Contents

[**QUICK SETUP** 2](#_Toc21441232)

[**KEY NOTES ABOUT USAGE** 3](#_Toc21441233)

[**CORE FILES** 4](#_Toc21441234)

[**MODULES** 5](#_Toc21441235)

[**CONFIGURATION OPTIONS** 6](#_Toc21441236)

[ANONYMIZATION 6](#_Toc21441237)

[DICOM HEADERS / USER 6](#_Toc21441238)

###############################################

# **QUICK SETUP**

###############################################

1. **Install the latest version of Python 3**
   1. \*If given the option/checkbox during installation, choose to YES, include python in the path
   2. <https://www.python.org/downloads/>
   3. To verify correct installation:
      1. Open the command prompt (find it by opening the windows search bar in the bottom left of your screen and doing a text searching for “CMD”)
      2. In the command prompt, type the text “python” and press enter. If python did not install properly or is not in the appropriate path, it will give you an error message.
2. **Download the following files** and place them into your folder of choice
   1. main.py
   2. myGUI.py
   3. pacs.py
   4. requirements.txt
3. **Install the prerequisite python modules** - open the command prompt, navigate to your folder, and run the command:
   1. pip install -r requirements.txt
4. **Run the program** by double clicking on “main.py”
   1. The program will create any necessary folders and config files, which can then be modified by the user.

###############################################

# **KEY NOTES ABOUT USAGE**

###############################################

1. **Try not to use the “\*\*\*ALL XXX” option in the study/series filter window**
   1. It does what it was designed to do, but it just does it un-intuitively. Need to rework the concept.
2. **You can use your own query IDs/query numbering if you like\*** – just specify one of the following column names: “Query”, “ID”, “Index”. The program will keep track of which results belong to which query.
   1. \*If no specific query ID column is provided, these query-result associations will still be tracked, but just with auto-generated numbering of the input queries
3. **“StudyID” is NOT equal to “StudyInstanceUID”**
   1. “StudyInstanceUID” is the one you want
   2. “StudyID” exists as a tag but is not regulated by the DICOM standard
4. **Do NOT specify StudyTime** in your query identifiers unless you know exactly what you are doing
   1. **study time is NOT equal to the report time**
      1. If you are using the date of the report as a stand-in for the date of the actual study, then that would cause you to miss any dates where the study and report date are different. E.g. a study is done at 11:45PM Monday night but not actually read/dictated until Tuesday morning.
   2. **The times/datetimes listed on Phillips PerformanceBridge-outputted excel sheet do NOT correspond to the actual study time recorded in the metadata of the DICOM image**
      1. NONE of these following PBP-outputted columns have the correct time: “Report Datetime”, “Begin Exam Datetime”, “End Exam Datetime”

###############################################

# **CORE FILES**

###############################################

Three main files.

|  |  |
| --- | --- |
| **Python File (.py)** | **Purpose** |
| myGUI.py | User Interface |
| pacs.py | Netcode |
| main.py | Main logic/event loop |

###############################################

# **MODULES**

###############################################

Install the latest version of Python 3

Six modules in use.

Install via the command line tool “pip”. Pip should have

e.g. cmd >>pip install pysimplegui

|  |  |
| --- | --- |
| **Modules** | **Purpose** |
| pysimplegui | User interface |
| pynetdicom | Netcode |
| pydicom | DICOM header modification |
| pandas | Data manipulation |
| xlrd | Load \*.xlsx file |
| openpyxl | Save to \*.xlsx file |

###############################################

# **CONFIGURATION OPTIONS**

###############################################

If absent, config files will automatically be recreated with default values. \*If a prior config file with the correct name exists but is incorrectly formatted, the config file will still be regenerated, and the prior file will be renamed rather than deleted.

##########################

## ANONYMIZATION

##########################

If anonymization is enabled, private tags are dropped completely. As for the standard DICOM header tags, they are filtered via the following config files.

**anon\_imgs.json**

If anonymization is enabled, ignore/drop any images that have these header values. Non-case-sensitive, will match if the string of interest is contained anywhere in the header string. E.g. a value of “Report” will match to “Dose Report”, “Report this man for high treason”

**anon\_vrs.json**

If anonymization is enabled, any headers that are of the listed VR TYPES will be anonymized with extreme prejudice.

**anon\_tags.json**

If anonymization is enabled, anonymize these individual tags . Organized by VR type as well\*\*\*, so that we can skip any tags that are of types already anonymized via “anon\_vrs.json”.

\*\*\*For better usability, need to internalize the VR organization, and instead just have the config file as a list of tagnames

##########################

## DICOM HEADERS / USER

##########################

**excel\_headings.json**

Mapping possible excel headers to the official DICOM tag names. Only partially case-insensitive just due to how this was implemented – will match all-lowercase, all-uppercase, first-letter capitalization. E.g. the excel headers “AccessionNumber”, “Accession”, “Accession Number”, “ACCESSION”, “accession” will all be converted into the official DICOM tag name “AccessionNumber”.

**peers.json**

End user should not need to mess with this. Self-explanatory connection info, in any case. “<SELF>” is ourself, and while our own connection info/port/AET can be changed, our specific name “<SELF>” is used internally by the program and should not be changed

**query\_identifiers.json**

The list of DICOM tags that are used by us to uniquely identify each study/series we are interested in and which will be selectively parsed from the user’s raw input excel file. Only excel columns with headers that correspond to these specific DICOM tag names (specific mappings found in “excel\_headings.json”) will be used, all other columns will be dropped. The user does not need to have all of these columns defined in their initial query input excel file, but be wary that based on what columns are included, each query can and will be as broad or as narrow as you want. E.g. if the only column in your excel file is “MRN”, then searching the database will return every single study associated with each given MRN. With great power comes great responsibility. Make your queries as specific as possible (ideally at the study level, and then you can manually filter which specific series you want using the UI after getting the search results). This is also used to create the table headers in the initial query table (leftmost table). Only excel columns with headers that correspond to these specific DICOM tag names (mapped via “excel\_headings.json”) will be used, all other columns will be dropped

**query.json**

A list of all DICOM header tags that will be requested in each search. Note that all DICOM tags in query\_identifiers.json will also be requested if not already explicitly provided by the user input file. Along with “query\_identifiers.json”, “query.json” is used to create the table headers in the results table (rightmost table).

**tags.json**

End user should not need to mess with this. Master list of all DICOM header tags that we are aware of. Eventually want to add additional functionality to automatically assimilate any new tags we encounter.

**user\_defaults.json**

End user may want to mess with this. Largely naming conventions and directory locations.