Wickers, Jason (DI SW GS&CS GS&S LCS)

jason.wickers@siemens.com



**Developer Guide**

Teamcenter REST API and Requirements Integration for MATLAB/SIMULINK

Abstract

This document will provide a brief introduction to establishing a development environment to improve the Teamcenter Requirements integration for MATLAB/SIMULINK

**Disclaimer**

This document is intended to provide Teamcenter deployment information, and related observations and recommendations. Siemens Korea Digital Industries is providing this information as is, without warranty of any kind. **Siemens Korea Digital Industries hereby disclaims and assumes no responsibility or liability for any results that occur due to the use of the information contained in this document.**

All product designations may be trademarks or other rights of Siemens AG, its affiliated companies or other companies whose use by third parties for their own purposes could violate the rights of the respective owner.

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| .1.2 | 2022-08-31 | Wickers, Jason | Documentation for initial release |
| .1.3 | 2022-09-14 | Wickers, Jason | Updated with Credentials, teamcenter and scripts folder, SSL and Known Issues/Workarounds |
|  |  |  |  |

Table of Contents

[Introduction 3](#_Toc114064092)

[Installation 3](#_Toc114064093)

[Pre-Requisites 3](#_Toc114064094)

[Obtain Source Code 3](#_Toc114064095)

[Create and Activate Virtual Environment 3](#_Toc114064096)

[Install PIP Packages 3](#_Toc114064097)

[Install Siemens Package for Python 4](#_Toc114064098)

[Teamcenter Environment Folder and Scripts 4](#_Toc114064099)

[Credentials 4](#_Toc114064100)

[Secure Sockets Layer (SSL) 5](#_Toc114064101)

[Install Python for Visual Studio Code 9](#_Toc114064102)

[Debugging Python Code 10](#_Toc114064103)

[Code Layout 10](#_Toc114064104)

[Build, Install, and Distribution 10](#_Toc114064105)

[Known Issues / Workarounds 11](#_Toc114064106)

# Introduction

Development of the Teamcenter Requirements integration for MATLAB/SIMULINK is completed primarily in Python. Any MATLAB code is primarily for application glue and wrapping around the Python components of the integration.

# Installation

## Pre-Requisites

This document assumes python has already been installed per the Install Guide and you have obtained and installed:

* Visual Studio Code: <https://code.visualstudio.com/>

## Obtain Source Code

Source code is available as zip or can be cloned from Siemens’ GITLAB site here:

* <https://code.siemens.com/jason.wickers/resttc>

The location of the extracted source / cloned site will be the development workspace **<DEV\_WORKSPACE>**

### Create and Activate Virtual Environment

1. Open command prompt
2. Change directory: cd /d **<DEV\_WORKSPACE>**
3. Execute the below command within your **<DEV\_WORKSPACE>**

python -m venv venv

The result will be a virtual environment called “venv” within your **<DEV\_WORKSPACE>**

Type the below command to ACTIVATE your virtual environment while within your **<DEV\_WORKSPACE>**:

venv\Scripts\activate.bat

Note:

* If you want to DEACTIVATE you can type: venv\Scripts\deactivate.bat

### Install PIP Packages

In this section you will install packages from Python’s standard package manager. The packages being installed are required dependencies of the integration.

Run the following from the command prompt in the ACTIVATED **<DEV\_WORKSPACE>**:

|  |
| --- |
| pip install certifi  pip install requests  pip install inflection  pip install bs4  pip install cryptography |

## Install Siemens Package for Python

1. Navigate to venv\Libs\site-packages
2. Create file: teamcenter.pth file
3. Add absolute path to the **<DEV\_WORKSPACE>** (ex: D:\dev\github\resttc) to the teamcenter.pth file and save.

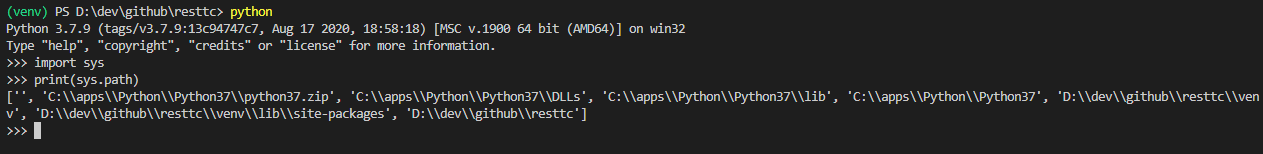
Next, type the below command to launch python in an ACTIVATED **<DEV\_WORKSPACE>**

python

With python loaded, type the following into the interpreter:

|  |
| --- |
| import sys  print(sys.path) |

Example result with the absolute path to source code folder:



## Teamcenter Environment Folder and Scripts

1. Create folder venv\**teamcenter**
2. Copy logging.ini from **<DEV\_WORKSPACE>** into venv\teamcenter
3. Copy tcaliases.ini from **<DEV\_WORKSPACE>** into venv\teamcenter
4. Copy tc\_credentials.py from **<DEV\_WORKSPACE>**\Scripts into venv\Scripts

## Credentials

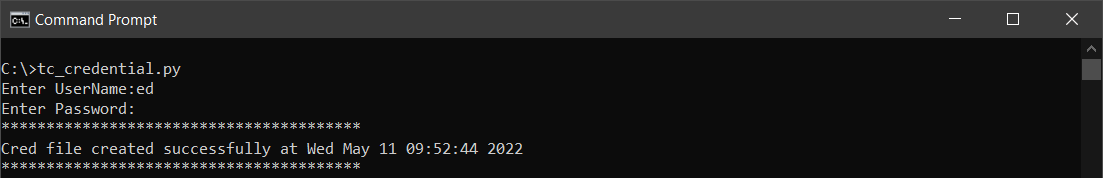
The integration requires credentials to be pre-defined for each alias currently. To do this, a credentials script exists. You may think of this like creating a password file for Teamcenter.

Credential files are stored in: venv\teamcenter

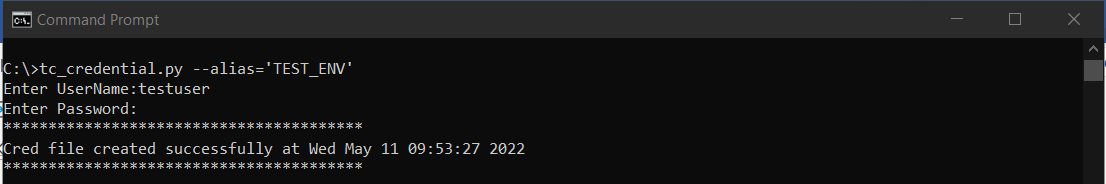
Usage: tc\_credential.py [options]

|  |
| --- |
| Options:  -h, --help show this help message and exit  -a ALIAS, --alias=ALIAS  alias credential is for |

Example 1:



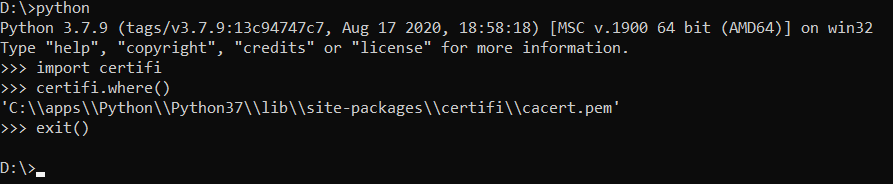
Example 2:



## Secure Sockets Layer (SSL)

The integration has SSL Verification enabled. Therefore communication via HTTPS requires the chain of ssl certificates for your active workspace url and added to cacert.pem.

To find the location of cacert.pem, you can do the following:



Next:

1. Using your web browser (Edge Chromium, Chrome, etc.), download the chain of certificates used in your Teamcenter installation and specified in your aliases.ini (see HOST) and save as Base64 encoded .cer files.

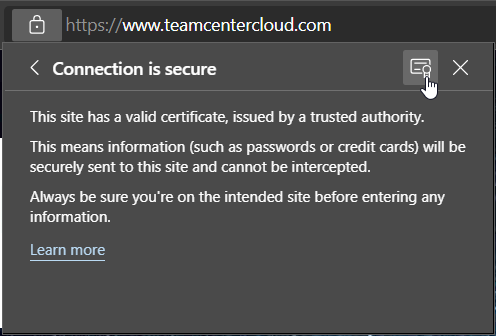


Figure Edge Chromium start save of certificate

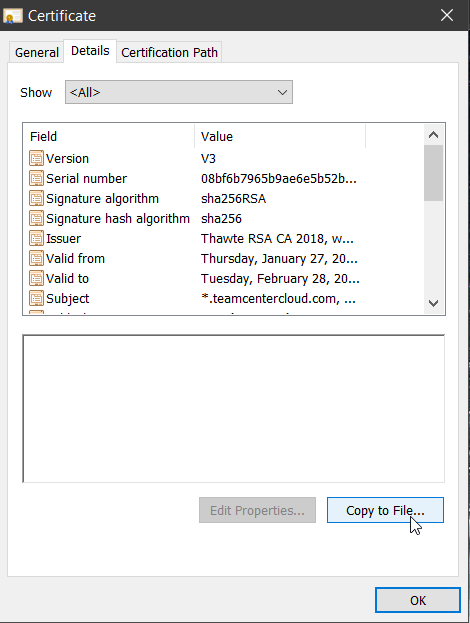


Figure Where to go to save the certificate to file

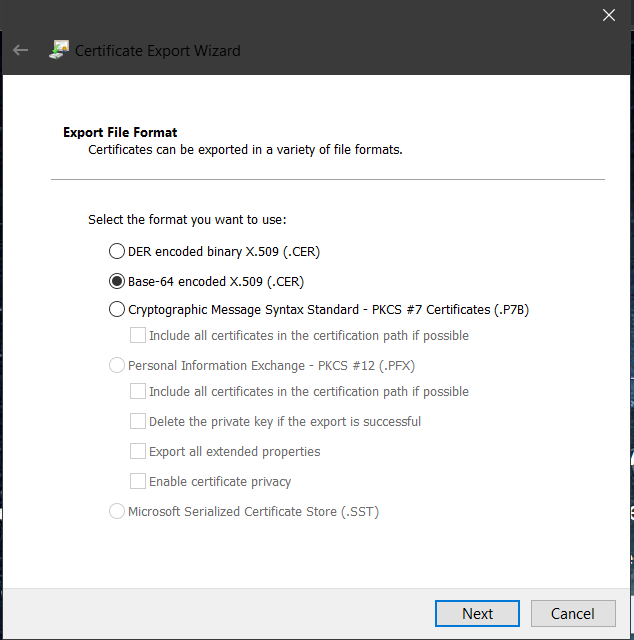


Figure Export file format should be Base-64

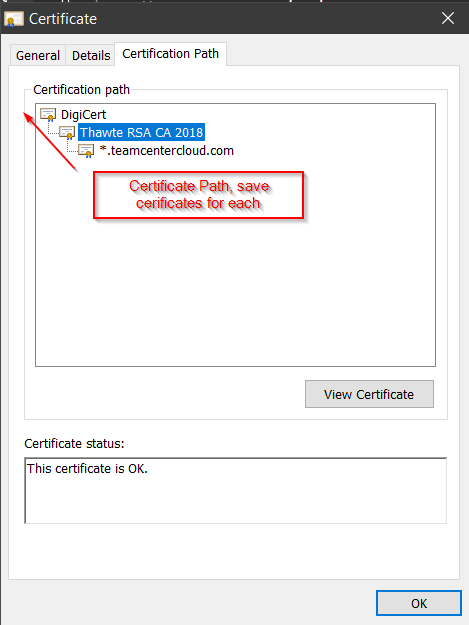


Figure Click on each certificate path level and "view certificate" to save each certificate as file

Open the cacert.pem in a notepad and add every downloaded certificate contents (---Begin Certificate--- \*\*\* ---End Certificate---) at the end. Do not remove the ---Begin Certificate--- header or ---End Certificate--- footer, it must be included.

## Install Python for Visual Studio Code

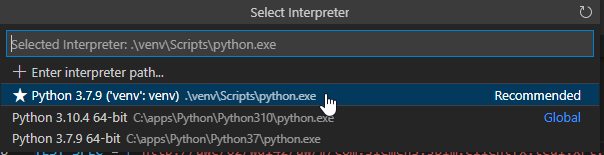
1. From command prompt within the **<DEV\_WORKSPACE>**, open VS Code:

|  |
| --- |
| code . |

Alternately, you can run VS Code through the operating system UI, then use **File > Open Folder** to open the **<DEV\_WORKSPACE>** folder.

1. Click on the Extensions command 
2. Search for Python (by Microsoft) and Install

From within VS Code, select the **<DEV\_WORKSPACE>** Python 3 interpreter for the venv by opening the **Command Palette** (Ctrl+Shift+P), start typing the **Python: Select Interpreter** command to search, then select the command.



### Debugging Python Code

Please refer to the following tutorial for how to debug Python code with VS Code.

[Get Started Tutorial for Python in Visual Studio Code](https://code.visualstudio.com/docs/python/python-tutorial#_configure-and-run-the-debugger)

# Code Layout

The **<DEV\_WORKSPACE>** has the following basic layout:

docs – documentation directory

examples – simple examples for how to use resttc

pytests – unittests for resttc

scripts – scripts to support resttc usage when installed

siemens – the main module containing resttc python code

siemens/icons – icon files (16x16 png)

siemens/matlab – the matlab scripts to support the SIMULINK integration

# Build, Install, and Distribution

**Build (will build the resttc package underneath 'build/'):**

python setup.py build

**Install (installs the resttc package to the active python environment):**

python setup.py install

**Distribution (create an installer):**

python setup.py bdist\_wininst

# Known Issues / Workarounds

1. **Running python -m unittest in DEV\_WORKSPACE\pytests has an error**
   * *Description:* Running all the pytests together results in an error in the test\_integration.py
   * *Workaround:* Run test\_integration.py unittest by itself to confirm ok