Scripts User Manual

This manual is a guide to setting up and using Kaley's scripts. The first section is dedicated to RayStation scripts, the second to all other scripts. There is both technical/implementation information aimed at scripters, as well as "how-to" usage information for any end user.

Contents

RayStation Scripts

How to Implement a Script in RayStation

See RaySearch's official Scripting Guidelines for comprehensive coverage of how to implement a script in RayStation, but following is CRMC's preferred system.

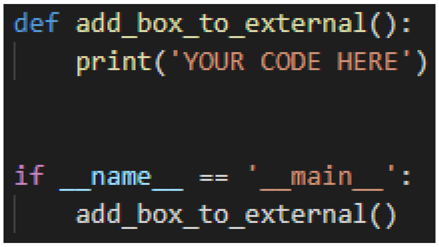
RayStation scripts are stored in `T:\Physics\KW\med-phys-scripts\RayStation`. Each "main" script defines a function with the same name as the file and calls this function inside `if __name__ == '__main__'`. The "blurb" scripts are in the `Blurbs` folder. Each "blurb" script has the same name as the "main" script prepended with an underscore, and simply runs the "main" script using RayStation's `connect.run` function. The "blurb" script is imported into RayStation. Separating the files in this way keeps us from having to invalidate, edit in RayStation or reimport from file, revalidate, and save the script every time we change the code.

We always use our custom RayStation scripting environment `CPython 3.8`.

Example

Here is how we set up `add_box_to_external` in RayStation:

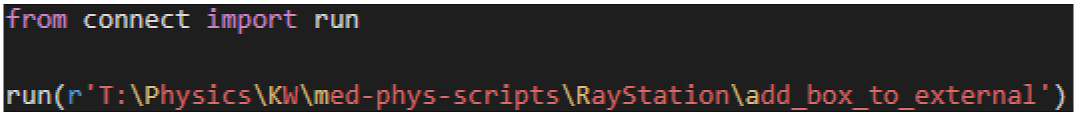
1. Write `add_box_to_external.py` and save it in the `RayStation` folder.



```
def add_box_to_external():
    print('YOUR CODE HERE')


if __name__ == '__main__':
    add_box_to_external()
```

2. Write `_add_box_to_external.py` and save it in the `Blurbs` folder.



```
from connect import run

run(r'T:\Physics\KW\med-phys-scripts\RayStation\add_box_to_external')
```

3. Import the blurb into RayStation and name the script `add_box_to_external`.

`add_box_to_external`

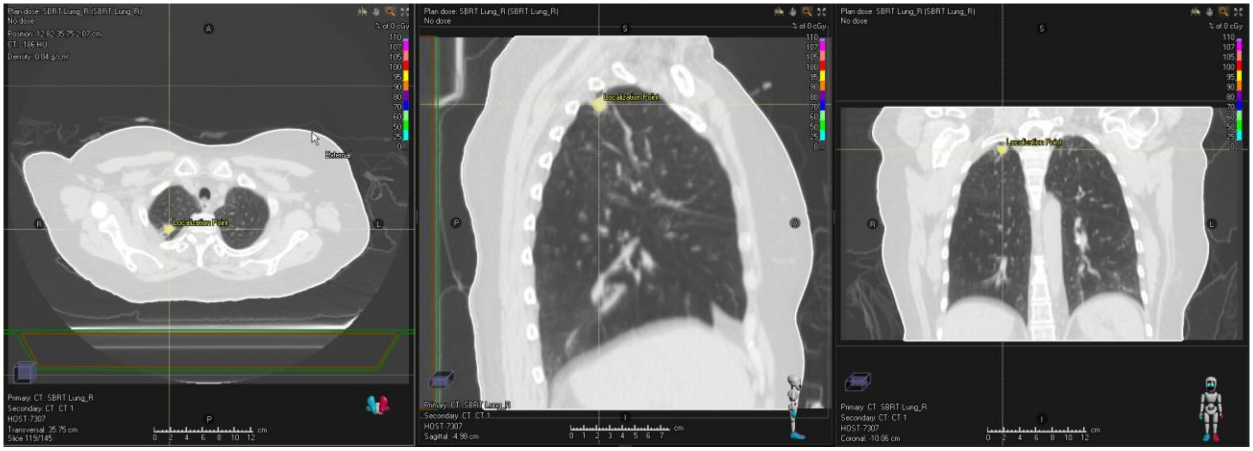
Modifies the external (body) geometry on the current exam, to include some of the Vac-Lok™ geometry used for SBRT patients.

SBRT lung patients are positioned using a Vac-Lok™ bag. When planning, we prefer our external geometry to include the body of the patient as well as some area posterior and right/left of the patient. This area ideally encompasses the entire couch geometry in the I-S and R-L directions, and up to the localization point in the P-A direction. In practice, we crop the area by a pixel on each R-L side to prevent possible calculation errors due to the external extending outside the image.

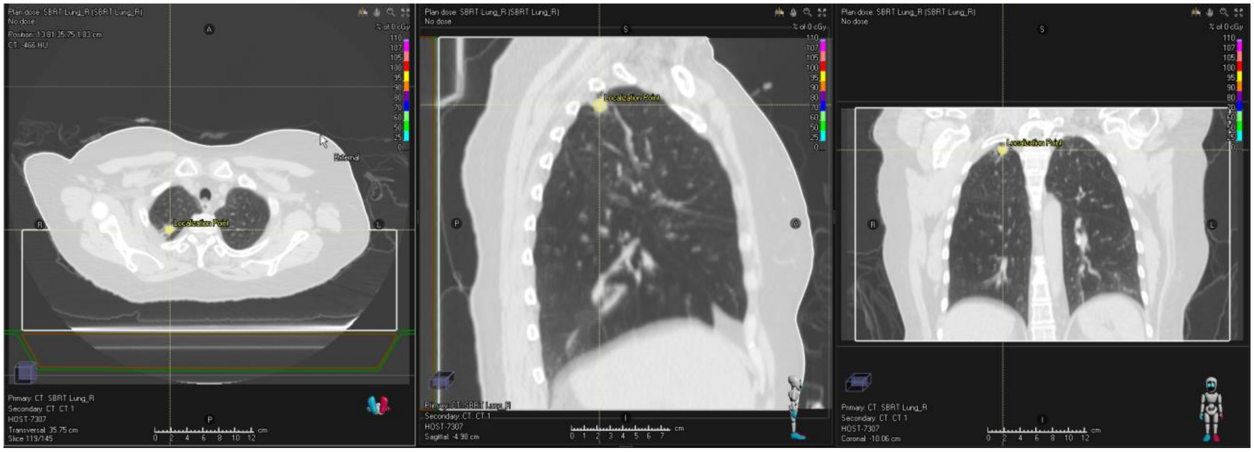
Of course, we want to retain the original external geometry, so before adding the box to it, we copy the external geometry into a new ROI called `External^NoBox`.

Example

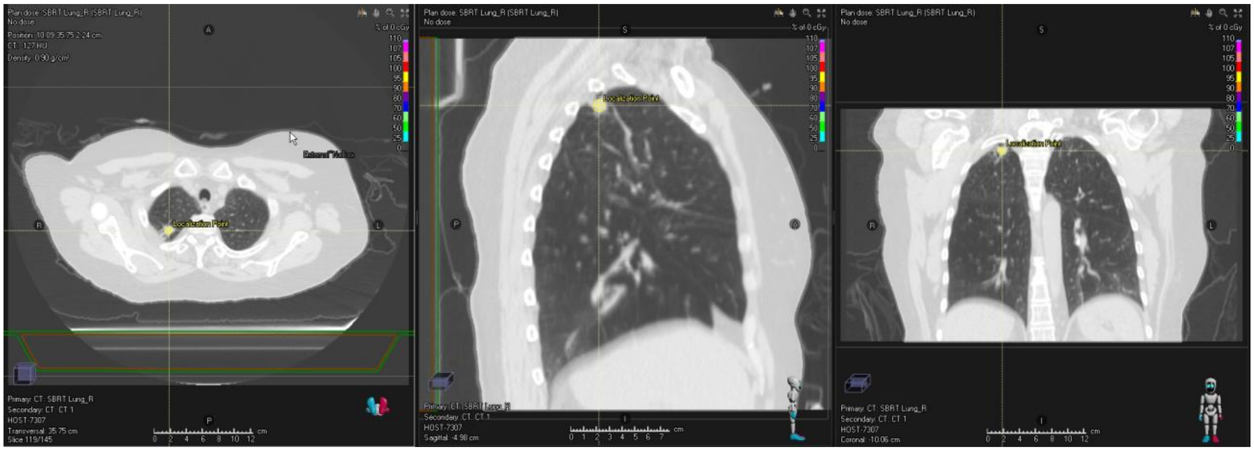
Here is what the external looks like before running the script; it's just the body:



After running the script, we have the modified external:



And `External^NoBox`, containing the original external geometry:



copy_plan_without_changes

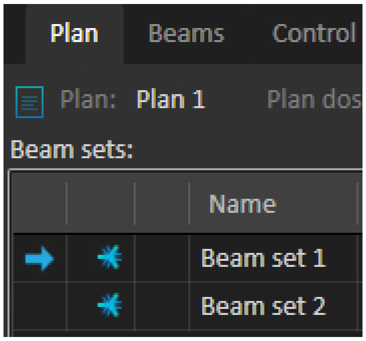
Copies the current plan, retaining beam set and isocenter names.

RayStation's **Copy plan** functionality is amazing, but we wanted to change a couple of naming conventions in the copy:

- Retain beam set and isocenter names.
- The beam sets and beam isocenters in the copied plan have the same name as the plan, plus a copy number. For example, if a plan called `R Breast` with two beam sets is copied to a plan called `R Breast (1)`, the new beam sets are called `R Breast (1)` and `R Breast (1)_2` respectively, regardless of their names in `R Breast`. Likewise, the beam isocenters are called `R Breast (1) 1` and `R Breast (1)`.
- Renumber the new beams.
  We make the new beam numbers unique among all beam numbers in the patient. The first beam is numbered one more than the greatest beam number in the patient, and beam numbers increase consecutively from there.
- Name beams the same as their numbers.
  We want to retain any important information from the old name, though, so if an old beam name is different from its number (or a setup beam name is not the default), we append the old name to the new description.

Example

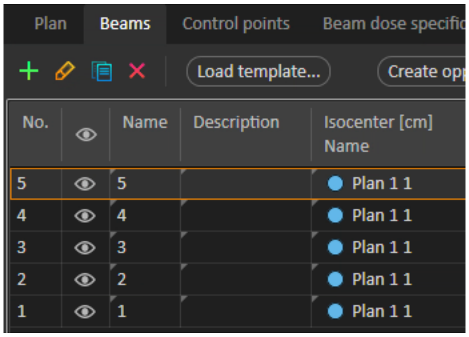
Given a plan `Plan 1`:



Plan: Plan 1

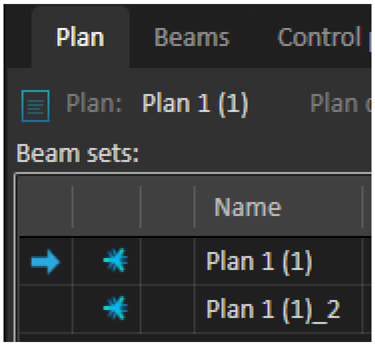
Beam sets:

Name
Beam set 1
Beam set 2



No.	Name	Description	Isocenter [cm] Name
5	5		Plan 1 1
4	4		Plan 1 1
3	3		Plan 1 1
2	2		Plan 1 1
1	1		Plan 1 1

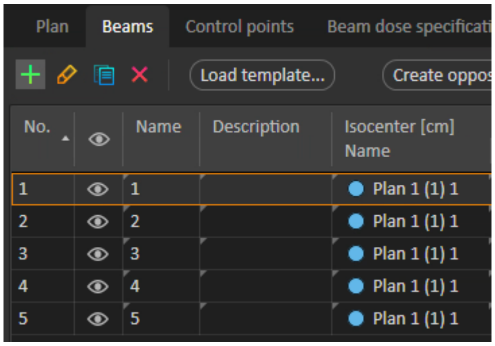
If we copy `Plan 1` using the **Copy plan** button, naming the copy `Plan 1 (1)`, we get these names and numbers:



Plan: Plan 1 (1)

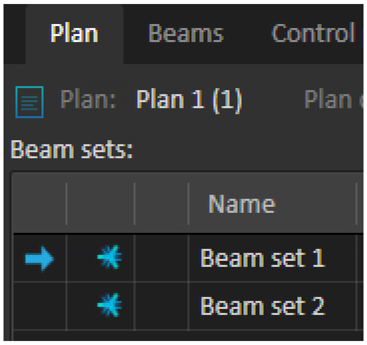
Beam sets:

		Name
➡	✳	Plan 1 (1)
	✳	Plan 1 (1)_2



No.		Name	Description	Isocenter [cm] Name
1	👁	1		● Plan 1 (1) 1
2	👁	2		● Plan 1 (1) 1
3	👁	3		● Plan 1 (1) 1
4	👁	4		● Plan 1 (1) 1
5	👁	5		● Plan 1 (1) 1

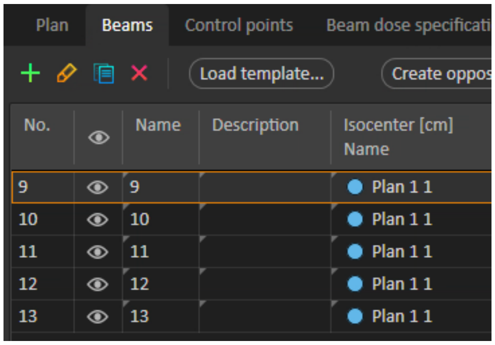
If we instead copy the plan using the script, the `Plan 1 (1)` beam sets and beams match those in `Plan 1` and are uniquely numbered:



Plan: Plan 1 (1)

Beam sets:

		Name
➡	✳	Beam set 1
	✳	Beam set 2



No.		Name	Description	Isocenter [cm] Name
9	👁	9		● Plan 1 1
10	👁	10		● Plan 1 1
11	👁	11		● Plan 1 1
12	👁	12		● Plan 1 1
13	👁	13		● Plan 1 1

Other Scripts

`ChromeDefaultBrowser`

We recently had an issue with our default browser. Every day when we logged in, we set Chrome as the default browser. But the next morning, the default browser was reset to Edge. It turned out that Group Policy had started resetting everyone's default browser to Edge upon logout. Therefore, in order to use a different default browser, we must manually set it every time we log in. Here is a way to automatically set Chrome as your default browser every time you log in. Like all good IT workarounds, it doesn't require admin permissions.

I got this solution from Stack Exchange, Mark McClelland, and Christoph Kolbicz.

Dependencies

- The browser you want to set as the default. To use the solution as is, this is Chrome.
- Christoph Kolbicz's `SetDefaultBrowser.exe`
- This solution was tested in Windows 10, so it is not guaranteed to work with any other operating system.

Implementation

The batch file `ChromeDefaultBrowser.bat` runs `SetDefaultBrowser.exe` (using the absolute path) with the `chrome` argument:

```
START T:\Physics\KW\med-phys-scripts\ChromeDefaultBrowser\SetDefaultBrowser.exe chrome
```

The file `ChromeDefaultBrowser.xml` is an exported task from Task Scheduler. Import it into Task Scheduler. The task runs the batch file whenever:

- Your computer is unlocked
- You log in
- You connect to a user session

The following are user-/computer-specific and should be changed:

- The path to `SetDefaultBrowser.exe` in the batch file. I recommend using the absolute path.
- The path to the batch file in the XML file. The path is inside the `<Command>` tags.
- Your username in the XML file. This is between the `<Author>` and the `<UserId>` tags, where the placeholder is `DOMAIN\USER`.