

# rtp-mangle

---

A Python DICOM-RT Plan "Mangler" for Radiotherapy Quality Assurance

Ever wanted to change your radiotherapy treatment plan in weird and wonderful ways? Well - now you can. With this tool, which is essentially a wrapper to pydicom, you can modify delivery properties to intentionally deliver incorrect treatment plans. This allows you to test the sensitivity of your quality assurance methods, such as diode arrays or in-vivo dosimetry systems.

## Installation

---

To run rtp-mangle, first download or clone this repository. To obtain the minimal prerequisite packages, run "pip install pip install -r requirements.txt"

## Usage

---

### Basic Operation

---

The script can be run with the following command - square brackets indicate optional parameters:

```
python mangle.py [options] "input.dcm" "<Command String>" ["<Command String>"
...]
```

A command string specifies how to edit the RT Plan file - see below for more details.

### Options

---

Additional options include displaying the help text (-h), specifying the output file name (-o "output.dcm"), verbose mode for command string debugging (-v), and keep SOPInstanceUID mode (-k).

The keep SOPInstanceUID mode is important for testing - some devices you might be testing will require this to be identical to the original planned treatment in order to allow analysis to be performed. Other systems will refuse to import files with a duplicate UID. The default behaviour of rtp-mangle is to create a new SOPInstanceUID.

# Command Strings

To make edits to the plan, we use a command string. Command strings comprise of two parts - filters and setters. Filters are used to specify which parts of the plan should be changed. Setters are used to make a change. All available filters and setters are listed in the table below.

The simplest command string would have no filters, and include a single setter. For example, "g=0" will set all control points for all beams to deliver at gantry angle 0. Whenever you don't specify a filter, it assumes you want to edit all available items.

Setters can be given an absolute value like the previous example, or they can perform a relative modification. To increase the gantry angle in every beam and control point by +5 degrees we'd use the command string "g=+5" - or if we wanted to decrease it by -5% we'd use "g=-5%". Note that currently there is no wrap around 360 degrees, so negative values or values >360 degrees are possible - this will probably be rejected when attempting to deliver the plan.

To set a beam limiting device position absolutely or relatively, we need to be more specific than this because negative values are allowed and =-10 is ambiguous. Therefore we have pr (position relative) and pa (position absolute) setters.

We can introduce filters to restrict the edits to only the first beam in the file. The string "b0 g=0" will set all control points for only the first beam to deliver at gantry angle 0. Note that filters don't use an equals sign - but setters do. Also, remember that DICOM uses a zero indexing system, so the first beam is beam 0.

We can specify multiple filters to restrict the edits even further - if we wanted to change the gantry angle only in the first control point of the second beam, we'd use a command string of "b1 cp0 g=0".

Filters also allow you to specify more than one number or even a range. If we wanted to change the collimator angle of beams 1, 2 and 3 we can achieve this with either "b0,1,2 c=0" or "b0-3 c=0".

Any more complex filtering might require using two command strings one after the other - in a plan with 6 beams, we may wish to leave beam 5 alone. "b0-3 c=0" "b5 c=0".

## Available Filters

Filter	Name	Description
b	Beam	DICOM Beam number. Starts at 0.

Filter	Name	Description
cp	Control Point	Control Points in Beam Sequence. Starts at 0.
j	Jaw	X or Y jaw - use 0 or 1. Experiment to determine which is which.
jb	Jaw Bank	X1/Y1 or X2/Y2. Use 0 or 1, experiment to identify.
lp	Leaf Pair	The leaf pair number. Starts at 0.
lb	Leaf Bank	MLC Bank A or B. Use 0 or 1, experiment to identify.

## Available Setters

Setter	Name	Description
mu=	MU	Change the prescribed monitor units.
m=	Machine	Change the treatment machine name. Use single quotes to include a space.
g=	Gantry	Gantry Angle
c=	Collimator	Collimator Angle.
pa=	Position Absolute	Change the absolute position of the BLD. Requires a jaw or MLC filter to work.
pr=	Position Relative	Change the relative position of the BLD. Requires a jaw or MLC filter to work.

## Rules

- Never, EVER, use this on a clinical treatment plan. This is a QA tool only.
- You can't edit the Jaw and MLC positions within the same command string, because they both require the pa= and pr= setters. Do one edit and then the next in two separate command strings.
- You can't use both the pa= and pr= setters within the same command string. It doesn't make any sense! If you want to shift the absolute position, then perform a relative movement, use two separate command strings.

## Examples

## Changing the MU

---

Our plan has three beams. We need to set the MU of the first beam to 100MU exactly. We need the second beam to be increased by 10MU, and the third beam's MU to be increased by 15%.

```
python mangle.py "input.dcm" "b0 mu=100" "b1 mu=+10" "b2 mu=+15%"
```

## Changing the Treatment Machine Name

---

We need to deliver this plan on "Linac 2" instead. Note that in some oncology management systems this must be an exact value before being imported.

```
python mangle.py "input.dcm" "m='Linac 2'"
```

## Changing the Jaw Positions

---

For the first beam, between control points 12 and 16, we need to move the X jaw (which in our case happens to be j0) on the X2 side (which in our case happens to be jb1) to an absolute position of -5.2.

```
python mangle.py "input.dcm" "b0 cp12-16 j0 jb1 pa=-5.2"
```

## Changing the MLC Positions

---

Open all of the Y1 side MLCs an extra 10mm in all control points of the second beam.

```
python mangle.py "input.dcm" "b1 lb0 pr=+10"
```

## Simulate a Stuck Leaf

---

Set leaf pair 7 on the Y2 side to an absolute value of -400 for all beams and control points.

```
python mangle.py "input.dcm" "lb1 lp6 pa=-400"
```

## Specify the output File

---

Set the gantry to 0 in all beams and control points, then save the edits to the filename specified.

```
python mangle.py "input.dcm" -o "output.dcm" "g=0"
```