

FAST NUCES – Fall 2017 IPT – Assignment 3

Due on: Monday 20th November 2017 by 17:00 hours

Please submit via SLATE. Please include a single .zip/.rar file (source code only, no binaries) and name your submitted file as "Assignment3IPT<your reg #>"

1. Create a windows service that reads an input xml/json file that specifies certain tasks, carries out those tasks and puts the result in an output xml file.
2. The windows service should read the name & location of the input xml/json file as well as name and location of the output file from the settings/config file.
3. For now, there are three tasks to be supported:
 - a. Secure backup of one important file by making two parts of the file such that first file has all odd numbered bytes of the original file and second file has all even number bytes of the original file.
 - b. Secure restore of on important file via vice versa of above task.
 - c. ~~Take a screen grab and store it in a designated place~~ Take a list of all processes currently executing together with their memory footprint (Private Working Set) and CPU time.
4. The structure of the input file is:

```
<Tasks>
  <Task name="{name of task}" type="{type of task}">
    <Occurrence starttime="{start date & time of task}" repeat="{none a valid timespan}"
      endtime="{end date time if repeat is not none}" />
    <{Task type specific details} {task type specific attribute}="{e.g
C:\ImportantFile.pst}" {another task type specific attribte}="{e.g e:\outputfolder}" />
  </Task>
</Tasks>
```

e.g.

```
<Tasks>
  <Task name="Backup PST File" type="SecureBackup">
    <Occurrence starttime="20171103 22:00" repeat="1d" endtime="20180331 23:00" />
    <SecureBackup src="c:\Importantfile.pst" dest="e:\outputfolder\" />
  </Task>
  <Task name="9 to 5 Screen grab" type="ScreenGrab">
    <Occurrence starttime="20171103 09:00" repeat="1h" endtime="20180131 17:00" />
    <ScreenGrab outputFolder="d:\ScreenGrabs" />
  </Task>
</Tasks>
```

5. The output should be in the following format. The TaskResult node is specific to the task type and could have a different and more complicated structure:

```
<Results>
  <TaskRun taskname="{name of task}" type="{type of task}" time="{date & time of run}">
    <TaskResult result="{resultant value}" />
  </TaskRun>
</Results>
```

6. The service should report any errors in the application event log.

Design Objectives

- A. The input/output should be decoupled from main Service code
- B. The design should be flexible enough so that adding a new type of task does not require wholesale changes in the service code.