the synchronization is happening manually. Users’ needs to create Part manually in Teamcenter and populate the details. This will be time consuming as they need to create and update Parts manually in Teamcenter. There is not Automated system to create/update Bulk Items in Teamcenter.

## Proposed Solution:

In The proposed solution we can do Bulk item Creation/Updating in Teamcenter. Using Automation schedular we can prepare input file and paste it in staging directory, the system will automatically process it and push the Part details in Teamcenter. In the upcoming section we will be discussing high level Architecture and process flows.

## Prerequisites

* + 1. **Java Availability:**
* Check in the system has Java available in the following directory path, **“C:\Siemens\Tc13\java**”

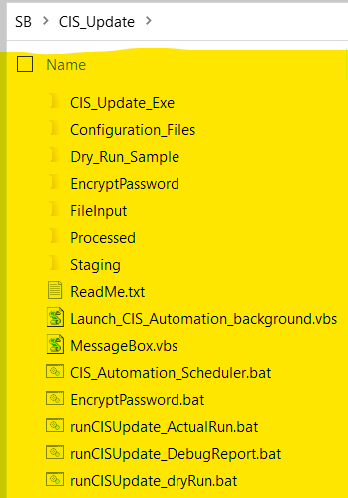


## Package Unpacking

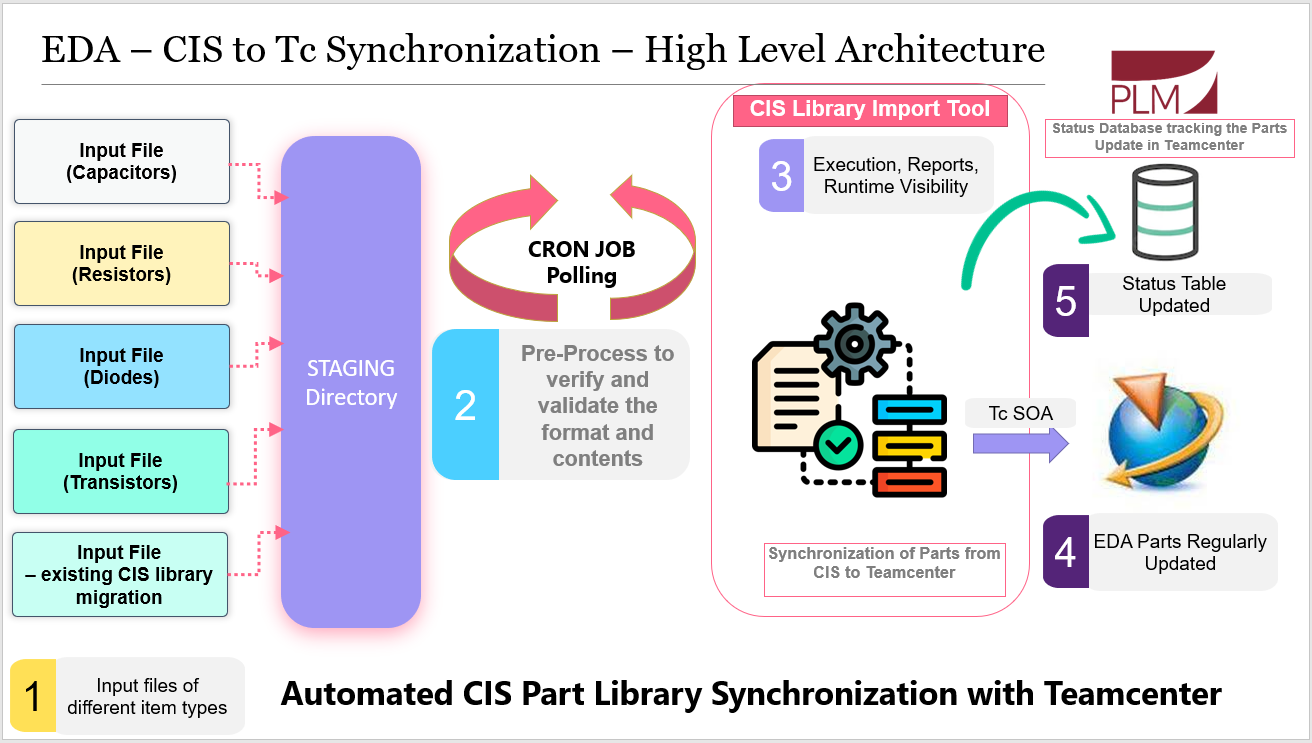
* Download the CIS\_Update.zip from repository.



* Unzip the CIS\_Update.zip into the local directory of the system.
* After, extraction the following content will be available in the extracted directory.



## CIS Update High Level Architecture:



* So, users will be preparing input files with defined format, and it will be pasted in Staging directory. CIS Automation will monitor every 5 seconds and move the files to input directory. Then the CIS Update Process will be called. Data will be populated from Input files to Teamcenter. Once CIS Update completed, input files will be moved to processed Directory. Also, respective log files will be generated.

## Input File Creation Guideline

*Input for this utility will be come from CIS Library tool in CSV format. In general, the Name & Description property from CIS extracted and given as input to update in Teamcenter.*

#### Input File Header

* The Input File Header should be in the following format. The first column should be Item ID, followed by the property to update on revision. It should be ***String.<property\_real\_name>***

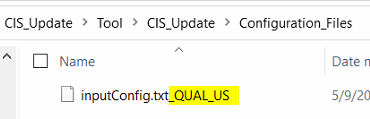


#### Input File Name

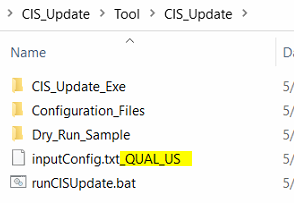
* The Input File should be a .csv file.Input File Placement
* Create a folder **“input”** inside the “**CISHomePath**” folder.
* The input file needs to be placed under the “**input”** directory.

## Preparing Configuration file:

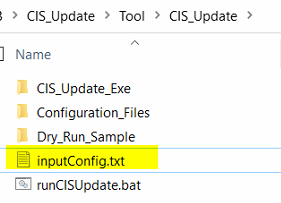
* You can use the Installation Package and use the config file from the samples.
* On Unzip directory, open the “**Configuration\_Files**” folder.
* Inside the folder, the environment specific configuration text file available. The environment name added as suffix on txt file name.



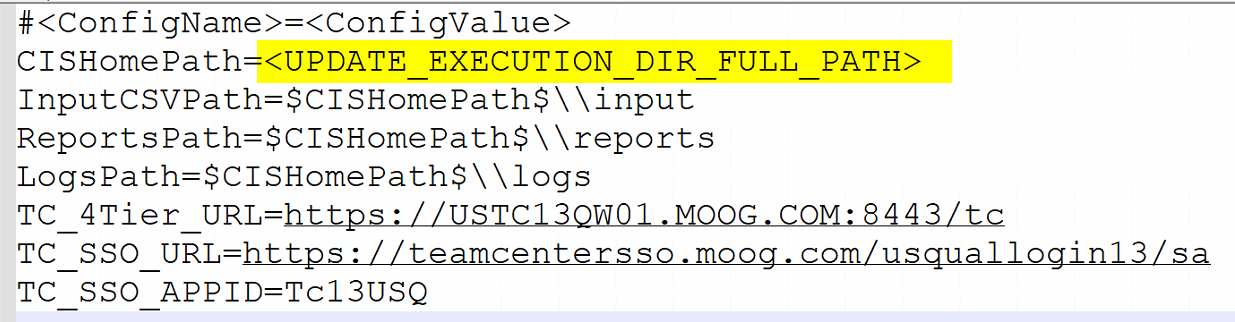
* Copy the desired environment file and paste it under “**CIS\_Update**“ folder.



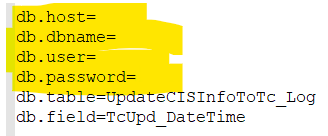
* Remove the environment name from the copied “**inputConfig.txt**” file name.



* Open the **“inputConfig.txt”** file in Text editor and the below highlighted section. CISHomePath, refers where the tool input, log, reports directory needs.

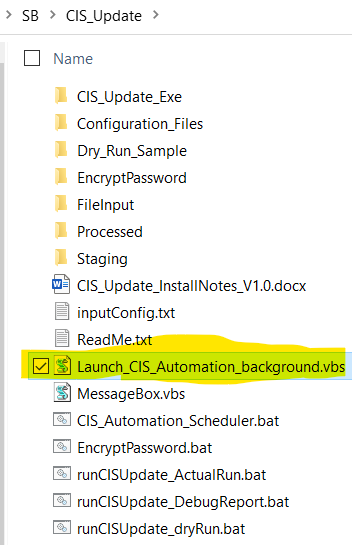


* Also update the Database host, username, password details in the config file.

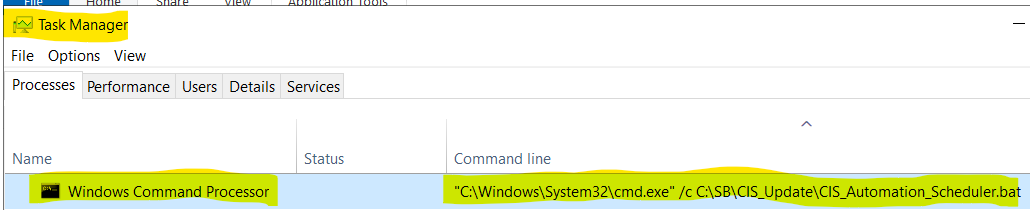


## CIS Update Automation Design:

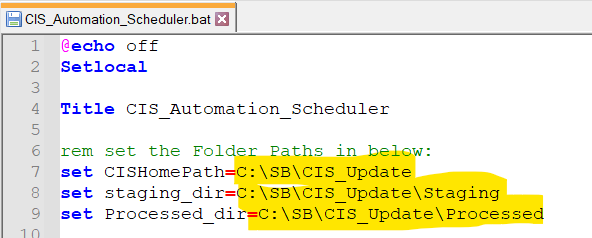
* Using “Launch\_CIS\_Automation\_background.vbs” file you can Launch CIS Update Automation in Background.
* Below highlighted script will launch the CIS Update Utility Automation in Background.



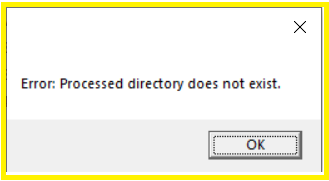
* In below image task manager, you can see that CIS Automation schedular is Launched in background.



* This script will launch “**CIS\_Automation\_Scheduler.bat”** file in background above is the screenshot.
* This Utility Requires Staging directory, CISHomePath directory and Processed directory locations to be mapped in “**CIS\_Automation\_Scheduler.bat”.**



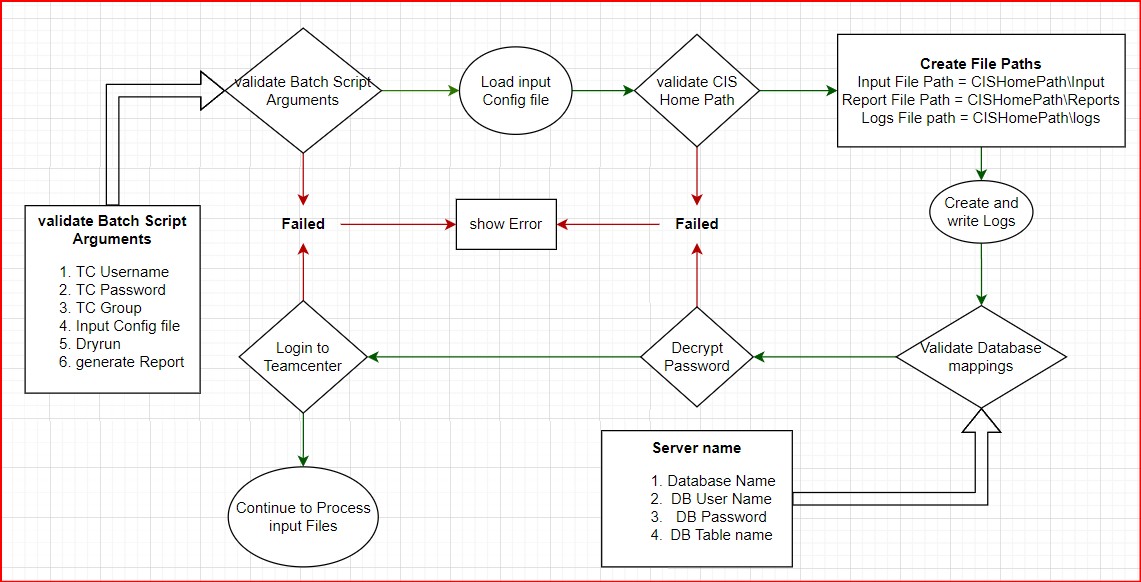
* If Staging directory, CISHomePath directory and Processed directory locations are not valid, system will show Error message and stops execution. Below image is sample error message if Processed directory is not valid.



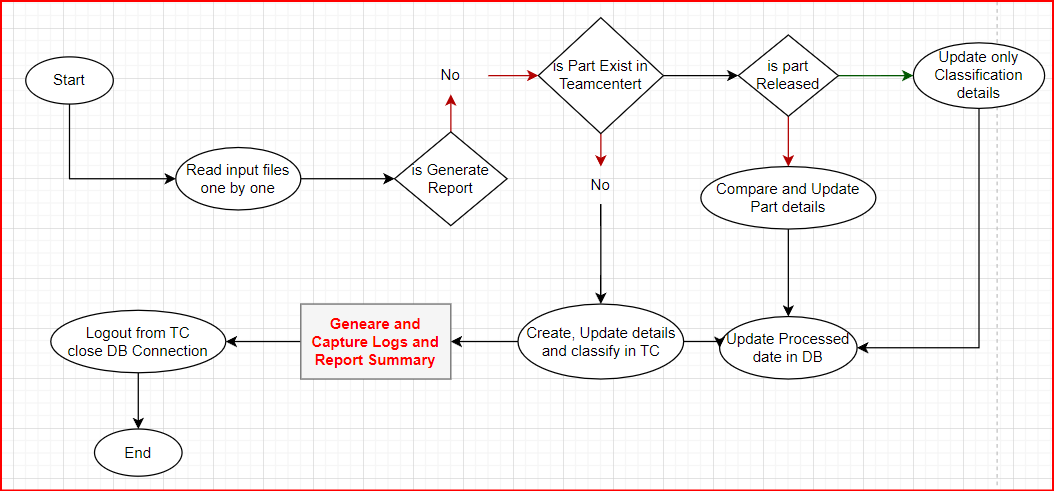
* Directory and Usage:
  + **“staging\_dir”**-> where users will be pasting the input files. (Accessible location to users and system).
  + **“CISHomePath”** -> Please provide the same location which you have provided in **inputConfig.txt** file. The system will move files from staging to the input directory for processing.
  + **“processed\_dir”** -> once the input files are processed by the CIS Update utility, these files will be moved to the Processed directory.
* Execution Logic:
  + In the Task schedular, a new schedular will be created it will be scheduled to start Launch\_CIS\_Automation\_background.vbs script.
  + Staging directory, CISHomePath directory and Processed directory locations will be mapped as explained above
  + Once users Paste Input files in the staging directory, they will be moved to the Input directory and CIS Update execution will be started.
  + Once all the Input files are processed, they will be moved to the Processed folder. (also input file names will be appended with the timestamp).
  + For more details on how Item properties are updated, please refer to Property Update Logic.
* CIS Automation schedular will check the Staging directory for any new files with every 5-second time interval.
* CIS Automation Scheduler will use runCISUpdate\_ActualRun.bat to process input files, So Actual Run configuration is required for CIS Automation, it will be discussed in next section.

## CIS Process flow Diagram:

* The following Image explains the Program flow Process of initial Validation from beginning of Program till the start of Input file Processing.

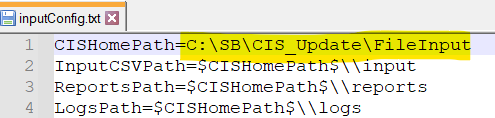


* Once the Teamcenter is logged in and Input files are exist then the next Diagram will explain how the part details are updated in Teamcenter

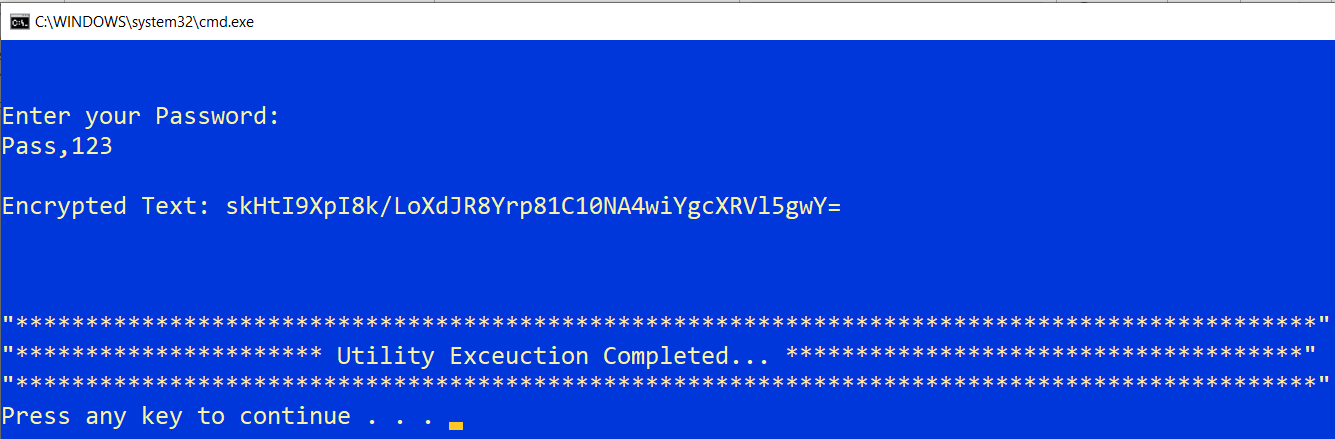


## Installation – Actual Run

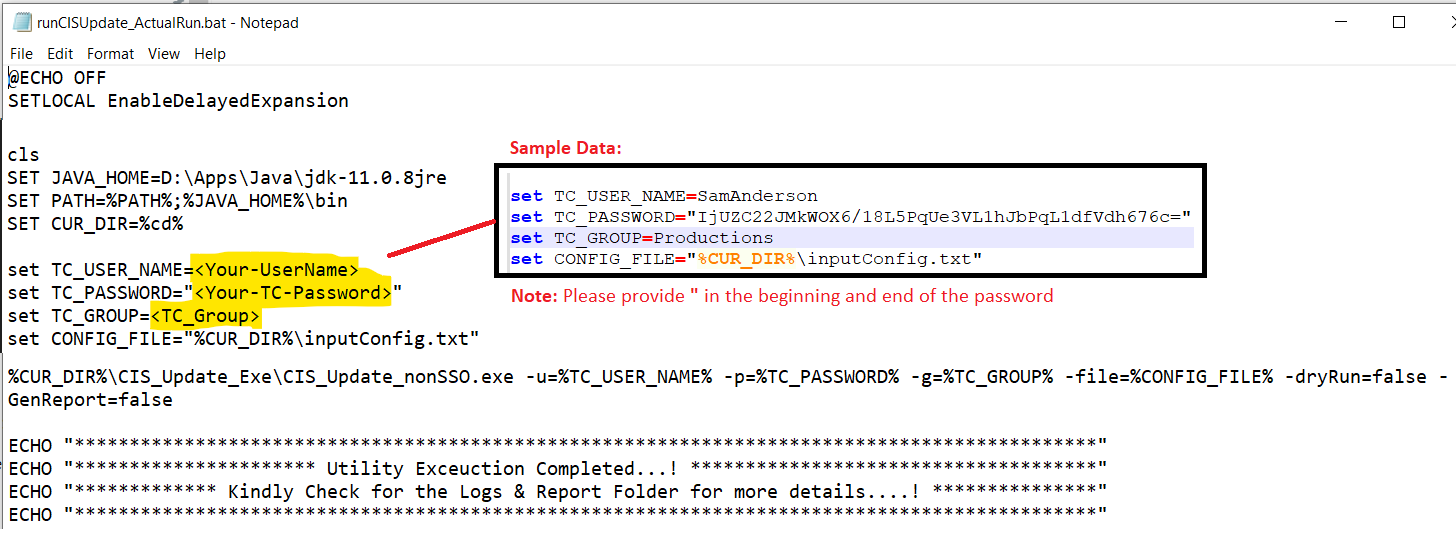
* Open the **inputConfig.txt** file and update the **CISHomePath** to **‘FileInput’** directory path.



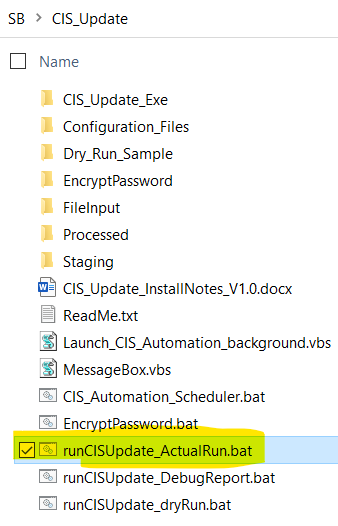
* + So here, based on the above mapping **InputCSVPath** will be C:\SB\CIS\_Update\FileInput\input. You need to provide your input CSV files in this directory.
  + Reports will be available in C:\SB\CIS\_Update\FileInput\reports.
  + Logs will be available in C:\SB\CIS\_Update\FileInput\logs.
  + For Input file samples you can use the dry run samples folder.
* Please run EncryptPassword.bat file to Encrypt your Teamcenter Password. Enter your password and **copy the Encrypted Text**.

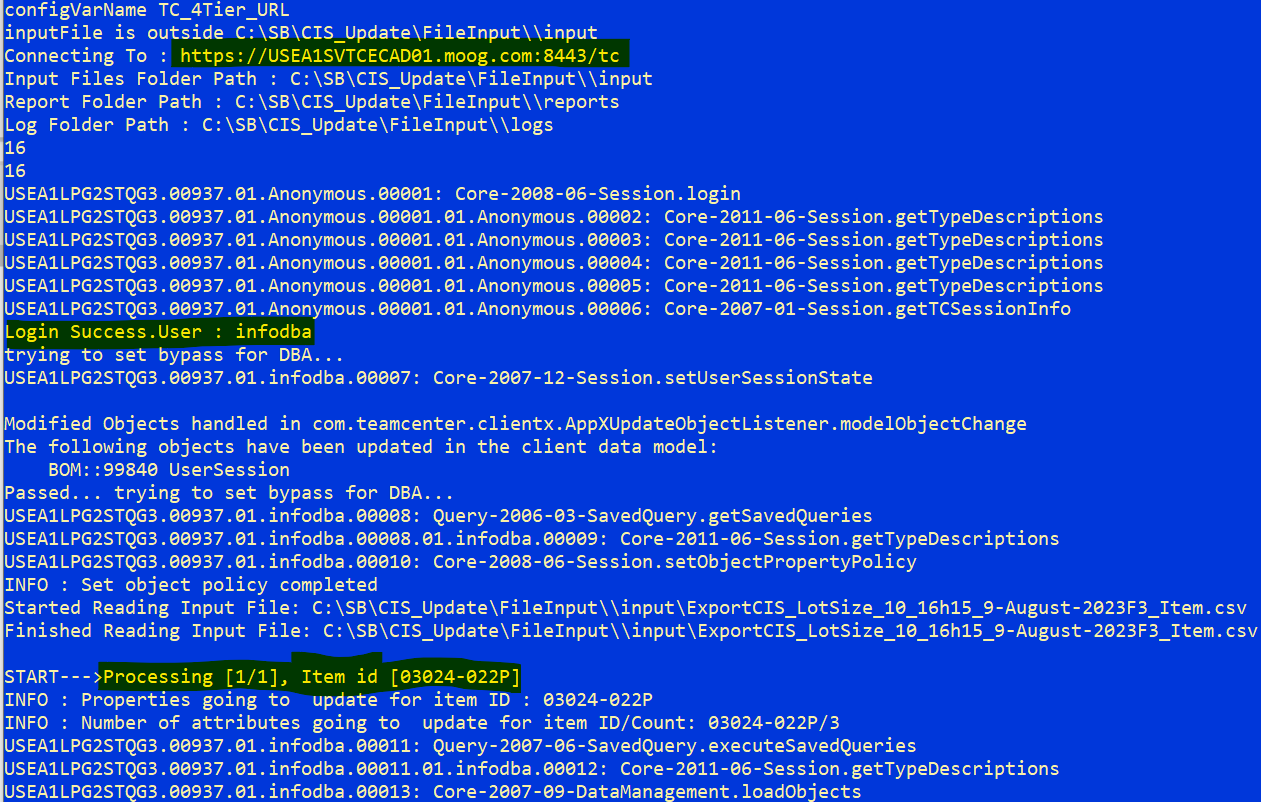


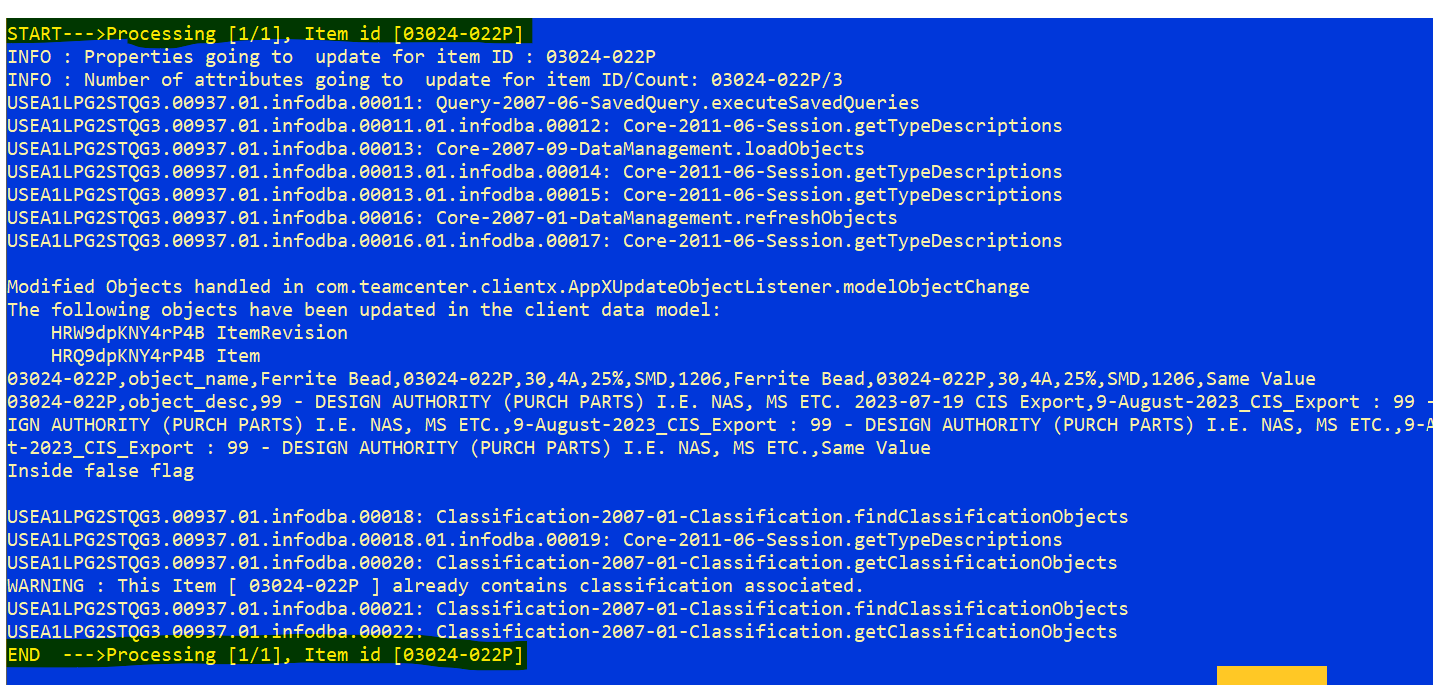
* Select file **“runCISUpdate\_ActualRun.bat” -> Right click -> Edit** and provide your Teamcenter Username, Password (Please provide encrypted password), Group as shown in below image and save it.



* Once you have proper CSV input files in input folder, Launch **“runCISUpdate\_dryRun.bat”**. On Run of batch script, the actual run will start. (Note: this will make change in Teamcenter.)







* After successful completion of batch run, the following output folder should be available inside the CSV Input path directory.



*Note: If any of the steps above failed, connect with Teamcenter Admins to resolve the issue, and proceed further on tool setup.*

## User Guide - Purpose

This document describes the entire process to update the CIS property into Teamcenter. The main purpose of the utility is to update the CIS Library information to Teamcenter.

* Input from CIS Library extracted in CSV format.
* Import the given CIS Input information in Teamcenter to respective Item Revision.

## Scope

|  |
| --- |
| Description |
| Create Input File as per the given guideline |
| Import CIS property information to Teamcenter |

Apart from the above activities, any other activity is out of scope for this work.

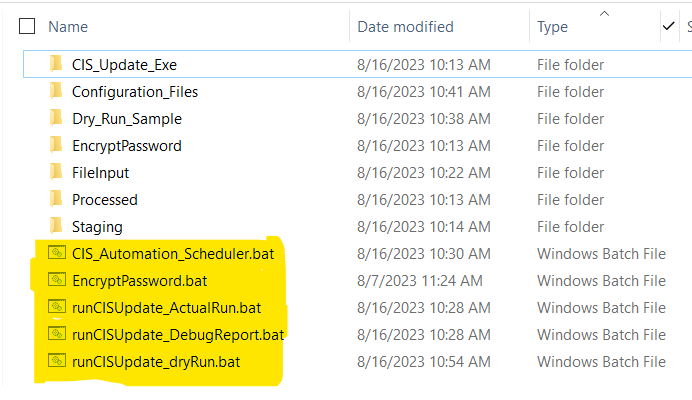
## Pre-Requisites

* CIS Update utility should be configured as per mentioned in the **section 1.5 to 1.11.**
* The CIS Update utility should be passed the Actual Run as per mentioned in the **section 1.11.**

***Note: If the utility is not configured as per mentioned guideline from section 1.1 – 1.4 tool will not give the desired result. On configuration, if any issue occurs connect with Teamcenter administrator to resolve it.***

## Execute CIS\_Update utility

* CIS\_Update utility is a command-line utility, where the necessary batch file provided with the required settings to execute the utility.
* As per this document guideline, configure the system and the CIS\_Update utility configuration.
* Create a Folder and set the folder path in “**CISHomePath**” of “**inputConfig.txt”** file. The utility logs, input, reports will be stored under this folder only.
* Also, prepare the CIS Library input file as per the given guidance.
* Utility is capable of handling more than one input file. Under **“input”** folder can have more than one input csv files.
* There are Five batch files provided to execute this utility in dry run, actual run, debug report, Password Encryption and Launch CIS Automation in Background.



* **“runCISUpdate\_dryRun.bat”** – to run the utility in dry run mode. Dry run mode does not make any changes in Teamcenter, only generates the status logs and reports on given input file.
* **“runCISUpdate\_ActualRun.bat”** – to run the utility in actual mode. Actual mode makes the changes in Teamcenter and generate the status logs and reports on given input file.
* **“runCISUpdate\_DebugReport.bat”** – to run the utility to generate Debug Report. This report will be generated in the reports folder which will have input sheet Part details from Teamcenter.
* **“EncryptPassword.bat”** – to encrypt Teamcenter Password. user need to enter the Teamcenter password, Utility will give the encrypted password. Later user needs to give this Encrypted password in runCISUpdate\_ActualRun.bat, runCISUpdate\_dryRun.bat, runCISUpdate\_DebugReport.bat
* **“CIS\_Automation\_Scheduler.bat”** – to run the utility in Automatic scheduler mode.

**Property Update Logic:**

CIS\_Utility updates the given input property value to the Teamcenter Revision as per the following logic.

* + Checks for the Item existence in Teamcenter for given Item ID.
  + If Item exists, gets latest revision of Item,
    - If Latest Revision Released, skip the update
    - If Latest Revision not Released, checks for its value with Teamcenter property value
      * If Values are same, skip update
      * If Values differs, updates the given value in Teamcenter.
  + If Item does not exist, it will create a New Item and update the property value.

**Dry Run:**

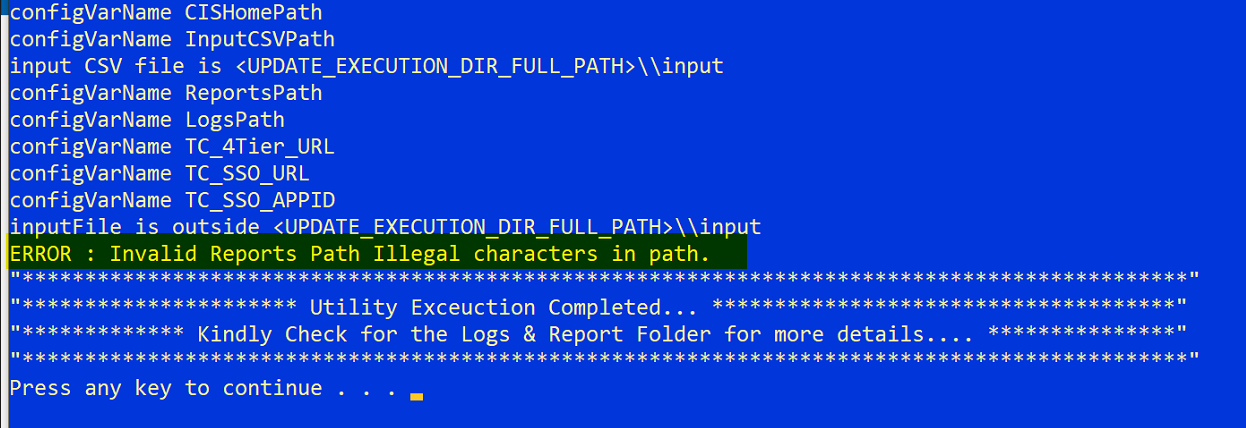
* **Dry Run does not make any data changes in Teamcenter.**
* Dry Run it is like execute, only difference is Teamcenter update will not happen.
* Dry Run generates the logs as per the Property update logic.
* It gives the summary of report on property update and revision existence in Teamcenter.

**Actual Run:**

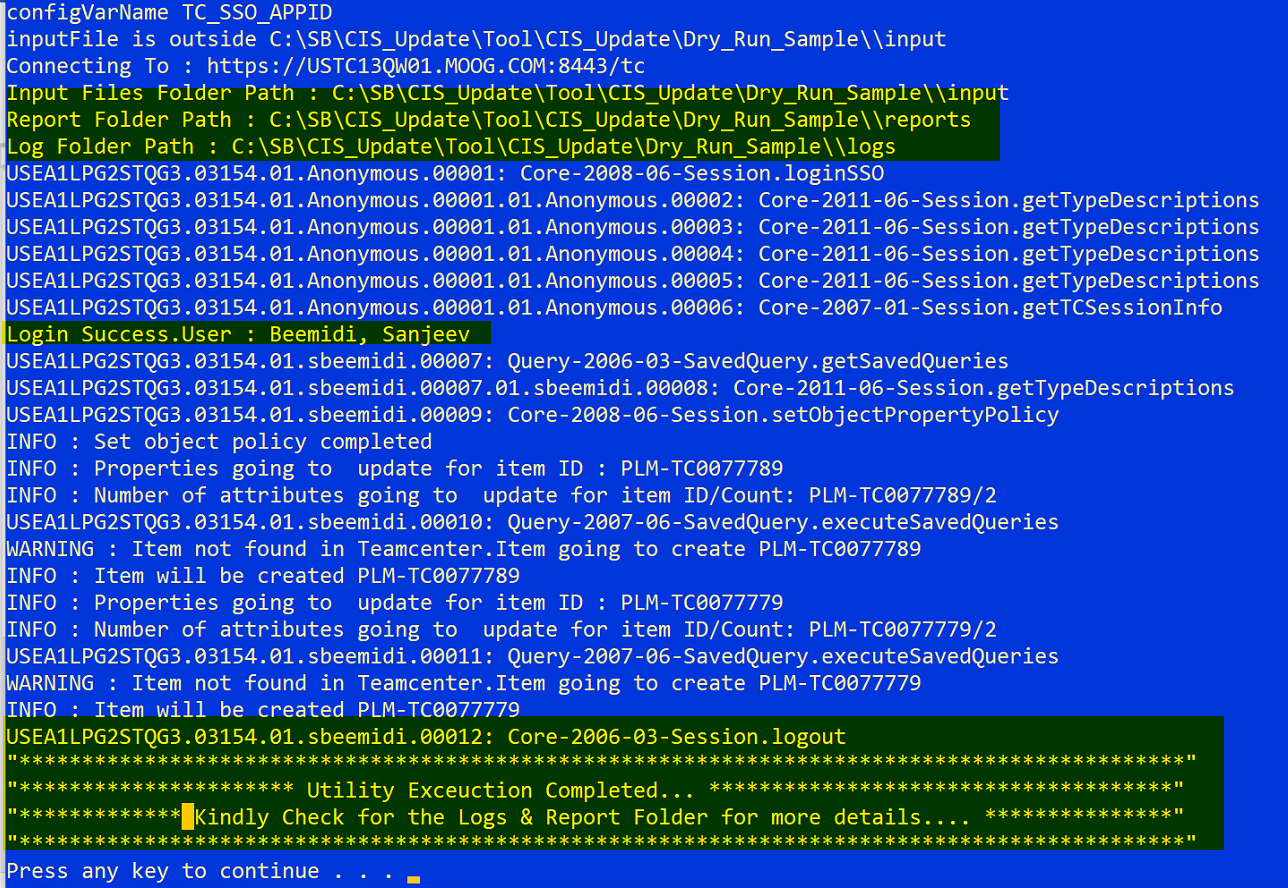
* **Execute make the data changes in Teamcenter.**
* Execute updates the given input file property in Teamcenter as per the Property update logic.
* It gives the summary of report on property update and revision existence in Teamcenter.

**Utility Logs & Reports:**

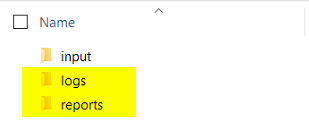
* If any error on configuration or Teamcenter login, the error will be displayed in the command prompt as highlighted below,



* On Successful completion of utility, the command prompt displays completion of utility. As per the below snapshot the highlighted lines ensure the successful completion of utility.

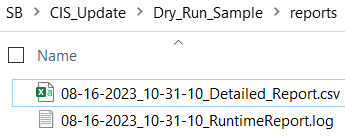
****

* The console displays, the path for Log File, Report File, Logged in User for Teamcenter and the Logout of Teamcenter session.
* Utility reports & logs created in respective folder under **CISHomePath** directory. Inside, reports and logs folder for each input file under input folder, will generate corresponding its report and logs file.

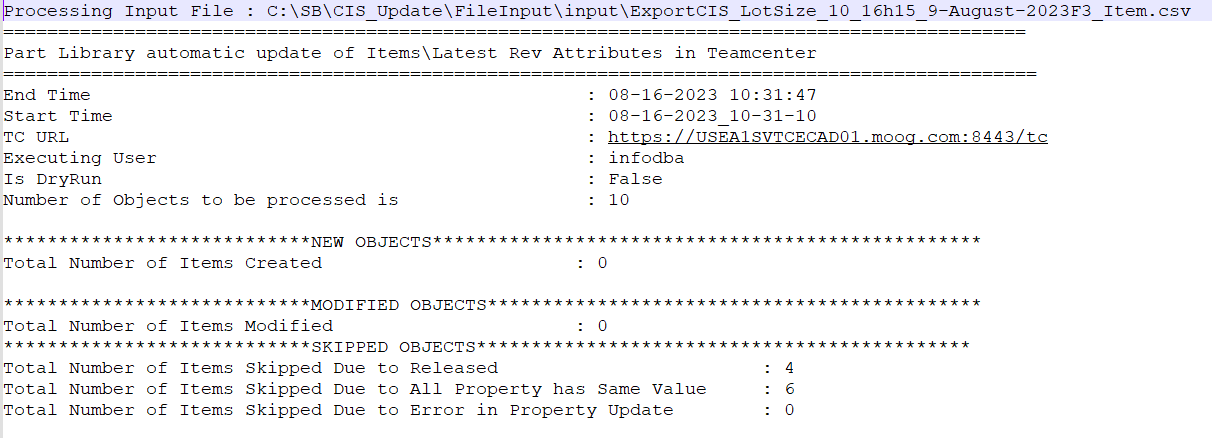
****

**Reports File:**

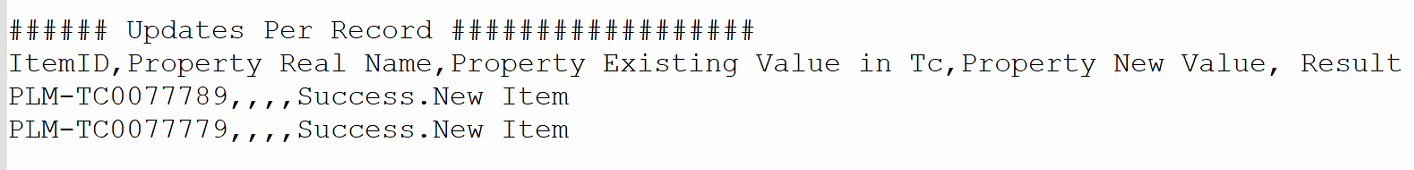
* **The report file useful to verify quickly on the status of property update in Teamcenter.**
* Report files gives the summary of property update for the given files. The report file available gives the summary of property update and its values update in detail.

****

* Below is the sample report summary, where gives the TC server environment the utility runs against, TC user used for run, utility mode (dry-run/update), No. of inputs given and its update status in Teamcenter.

ss

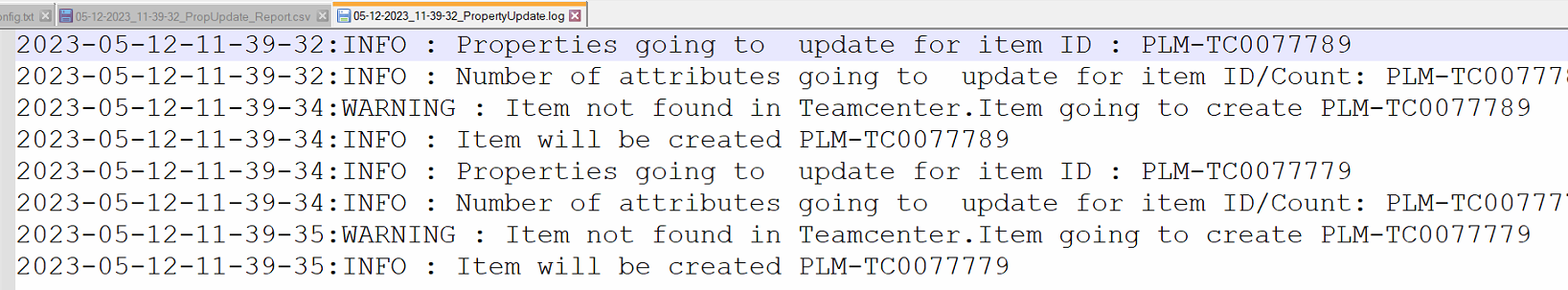
* The second part of the report will have the details of Teamcenter object status, value comparison against given input and its process state.



* Below are the possible available Result value and its meaning,
  + **Success.New Item** – New Item Created for given input
  + **Success** – The Property Value updated on the revision for given input
  + **Not Changed** – The Property values same on the revision for given input. Not Modified in Teamcenter.
  + **RELEASED.SKIP** – The Latest Revision is Released. Not Modified in Teamcenter.
  + **Failed** – Failed to Create New Item / Update the property in Teamcenter.

**Logs File:**

* The Log files gives the detail processing log for input file line by line.
* This log contains detail of update property information as per below snapshot. If any technical error during the process, the detailed error information will be available in this log.

****

* Thislogs mostly can be used for issue analysis purpose.

**Assumptions:**

* + Input file and tool configuration as per the given guideline. Otherwise, utility may fail.
  + JDK / JRE must be available in system for tool execution.
  + Teamcenter Security Agent must be available in system for Teamcenter login.
  + Needed input must be provided in the config file (TC-server URL, SSO URL, Tc-APPID)
  + When Teamcenter has connection issues, this related error details available in that Logfile.
  + We must proceed Dry run for validation before actual execution.
  + In another case If Input file got some error issue, can move with Input validation.
  + If parts are missing Teamcenter, its needed to add those missing parts manually.
  + During the testing, all given scenarios are must referable.
  + Item ID play’s important role in input.
  + Only the latest revisions of available item revisions will be considered.

**Testing Scenarios:**

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Description** | **Result** |
| 1 | Item ID Not Exist in Teamcenter | Creates new Item and updates property |
| 2 | Item ID Exist, Latest Revision Released | Skip the update |
| 3 | Item ID Exist, Latest Revision WIP, Values same | Skip the update |
| 4 | Item ID Exist, Latest Revision WIP, Values differs | Update the Value |

**Testing Assumption:**

* The below are the brief of various testing scenario of CIS Property update.

**Scenario-1:**

* This scenario represents, the given Item ID not found in Teamcenter.
* On this, new Item will be created in Teamcenter for the given ID and updates the property value on the newly created revision.

**Scenario-2:**

* This scenario represents, the given Item ID found in Teamcenter. But the Latest Revision is in Released State.
* On this, the property update will be skipped.

**Scenario-3:**

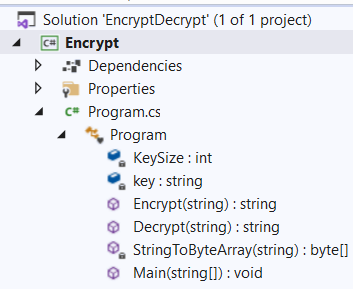
* This scenario represents, the given Item ID found in Teamcenter. The Latest Revision is in WIP state. But the input file values and Teamcenter property values are same.
* On this, the property update will be skipped.

**Scenario-4:**

* This scenario represents, the given Item ID found in Teamcenter. The Latest Revision is in WIP state. But the input file values and Teamcenter property values are different.
* On this, the given property will be update in Teamcenter.

## Code snippet of Encrypt Password:

* Following code snippet is used to Encrypt the Users Password. Which will be given as input for Teamcenter Password in Dry-Run / Actual-Run batch scripts.
* The below code snippet contains both Encryption and Decryption code. Currently from the following program we are using Encrypt function to Encrypt user passwords and the Decrypt function to Decrypt user's Password in CIS Update

****

.

****

## Code snippet of CIS Update Utility:

* Following code snippet is used for CIS Update Utility.

Property\_Update.cs:

****

**Property\_Update\_Pojo.cs:**

****