

## Scripting to Improve Planning Consistency, Efficiency and to Streamline the Recommissioning Process

**A**kron General Health System, found within the heart of Ohio, has implemented the RayStation treatment planning system from RaySearch Laboratories, AB. This case study describes how they are using Python Scripting to streamline their workflow while providing continuity during the planning process. The treatment planning system supports the integration of automated scripts that allows users to interact with objects in the patient database and present them or analyze them to develop custom output and reports. Jeremy Donaghue, chief physicist, shares three scripts that have been clinically implemented, having a positive impact on their facility.

### As described by Jeremy Donaghue:

In Radiation Oncology, departments usually define their own standards for reports. Utilizing the scripting features of RayStation, Akron General has been able to streamline these efforts while also reducing the potential for human input errors (typos). This has simplified the process and reduced the amount of time spent by the staff.

After a patient's CT is imported a script is run to generate a treatment plan with appropriate naming conventions, prescription settings (when available), localization beams and contour names (when available). Use of the script helps maintain consistency across the database and planning process. Future development will include the recognition of all of the information in the CT/Sim's description tags, allowing for auto population. The currently implemented script prompts the screen setup window shown below in Figure 1.

A second scripting example demonstrates how efficiency and safety can be improved with post processing. The script is executed once a treatment plan is complete. It is used to sort the order of the treatment fields into



Jeremy Donaghue



Akron General Health System has streamlined their workflow while providing continuity for the planning process.

what has been determined to be the most efficient order for the machine and then names the fields appropriately. Once the treatment order has been established, the expected treatment table coordinates are computed. The plan is then transferred to the record and verify system. The script checks to verify that the data was transferred appropriately and that no corruption or manipulation after transfer has occurred. This is a crucial step for complex IMRT or VMAT plans. Figure 2 shows the output of the script. Prescription data and treatment field parameters are checked and a report is created. This file can be automatically imported into the R&V system. At Akron General, we use Mosaiq from Elekta.

The third and final script assists with software upgrades and recommissioning. Once the system has been commissioned and the acceptance treatment plans have been established, it is possible to programmatically pull data out of all the defined reference plans. When a new version is released, comparing the original "gold standard" data to the calculated

results from the new version of software is an automated process. This allows our facility to rapidly commission the treatment planning software as upgrades occur.

**"Akron General was able to streamline these processes while also avoiding human input errors, making the entire process less intense and time-consuming on the dosimetrist."**

In summary, we think the Python scripting functionality in the RayStation Treatment Planning System provides the following benefits:

- Eliminates time spent on fixing documentation errors that may have little to no significant impact on patient treatment.
- Brings consistency to plans performed by all of the dosimetrists or physicians.
- Allows more time to be invested in evaluating the quality of the treatment plan.
- Helps reduce possible treatment errors.
- Allows us to recommission new software versions quickly.

Case study supplied by RaySearch Laboratories.

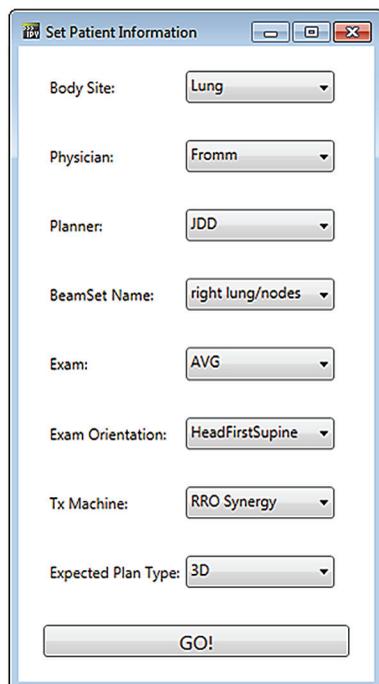


Figure 1: Script-generated window for setting patient plan information.

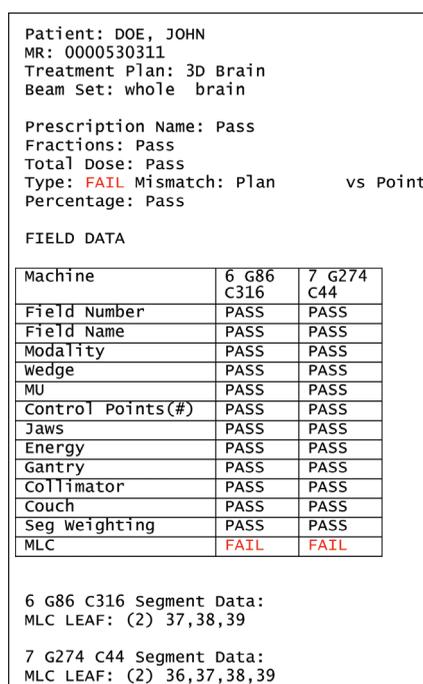


Figure 2: Sample report showing an MLC mismatch and other test output.