



### Introduction to Scripting in RayStation

John Eley  
AAPM 58  
Washington, DC  
4 August 2016

---

---

---

---

---

---

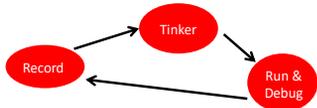
---

---



### How to Get Started

- Read RaySearch documentation (brief)
- Watch video tutorial on scripting on RaySearch website or YouTube
- Use "record" function
- Read up on basic Python code syntax
- Try to modify the recorded scripts and insert your own variables
- Try to use IronPython functions outside of RayStation



---

---

---

---

---

---

---

---

### Try Not to Get Discouraged

- "unscriptable action" => Not always true (Actually this is mostly not true)
- Look for workarounds
- Dig through the "StateTree" / "StateViewer" and try to understand all of the structures as to where data is kept and what functions are there to modify that data
- Try to mirror the syntax used in the recorded scripts, but also, do try to change things and re-run



---

---

---

---

---

---

---

---

### Script Header

- Typically a version of this will auto-generate when recording the script
- But this is also the place to import additional functionality (libraries) to IronPython that you wish to use

```

Bring
RayStation
data into the
memory
(variables)
accessible
by the scrip
# Load necessary objects into memory and import libraries
from connect import *
import math
import string
import json
import time
patient = get_current("Patient")
plan = get_current("Plan")
beam_set = get_current("BeamSet")
    
```

Import extra libraries for specific python functionality

---

---

---

---

---

---

---

---

---

---

### Example 1

Automatically extract a set of dose-volume statistics and display in terminal

- Assumes plan is finished, just wish to see a specific dose statistic, for example *V-60Gy to the brain*
- Useful for internal reporting, clinical trials, faster than clicking around with a mouse on a DVH curve

```

dV60GyBrain = patient \
    .TreatmentDelivery.FractionEvaluations[0] \
    .DoseOnExaminations[0].DoseEvaluations[!LastDose] \
    .GetRelativeVolumeAtDoseValues(RoiName=pcBrainName, DoseValues=[6000])
dV60GyBrain = dV60GyBrain[0] * dBrainVolume # cm3

dBrainVolume = patient.PatientModel.StructureSets['CT 1'].RoiGeometries[pcBrainName].GetRoiVolume() # cm3
    
```

---

---

---

---

---

---

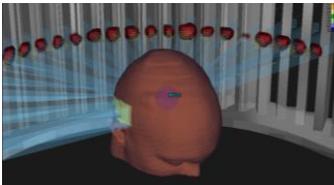
---

---

---

---

### Example 2: Automatically Create Treatment Plans



- Scripted:
- Add proton fields and set gantry angles
  - Specify target and dose prescription
  - Perform robust optimization for each field to ensure adequate margins for proton range uncertainty

---

---

---

---

---

---

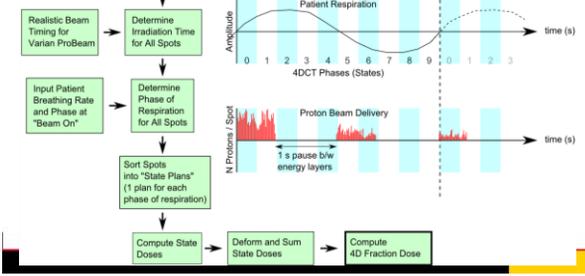
---

---

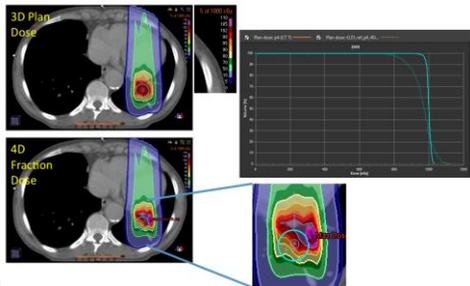
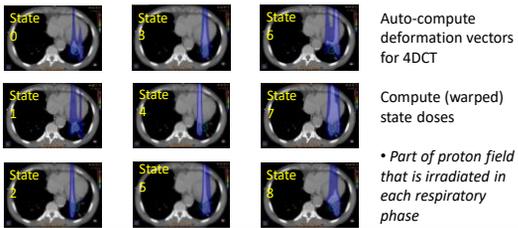
---

---

Example 3: Simulate scanned proton treatment during patient respiration



Example 3: Simulate scanned proton treatment during patient respiration



### Limitations

There are some limits to what can be scripted, for example

- "Plan Approval" status for treatment cannot be automated
- Manual contouring (but auto-segmentation can be scripted)




---

---

---

---

---

---

---

---

### How to Get Better

- Trial/error and explore the StateTree (now called StateViewer, I think)
- Quarterly training courses offered by RaySearch staff
- General practice with Python ... *perhaps those motivated will try to connect IronPython to Python (which supposedly is possible)*

---

---

---

---

---

---

---

---

### Summary

- Don't believe "action is unscriptable", keep trying
  - This just means it cannot be *recorded* by the script recorder
- The StateTree / StateViewer is extremely helpful to find the functions and data structures
  - The official documentation was scarce (at the time of my learning)
- RayStation seems to allow access and modification of nearly all aspects of treatment planning and patient data via the scripting interface
- The IronPython scripting capabilities allow a full, real-time link between the internals of the TPS and the rest of your computing system

---

---

---

---

---

---

---

---