Zackary Morelli

[zackmorelli@gmail.com](mailto:zackmorelli@gmail.com)

1/14/2022

ChartQADoc is an Eclipse script that collects Chart QA (which is a specific module of Aria) information for a specific plan or an entire course and puts it into a PDF document, using the Migradoc/PDFSharp library used in my other programs. It does this by making SQL queries of the Aria database, so whoever is running it needs SQL access (which you need to ask Varian for since they are the database administrator). As of recently all the physicists have SQL read-only access.

After making the PDF, the program then uses the Aria Web API to insert the document into Aria’s document module. The document is saved as one of the document types that is set up to flow to EPIC via the HL7 MDM interface.

The idea is that when a physicist has completed the final chart check for a patient who has recently completed treatment, which they do in the Chart QA module of Aria, when they go to open the patient in Eclipse to mark the plan/course as completed, they also run the ChartQADoc script. The script makes a PDF file with all the records from the ChartQA module for the given plan/course, and puts it into the patient’s documents in Aria. The only thing the physicist does after that is open the document to double-check it, and then approve it. On approval, the document automatically is copied to EPIC, such that there is a record of Chart QA tasks that the physicists have performed outside of Aria, which is more accessible to billing people and so on might want to look at it.

So this program is just simply making a document out of information in the Aria database, it doesn’t edit anything. Its purpose it to automate what would otherwise be an onerous documentation task that the physicists would have to perform themselves. Even though all the records are there in Aria, Aria does not create a document for them, which is what the billing department wants.

The program starts with my standard script execute file that launches a WinForm GUI. The user makes their selections and then presses the execute button to trigger an event handler. The execute method of the GUI then runs the rest of the program. All the SQL queries take place in the execute method. After that, the execute method passes lists with the pertinent information to a method stored in a separate class file which creates the PDF document and saves it. It then then calls another method, from another class file, which will perform the REST request to add the document to Aria. There are comments throughout that explain some of the details.