**SamplePro API Tasks**

**Author: Wayne Schou**

**Date: 30/8/2018**

****Overview****

**Multiple scheduled tasks have been created to regularly check for certain conditions in the SamplePro system using the FreezerPro API interface and email the relevant owners, operations officer or support person.**

**The tasks currently implemented are:**

* **Email Operations Officers about Requests**
* **Email Approval Contact about Requests**
* **Email Owner about State Changes**
* **Email Owners about Samples nearing Review Date**
* **Email Owners about Samples with Review Date overdue**

****Email Operations Officers about Requests****

Email members of OperationsOfficer user group a list of sampleids and barcodes that have state

'Retrieve Request', 'Store Request', 'Dispose Request', 'Return to External Owner', 'Approval Requested'. An email will not be sent if there are no samples in any of these states.

****Email Approval Contact about Requests****

Email user in "Approval Contact" udf of sample about samples with “Approval Requested” status

Any samples with an invalid "Approval Contact" (in state “Approval Requested”) will be sent to OperationsOfficer group. No email will be sent if there are no samples in “Approval Requested” state.

****Email Owner about State Changes****

Email owners of samples of any changes made by others to sample status occurring today. No emails will be sent if no such state changes have been made.

Note: Any changes made after last task run for a given day will not be notified.

****Email Owners about Samples nearing Review Date****

Email owners of samples when samples are within 30 days of Review Date. Will not report on samples where ANY sample vial/location has a Disposed or Dispose Request state

****Email Owners about Samples with Review Date overdue****

Email owners of samples any samples where Review Date has passed. Will not report on samples where ANY sample vial/location has a Disposed or Dispose Request state

****Installation****

1. **Install python 3.6**

**Note: current installation uses 64bit and has been installed for all with changes to path env variable)**

1. **Copy python scripts to C:\SamplePro\Source, required packages to C:\SamplePro\Requirements (see step 5) and Scheduled tasks to C:\SamplePro\ScheduleTasks.**

**Note: C:\SamplePro\ScheduleTasks contains xml files and bat files with hardcoded paths to C:\SamplePro.**

1. **Create virtual environment (to isolate versions of required packages to just this usage)**

**python -m venv C:\SamplePro\VirtualEnv**

**Note: If a later version of Python is installed (in addition to 3.6) then care must be taken with the following commands to ensure when creating the virtual environment that python version 3.6 is used (either use a full path to python 3.6 exe or if the python launcher has been installed replace python in the commands above with py -3.6.). Once virtual environment is activated then “python” will reference the correct python version.**

**Note: Folder C:\SamplePro\VirtualEnv now contains a copy of python 3.6.**

1. **Activate virtual environment (using command prompt)**

**C:\SamplePro\VirtualEnv\Scripts\activate.bat**

**Note: There is also a powershell script (C:\SamplePro\VirtualEnv\Scripts\Activate.ps1) if preferred.**

1. **Install required packages (without internet connection)**

**python -m pip install -r C:\SamplePro\Source\requirements.txt -f C:\SamplePro\Requirements --no-index**

**Note: As this server does not have access to the internet the required packages were download on another computer (with python 3.6 installed) using the command**

**“python –m pip –r \path\to\requirements.txt –d \path\to\downloadfolder” – the wheels downloaded where then copied to C:\SamplePro\Requirements.**

**Note: If this computer did have access to the internet the process of installing the required packages would have be simplified to “python –m pip –r C:\SamplePro\Source\requirements.txt”.**

1. **Save SamplePro account password**

**Before running ensure that the username stored in config.ini has the username of the account that can access SamplePro via API (currently set to FreezerLDAP) (see configuration below for details on config.ini)**

**This command should be run from the same account that will be running the scheduled tasks.**

**Password is stored in the Windows Credential Locker.**

**python C:\SamplePro\Source\set\_password.py**

**Note: Alternate methods for storing password**

**a. Run commandline utility keyring (installed in python Scripts folder) e.g. keyring set FreezerPro FreezerLDAP**

**b. Run keyring module e.g. python -m keyring set FreezerPro FreezerLDAP**

1. **Deactivate virtual env (using command prompt)**

**C:\SamplePro\VirtualEnv\Scripts\deactivate.bat**

1. **Create Tasks in Task Scheduler**

**Import each of the xml task descriptions in C:\SamplePro\ScheduleTasks (there are 5 separate tasks) using Task Scheduler | Action | Import Task…**

**Note: Currently the tasks have been loading into a SamplePro folder and are run using the schouw account. If this account is changed then will need to rerun step “6. Save SamplePro account password” from the new account.**

**Note: The tasks reference bat files in C:\SamplePro\ScheduleTasks**

****Expected FreezerPro database settings****

* **OperationsOfficer group**
* **Following states exist (names exactly as specified):**
  + **“Store Request”**
  + **“Approval Requested”**
  + **“Store Request Approved”**
  + **“Stored”**
  + **“Retrieve Request”**
  + **“With Owner”**
  + **“Dispose Request”**
  + **“Disposed”**
  + **“Received”**
  + **“Return to External Owner”**
  + **“Returned”**
* **Following states must have specified ids:**
  + **“Store Request” 4**
  + **“Retrieve Request” 5**
  + **“Return to External Owner” 6**
  + **“Dispose Request” 8**
  + **“Disposed” 9**
  + **“Approval Requested” 10**
  + **“Store Request Approved” 11**
* **Sample has user-defined field: “Approval Contact” containing full name of existing user in the system. (Used by EmailApprovalContactAboutRequests task)**

****Configuration****

**There are several configuration parameters that can be adjusted. These are located in C:\SamplePro\config.ini.**

**[FreezerPro]**

**username = FreezerLDAP**

**api\_url = https://freezerpro.scionresearch.com/api**

**[MailServer]**

**smtpserver = 163.7.18.150**

**smtpport = 25**

**[System]**

**support\_email = jenny.simpson@scionresearch.com**

**email\_support\_only = True**

USER\_NAME: is the username used to access the FreezerPro API – this should match the username stored Windows Credential Locker in Step 6 of the installation process.

API\_URL: is the url to the FreezerPro API

SMTPServer: address of the mail server

SMTPPort: port of the mail server

SUPPORT\_EMAIL: email to use when issues arise in the system.

Email\_support\_only is an optional key, if present and true then all emails will be redirected to the support\_email.

Config.ini must exist.

A default config.ini can be hand-crafted or can use create\_config.ini.py.

****Comments****

**Ideally these emails should be sent as soon as the corresponding condition is met but FreezerPro does not contain an event triggered mechanism that we can hook in to. The alternate approach taken here requires regularly checking of all states (not just changes) and has the potential to send out duplicate emails to users if they have not yet acted on the previous email. To avoid duplication of emails would require some additional coding (plus a database) to record which changes had already been sent. Most emails are about conditions that are transitional and should not be occurring for long periods.**

**One limitation of the EmailOwnerAboutStateChanges and EmailOperationsOfficersAboutRequests tasks is that they rely on the FreezerPro API “audit()” which has a maximum record limitation of 1000. Any day where the number of audit records produced exceeds 1000 will mean that not all state changes can be returned and will not be acted upon. At present if the number of audit records exceed 1000 an email is set to the Support email and no email is set to intended recipients. Potential workaround is to make multiple calls to audit using the optional “start” parameter.**

**Email Owner About State Changes: Because the check only reports on changes made ‘today’ there should be a task run at the end of the day i.e. 11.59pm to ensure that all changes are reported to the owner. Alternately, there should be a separate check reporting on changes for ‘yesterday’.**

**Rather than having the Support email hard coded in the config.ini, another group could be added to SamplePro that contains the emails to use. The issue with this approach is that if access to SamplePro fails (e.g. Authentication failure) then you also lose the ability to retrieve the support email. A similar issue does exist with placing details if the config.ini as any issues with these settings will also fail to email support.**