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 4](#_Toc101533837)

[How to Implement a Script in RayStation 4](#_Toc101533838)

[Example 4](#_Toc101533839)

[General End User Info 4](#_Toc101533840)

[Scripts You Should Not Use 4](#_Toc101533841)

[Script Validation Message 4](#_Toc101533842)

[Assumptions 4](#_Toc101533843)

[Warnings 5](#_Toc101533844)

[Add Box to External 5](#_Toc101533845)

[Assumptions 5](#_Toc101533846)

[Example 6](#_Toc101533847)

[Coming Soon 6](#_Toc101533848)

[Add Clinical Goals 6](#_Toc101533849)

[Assumptions 7](#_Toc101533850)

[Template Names 7](#_Toc101533851)

[Clinical Goal Format 7](#_Toc101533852)

[ROI Names 7](#_Toc101533853)

[Notes Column 7](#_Toc101533854)

[Invalid Goals 8](#_Toc101533855)

[Approved Plans 8](#_Toc101533856)

[Other Goals 8](#_Toc101533857)

[Visualization Priority 8](#_Toc101533858)

[Example 8](#_Toc101533859)

[Coming Soon 10](#_Toc101533860)

[Add Couch 11](#_Toc101533861)

[Adds template couch structures at the correct location on the current exam. 11](#_Toc101533862)

[Sim Couch Location 11](#_Toc101533863)

[Assumptions 11](#_Toc101533864)

[Dose Grid Update 11](#_Toc101533865)

[Dose Re-computation 11](#_Toc101533866)

[Example 11](#_Toc101533867)

[Add Derived ROIs 12](#_Toc101533868)

[Clean Up Geometries 12](#_Toc101533869)

[Contour Chestwall 12](#_Toc101533870)

[Assumptions 14](#_Toc101533871)

[Warnings 14](#_Toc101533872)

[Example 14](#_Toc101533873)

[Coming Soon 15](#_Toc101533874)

[Compare Contours 15](#_Toc101533875)

[Assumptions 15](#_Toc101533876)

[Example 15](#_Toc101533877)

[Convert Virtual Jaw to MLC 17](#_Toc101533878)

[Assumptions 17](#_Toc101533879)

[Example 18](#_Toc101533880)

[Copy Beam Set 19](#_Toc101533881)

[Assumptions 19](#_Toc101533882)

[Approved Plans 20](#_Toc101533883)

[Naming and Numbering Conventions 20](#_Toc101533884)

[Example 20](#_Toc101533885)

[Copy Opt Stuff 20](#_Toc101533886)

[Assumptions 20](#_Toc101533887)

[Approved Plans 20](#_Toc101533888)

[Example 20](#_Toc101533889)

[Coming Soon 21](#_Toc101533890)

[Copy Plan to New CT or Merge Beam Sets 21](#_Toc101533891)

[Not a “Real” Copy 22](#_Toc101533892)

[How It Works 22](#_Toc101533893)

[Assumptions 23](#_Toc101533894)

[Warnings 23](#_Toc101533895)

[Coming Soon 23](#_Toc101533896)

[Example 23](#_Toc101533897)

[Copy Plan without Changes 24](#_Toc101533898)

[Example 24](#_Toc101533899)

[Copy Beam Set 25](#_Toc101533900)

[Copy Exam 25](#_Toc101533901)

[Create QA Plan 25](#_Toc101533902)

[Delete Empty ROIs 25](#_Toc101533903)

[Dose Grid Box 25](#_Toc101533904)

[Example 26](#_Toc101533905)

[Exclude from MOSAIQ Export 26](#_Toc101533906)

[Export DVH Curves 26](#_Toc101533907)

[Export Line Doses 26](#_Toc101533908)

[Extend Exam 26](#_Toc101533909)

[Generate Shifts Comments 26](#_Toc101533910)

[Last Modified Times 26](#_Toc101533911)

[List Patients 26](#_Toc101533912)

[Localize to Beam Iso 26](#_Toc101533913)

[Parse Logfiles 27](#_Toc101533914)

[Logfile Types 27](#_Toc101533915)

[Filepaths 27](#_Toc101533916)

[Spreadsheet Name 27](#_Toc101533917)

[Data Dictionary 27](#_Toc101533918)

[Example Events 28](#_Toc101533919)

[Example 29](#_Toc101533920)

[Plan Check 29](#_Toc101533921)

[Plan Quality Metrics 30](#_Toc101533922)

[Prepare Exams 30](#_Toc101533923)

[Print Report 30](#_Toc101533924)

[QACT Adaptive Analysis 30](#_Toc101533925)

[TG-263 Names & CRMC Colors 30](#_Toc101533926)

[Scale Objectives & Constraints 30](#_Toc101533927)

[Show Non-Empty Geometries 30](#_Toc101533928)

[Update Clinical Goals Templates 30](#_Toc101533929)

[Update Objectives & Constraints Templates 30](#_Toc101533930)

[Other Scripts 30](#_Toc101533931)

[ChromeDefaultBrowser 30](#_Toc101533932)

[Dependencies 30](#_Toc101533933)

[Implementation 30](#_Toc101533934)

[Appendix: Common RayStation Script Error Messages 31](#_Toc101533935)

[Incorrect Modality 31](#_Toc101533936)

# RayStation Scripts

## How to Implement a Script in RayStation

See RaySearch's official [Scripting Guidelines](file:///\\D117raysql\rsconfig$\Manuals\11ASP1\RSL-D-RS-11A-SG-EN-1.0-2021-05-21%20RayStation%2011A%20Scripting%20Guidelines.pdf) for comprehensive coverage of how to implement a script in RayStation, but following is CRMC’s preferred system.

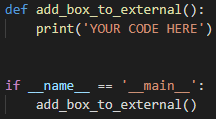
RayStation scripts are stored in [T:\Physics\KW\med-phys-scripts\RayStation](file:///T:\Physics\KW\med-phys-scripts\RayStation). Each “main” script defines a function with the same name as the file. The “blurb” scripts are in the [Blurbs](file:///T:\Physics\KW\med-phys-scripts\RayStation\Blurbs) folder. Each “blurb” script has the same name as the “main” script prepended with an underscore, and simply imports and calls the function from the “main” script. 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, and we give the scripts in RayStation, more human-friendly names.

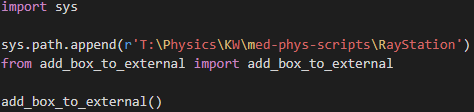
### Example

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](file:///T:\Physics\KW\med-phys-scripts\RayStation) folder.



1. Write \_add\_box\_to\_external.py and save it in the [Blurbs](file:///T:\Physics\KW\med-phys-scripts\RayStation\Blurbs) folder.



1. Import the blurb into RayStation and name the script Add Box to External.

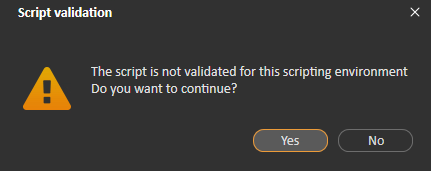
## General End User Info

### Scripts You Should Not Use

A script name starts with a single *z* if it is really only useful for scripters. It starts with *zz* if it is currently being tested and is not ready for use.

### Script Validation Message

If the following message appears when you run a script in RayStation, you may ignore it (click **Yes**).



### Assumptions

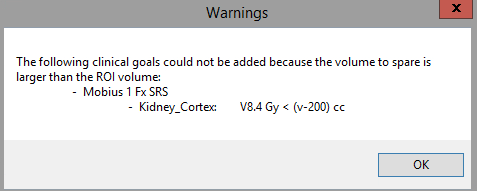
If any of the script’s assumptions are violated, you are prompted to end the script or take some other action. These are not error messages! Here is an example popup from [Add Clinical Goals](#_Add_Clinical_Goals).



Each script section lists the conditions under which a popup message will appear for violated assumptions.

### Warnings

At the end of the script, a popup may appear with warnings about what the script could not do. Here is an example warning from [Add Clinical Goals](#_Add_Clinical_Goals). The script was able to finish but could not do everything we wanted it to.



Each script section lists the conditions under which a popup message will appear for warnings.

## 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.

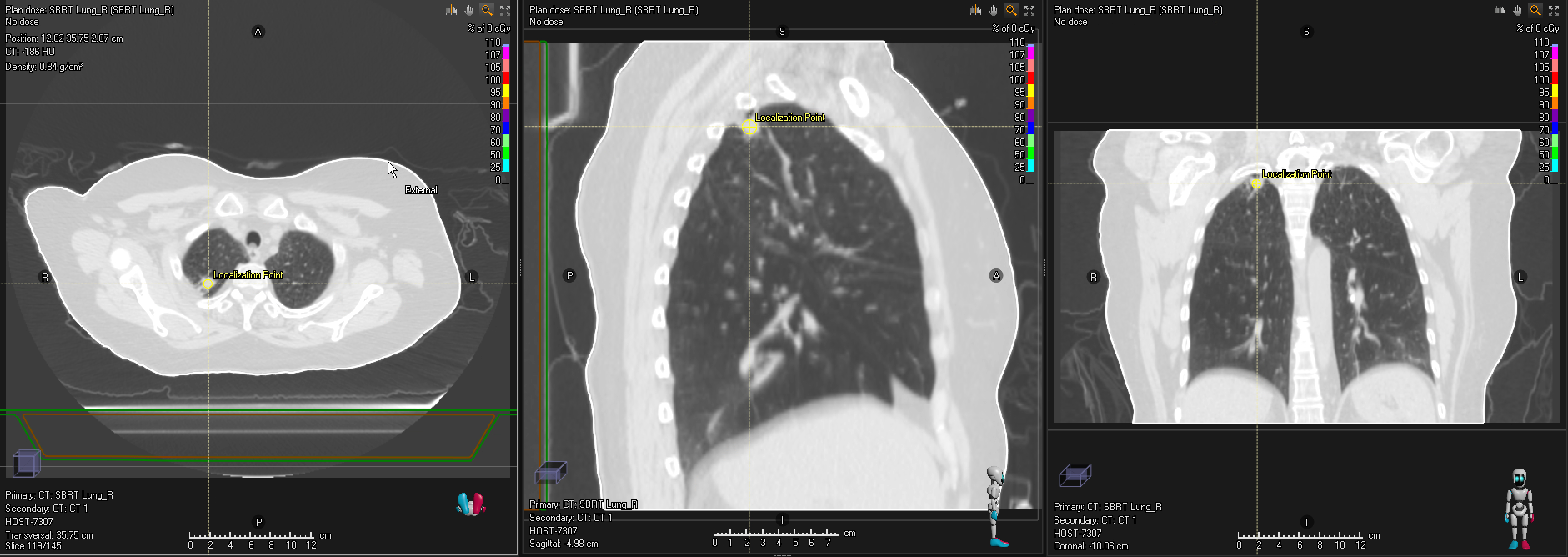
Of course, we want to retain the original external geometry, so before adding the box to it, we copy the external geometry into a new ROI called External^NoBox.

### Assumptions

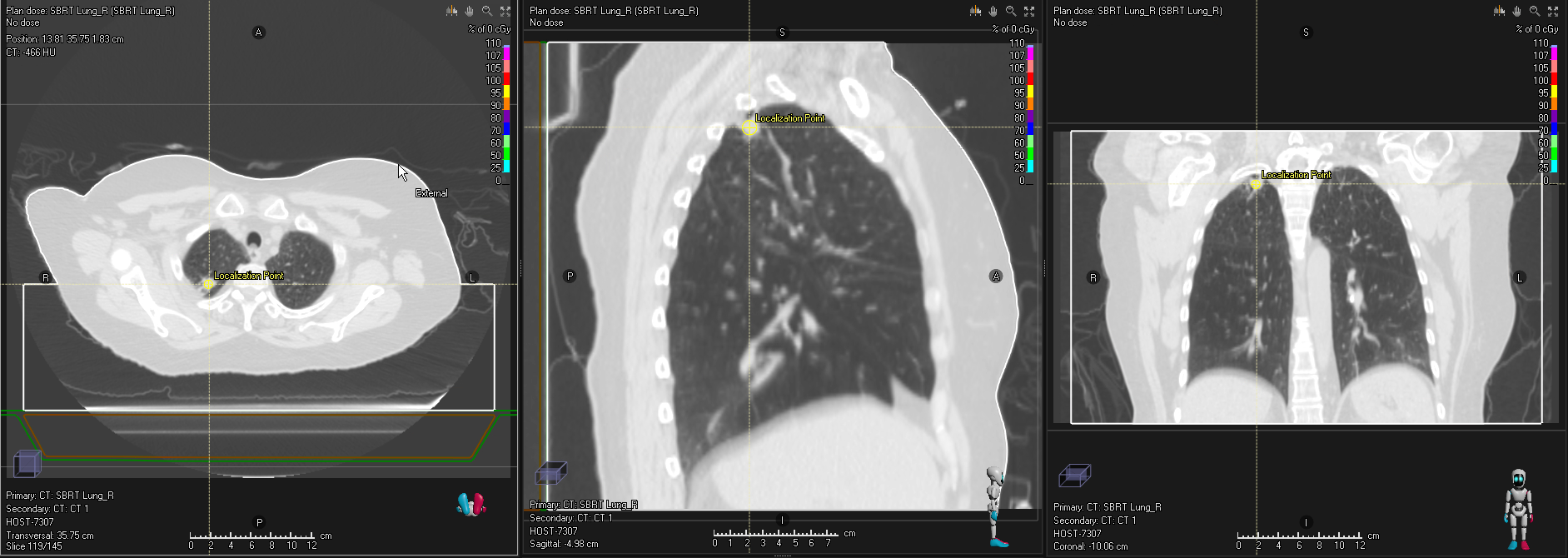
* A case is open.
* The open case has at least one exam.
* The current exam has an external geometry.
* The external geometry on the current exam is unapproved.
* The couch ROIs exist in the current case.
* The current exam has both couch geometries.
* The current exam has a localization geometry.
* The box geometry (Box), if it exists, is unapproved on the current exam.
* The external geometry without the box (External^NoBox), if it exists, is unapproved on the current exam.

### Example

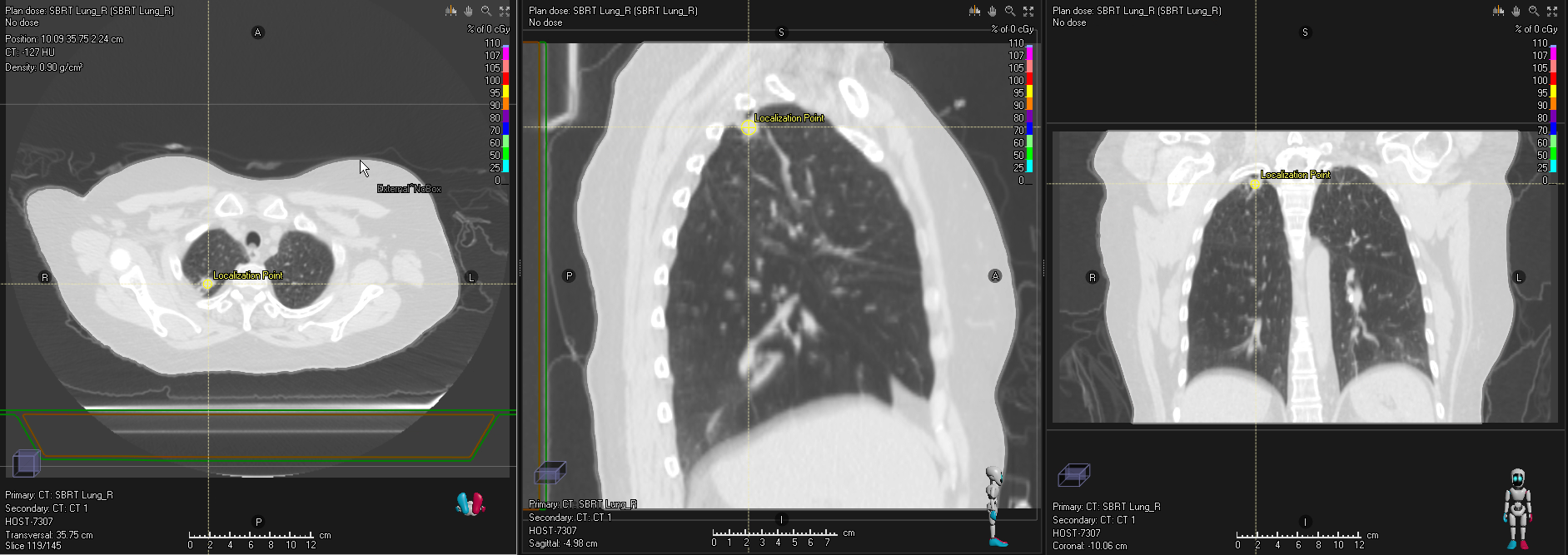
Here is what the external looks like before running the script; it's just the body:



After running the script, we have the modified external:



And External^NoBox, containing the original external geometry:



### Coming Soon

* Add the couch if it doesn’t exist.
* Add the external geometry if it doesn’t exist.
* Add new Box and External^NoBox geometries if the existing ones are approved.

## Add Clinical Goals

Adds clinical goals templates from a spreadsheet, to the open plan.

RayStation has clinical goals templates, but they are cumbersome to update. It is easier to keep an updated spreadsheet of clinical goals for various sources/MDs, body sites, treatment techniques, Rx’s, fractionation patterns, etc. We keep this spreadsheet in [T:\Physics\KW\med-phys-spreadsheets](file:///T:\Physics\KW\med-phys-spreadsheets). The priority columns in the spreadsheet are ignored.

### Assumptions

* A patient is open.
* A case is open.
* A plan is open.
* The open plan has at least one beam set.
* The current beam set’s modality is photons.
* The current beam set’s machine is commissioned.

### Template Names

Each sheet in the spreadsheet is a clinical goals template. The sheet name is in the following format. Each part is optional.

1. Source (e.g., Dr. J, Chao, or RTOG 0813)
2. Fractionation description (e.g., Conventional) or number (e.g., 1 Fx, 8 or 10 Fx, or 5-6 Fx)
3. Tx technique (e.g., IMRT)
4. Body site(s) (e.g., Pros or Rectum Anus)
5. Rx(s) (e.g., 60 Gy or 70 74 Gy)

Example: Dr. J 5 Fx SRS Brain 30 Gy

### Clinical Goal Format

Goals in the spreadsheet are written in the following formats. Spacing is ignored.

|  |  |  |  |
| --- | --- | --- | --- |
| **Dose goal** | | | |
| D | \_\_%, \_\_ cc, (v-\_\_ cc), max, min, mean, median | > or < | \_\_% Rx, Rx, Rxp, Rxn, or\_\_ cGy |
| **Volume goal** | | | |
| V | \_\_% Rx, Rx, Rxp, Rxn, or\_\_ cGy | > or < | \_\_%, \_\_ cc, or (v-\_\_ cc) |

### ROI Names

ROI names in the spreadsheet are TG-263 compliant. ROI names in RayStation are matched if:

* They match exactly
* They match with an \_L or \_R suffix
* They match with some “extra” information: ^ plus the info
* They match with a copy number in parentheses
* The ROI in the spreadsheet is “general,” and the ROI in RayStation is in the list of “specific” ROIs listed at the top of the script for that “general” ROI, accounting for the above rules

For example, a goal for Lung is applied to Lung, Lung\_L, Lung^Adapt, Lung (2), etc.

TG-263 PRV names do *not* match a spreadsheet ROI name that is not a PRV. A PRV ROI in the spreadsheet that does not include an expansion matches any PRV ROI in RayStation. For example, a goal for Lens\_PRV is applied to Lens\_PRV, Lens\_PRV03, etc.

### Notes Column

The Notes column in the spreadsheet can be anything, but the script uses the following types of notes (case insensitive):

* \_\_ Gy. Only apply the goal if the Rx matches.
* \_\_ fx. Only apply if the Rx fractionation matches.
* Ipsilateral or contralateral (case insensitive): For goals to paired organs, the side of the organ that the goal is for. The left organ is to the left of the Rx side, the right organ to the right. The “center” of the Rx is determined by:
  + If the Rx is to a target, the center of the target geometry
  + If the Rx is to a DSP, the DSP coordinates
  + If the Rx is to a site that is not a DSP, the coordinate of max dose
  + Otherwise, the coordinates of the initial laser isocenter if it exists

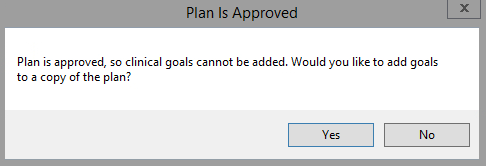
### Invalid Goals

The following goals cannot be added. You are warned of any of these goals before the script exits

* Goals that do not match the above formats
* Volume-to-spare goals for which geometries are empty
* Volume-to-spare goals for which the volume to spare is larger than the geometry volume
* Ipsilateral/contralateral goals for which the Rx side could not be determined
* Goals for a nodal PTV, but there is no nodal PTV

### Approved Plans

If the plan is approved, you may add goals to a copy of the plan.



See [Copy Plan without Changes](#_Copy_Plan_without) for details on how the copy is created.

### Other Goals

In addition to the template goals, the script adds the following goals:

* If there is a prescription and the exam has an external geometry, add a Dmax (D0.03 cc) goal to the external. For SABR plans (including SBRT, SRT, and SRS), this is 125 percent of the primary prescription. For non-SABR plans, it is 110 percent.
* For each prescription to a PTV, add D95% ≥ Rx, V95% Rx ≥ 100%, D100% ≥ 95%, and VRx ≥ 95% goals to the PTV. If the PTV is derived, add D100% ≥ Rx and VRx ≥ 100% goals to each CTV in the PTV’s derived expression.

### Visualization Priority

The **Priority** column in the clinical goals display in RayStation tells which template a goal comes from. The templates are applied in alphabetical order. If a goal is in multiple templates, the priority is the first template that includes it. A key for these numbers is added to the plan comments. Goals that are not part of a template (see [Other Goals](#_Other_Goals) above) have no visualization priority. Existing goals’ priorities are unchanged.

### Example

Here, we add two clinical goals templates to an SBRT lung plan: RTOG 0813’s and Mobius3D’s.

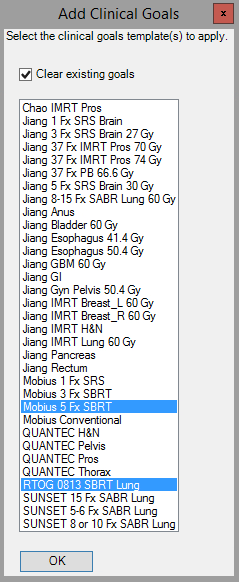
1. The script launches with the following window:
   1. By default, the existing goals in a plan are deleted and replaced with the ones from the selected templates.
   2. The script selects the appropriate Mobius3D script based on the plan’s treatment technique and fractionation.



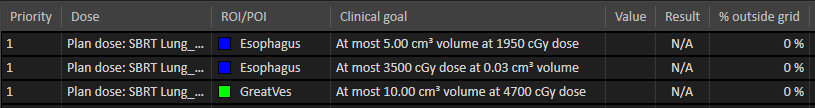
2

1

1. Select the templates that you want to apply. These may include the default selection but don’t have to. You can select multiple templates using Ctrl+Click and Shift+Click.



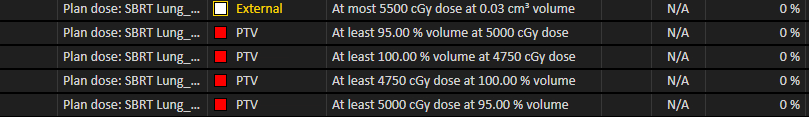
1. The goals for ROIs that exist in the current case are applied:



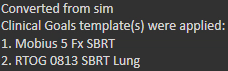
⋮



⋮



The plan comments:



### Coming Soon

* Disable and uncheck the **Clear existing goals** checkbox if there are no goals in the current plan.
* Make the visualization priorities unique.
* Add Jiang 3-5 Fx SRS Brain and others.
* Add tab for objectives/constraints templates.

## Add Couch

Adds template couch structures at the correct location on the current exam.

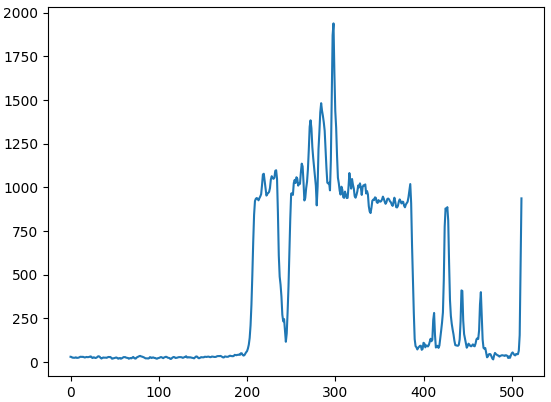
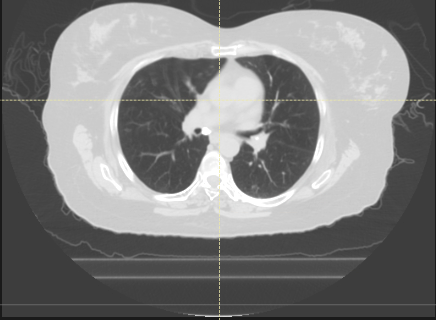
The Elekta Couch template is applied for supine patients, the Elekta Prone Couch template for prone. The correct location is the R-L center and I-S center and at the top of the sim couch in the P-A direction.

### Assumptions

* A patient is open.
* A case is open.
* The current case has at least one exam.
* The patient position of the current exam is HFS, HFP, or FFS.
* Neither couch geometry, if it exists, is approved on the current exam.

### Sim Couch Location

The sim couch location is determined using the exam’s pixel intensities. The below graph shows the y-axis intensities of an example CT. We select all peaks meeting certain height and prominence criteria (determined via trial and error). The last (most posterior) of these corresponds to the bottom of the sim couch.

2.5 cm

The coordinate of the peak is then converted to an exam coordinate and moved 2.25 cm anteriorly to the top of the sim couch instead of the bottom. The couch structures are then added with centers at the P-A center of the sim couch.

### Dose Grid Update

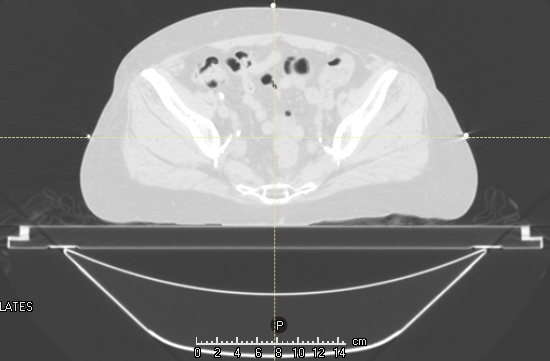
The default dose grid encompasses all geometries. The script sets the default dose grid for any unapproved plans on the current exam that are not initial sim plans. (An initial sim plan is named Trial 1, Trial\_1, Initial Sim, etc. [case insensitive]).

### Dose Re-computation

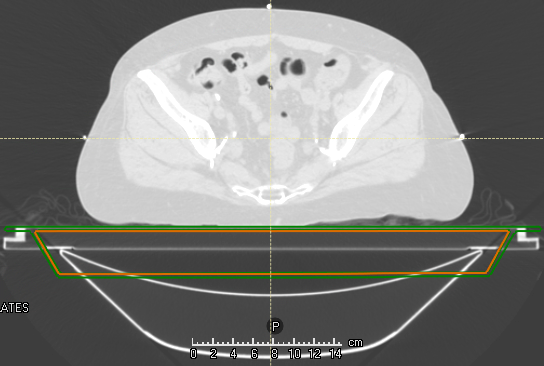
Adding the couch invalidates any computed dose, so the script re-computes the dose on any unapproved beam sets.

### Example

Before running script, there are no couch geometries, but the script would also work if the geometries existed but were unapproved.



After running script, the couch geometries are in the correct location.



## Add Derived ROIs

[Coming soon]

Adds derived ROIs from the [TG-263 Nomenclature with CRMC Colors](../med-phys-spreadsheets/TG-263%20Nomenclature%20with%20CRMC%20Colors.xlsm) spreadsheet, to the current case, if all source ROIs exist.

## Clean Up Geometries

[Coming soon]

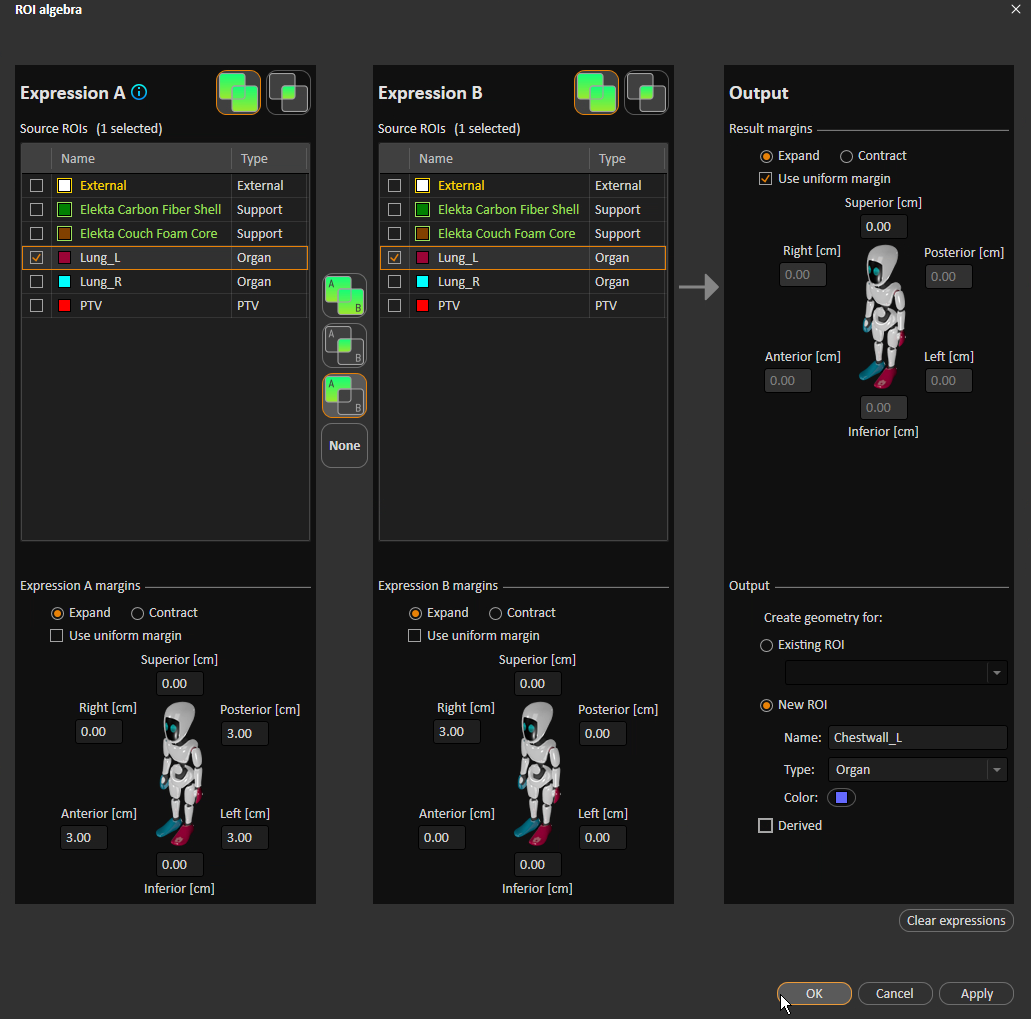
For the selected ROI(s) and exam(s), removes overlap; smooths via expansion then contraction; adds PRV(s); crops to external, FOV, exam, and/or dose grid; and/or centers in R-L, I-S, and/or A-P directions.

## Contour Chestwall

Contours Chestwall\_L and Chestwall\_R geometries using ROI algebra of lung geometries.

The chestwall is the difference of the anteriorly, posteriorly, and ipsilaterally expanded ipsilateral lung, and the ipsilaterally expanded contralateral lung, using the expansion LUNG\_EXPANSION\_FOR\_CHESTWALL specified at the top of the script. In this example, it is 3 cm.

Chestwall\_L:



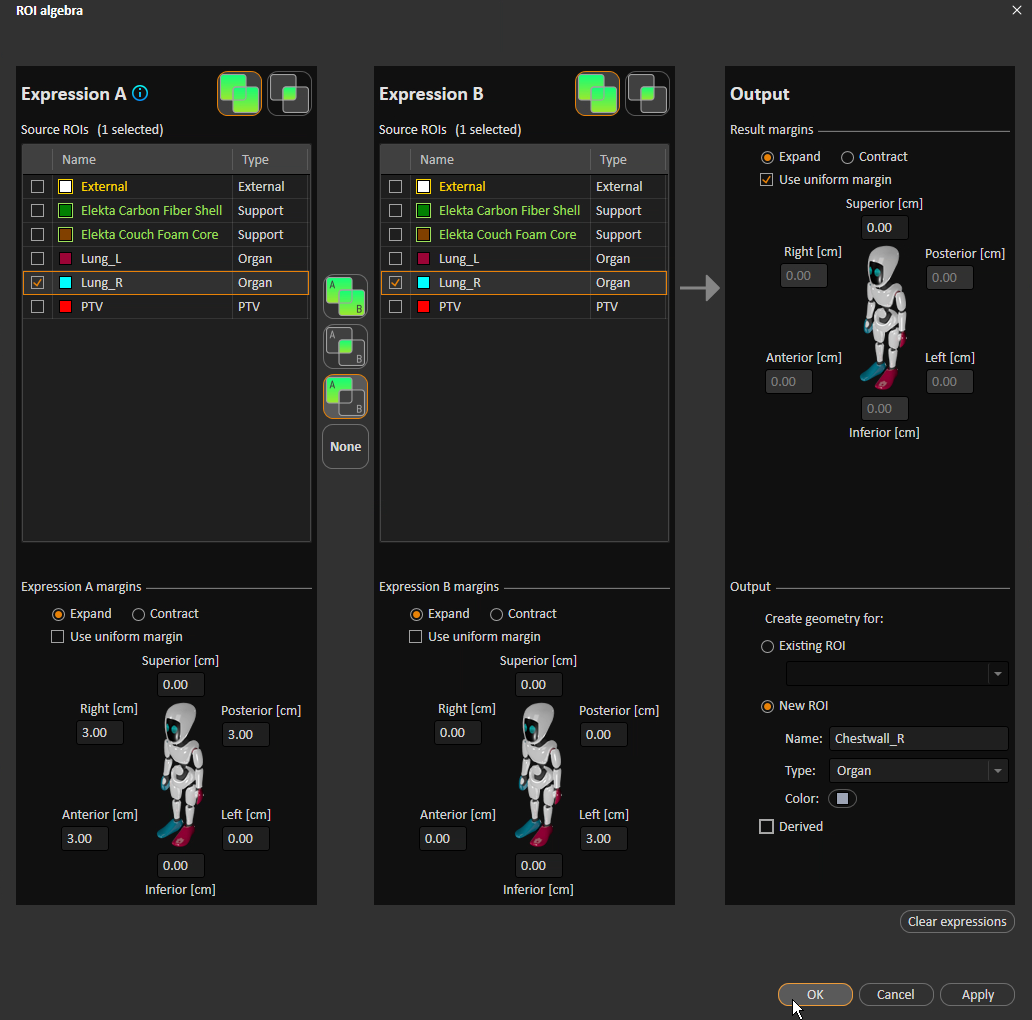
**R expansion**

**P, A, & L expansion**

**Subtraction**

**Ipsilateral lung**

Chestwall\_R:



**Ipsilateral lung**

**Subtraction**

**P, A, & R expansion**

**L expansion**

Crops chestwall geometries to the external geometry.

If both chestwall geometries are successfully added, removes overlap by subtracting a random one of them from the other.

### Assumptions

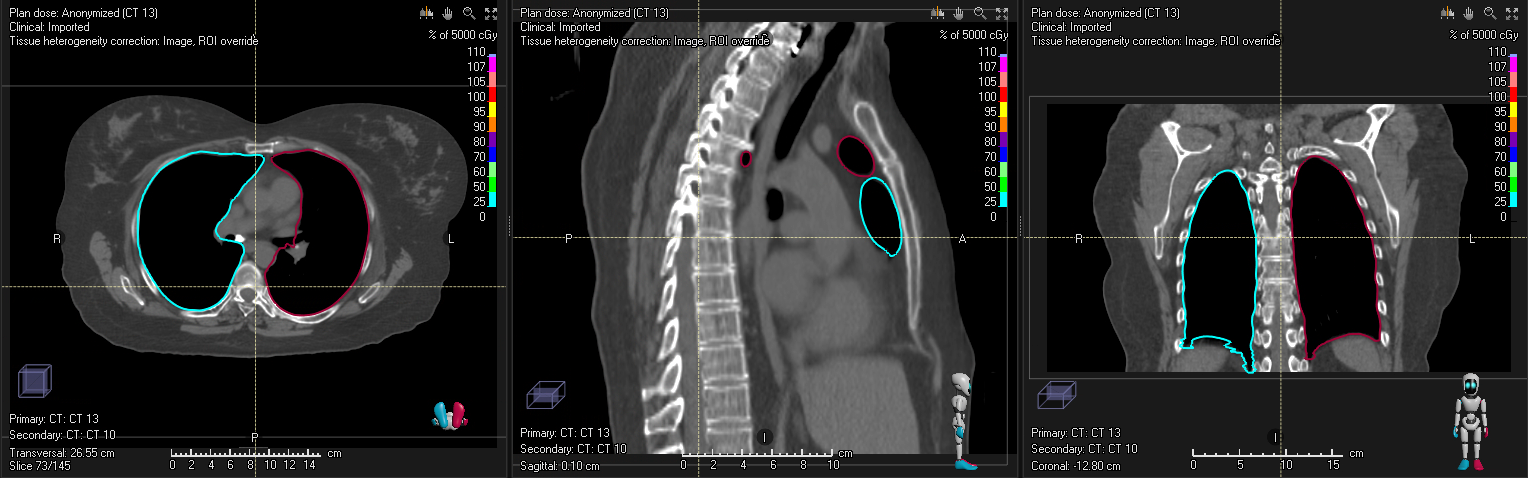
* A case is open.
* The current case has at least one exam.
* The current case has an external ROI.

### Warnings

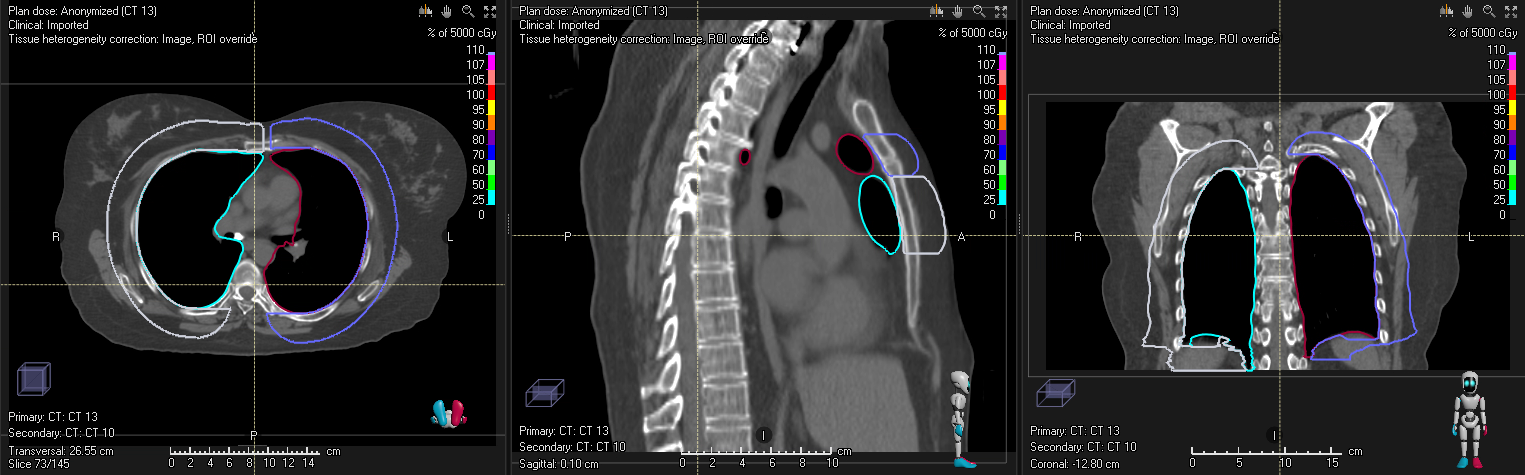
* A chestwall geometry could not be added due to a missing lung geometry.
* A chestwall geometry already exists and is approved.

### Example

Before running the script:



After running the script:



Some cleanup in the inferior direction is obviously necessary.

### Coming Soon

* Add external ROI if it doesn’t exist.
* Add external geometry if it is empty.
* Create Chestwall as the sum of Chestwall\_L and Chestwall\_R.

## Compare Contours

Copies ROI geometries from selected exam(s) to another exam to enable geometry comparison across exams.

### Assumptions

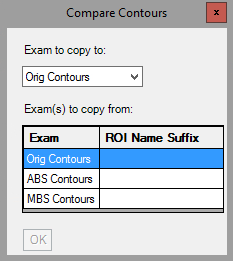
* A case is open.
* The open case has at least one exam.
* At least one exam in the open case has geometries that are not external, support, or fixation.

### Example

We compare the geometries across three exams

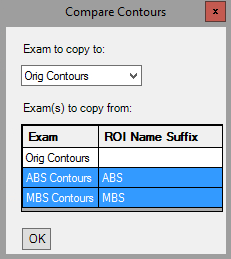
* Orig Contours: Dosimetry’s contours
* ABS Contours: Contours created with the HFS abdomen atlas
* MBS Contours: Contours created using model-based segmentation

The script allows the user to choose the exam(s) to copy geometries from, and the exam to copy contours to. If the current exam has contours, it is by default selected to copy to. Otherwise, no exam is selected. By default, the first row is selected to copy from.



Note that the **OK** button is grayed out. It is enabled only if (a) an exam is selected to copy to, (b) at least one exam is selected to copy from, (c) a suffix is provided for each selected exam to copy from, and (d) each suffix is unique.

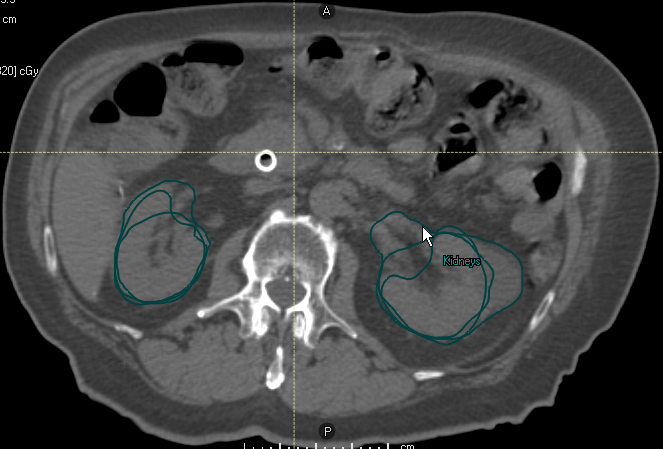
We want to copy geometries from both ABS Contours and MBS Contours to Orig Contours, so we slect those two rows. I’ll add an appropriate suffix for each.



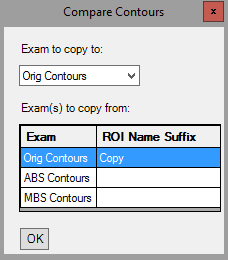
When I click **OK**, the script will create a new ROI for each geometry on ABS Contours. The new ROI name is the same as the old one, plus the suffix according to TG-263 (suffix comes after a carat). For example, if ABS Contours has an ROI called Kidneys, the script creates Kidneys^ABS. The Kidneys geometry is copied into Kidneys^ABS, the Kidneys^ABS geometry is copied from ABS Contours to Orig Contours, and the Kidneys^ABS geometry is deleted from ABS Contours. This process is repeated for each MBS Contours geometry.

Suppose that before running the script, Orig Contours had two geometries, Kidneys and Liver; ABS Contours had a single geometry, Kidneys; and MBS Contours had two geometries, Kidneys and Bag\_Bowel. After running the script, there will be three new ROIs: Kidneys^ABS, Kidneys^MBS, and Bag\_Bowel^MBS, each of which only has contours on Orig Contours.

In the below screenshot of Orig Contours, all geometries are hidden except for Kidneys, Kidneys^ABS, and Kidneys^MBS. We can now easily evaluate the differences among the three geometries.



You can copy from an exam to itself. Suppose the same conditions as above, but instead of copying from ABS Contours or MBS Contours, I copy from Orig Contours. After running the script, each Orig Contours contour will simply be copied into a new ROI with the suffix Copy.



## Convert Virtual Jaw to MLC

Converts virtual jaw fields from external simulation (no MLCs) fields into jaw- and MLC-defined fields.

This script is modified from one provided by RaySearch support. My version supports RayStation 11A and multiple beam sets. A use case for this beam set is converting plans sent to RayStation from sim, into “Initial Sim” plans.

For each beam set in the open plan whose machine is not SBRT 6MV or ELEKTA (CRMC’s two clinical machines as defined in RayStation), a new, single-fraction, conformal beam set is created with the localization point as isocenter and the same exam and patient position as the old beam set. Its beams have the same isocenter (name has a 1 added to the end), beam quality, name, description, gantry and collimator angles, and couch parameters as the old beams. Each new beam has a rectangular field with the same jaw positions as the old beams.

The new beam set’s machine is SBRT 6MV if any of the exams in the case have AVG, MIP, or Gated in their SeriesDescription DICOM attributes. The machine is ELEKTA otherwise.

The new setup beam names are changed to match their descriptions.

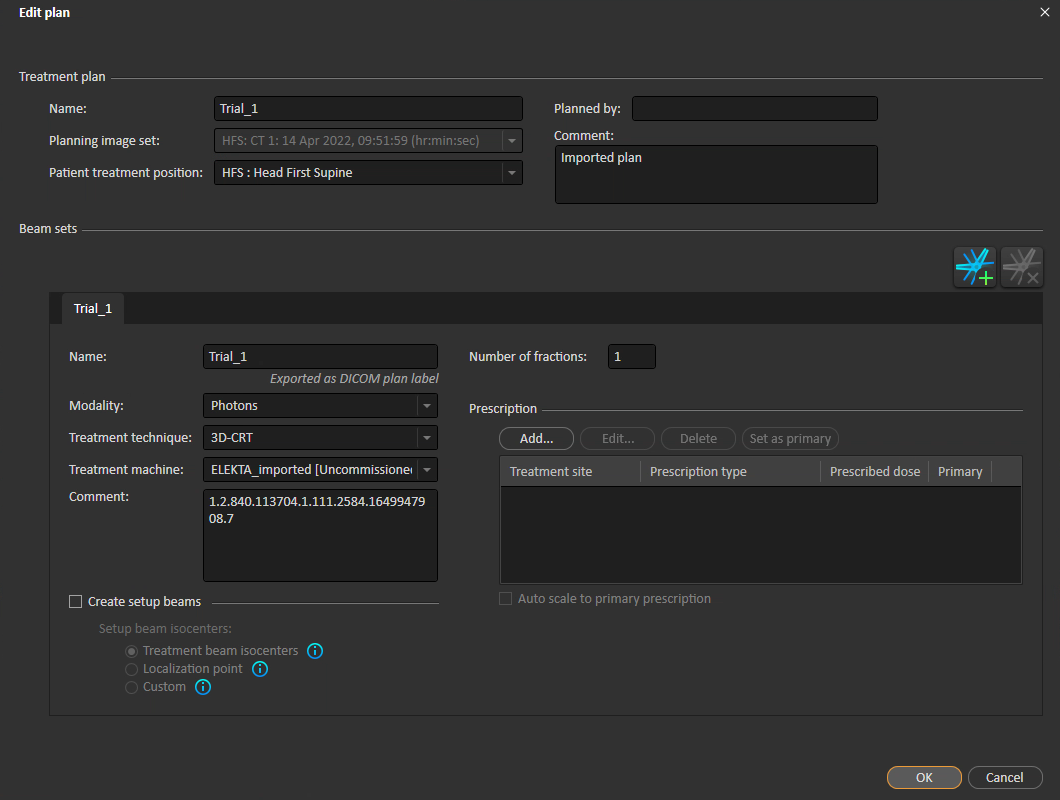
The old beam sets are deleted.

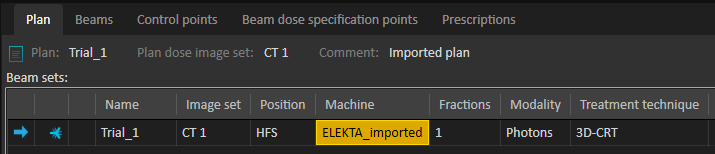
### Assumptions

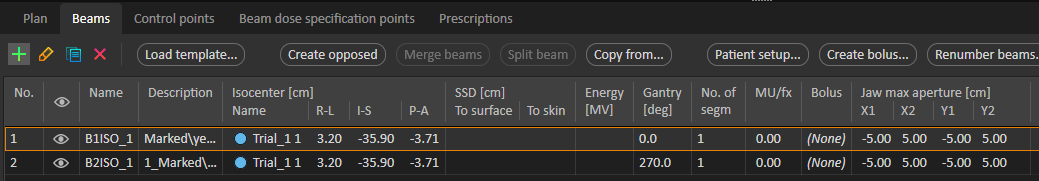
* A patient is open.
* A case is open.
* A plan is open.
* All sim beam sets are photons.

### Example

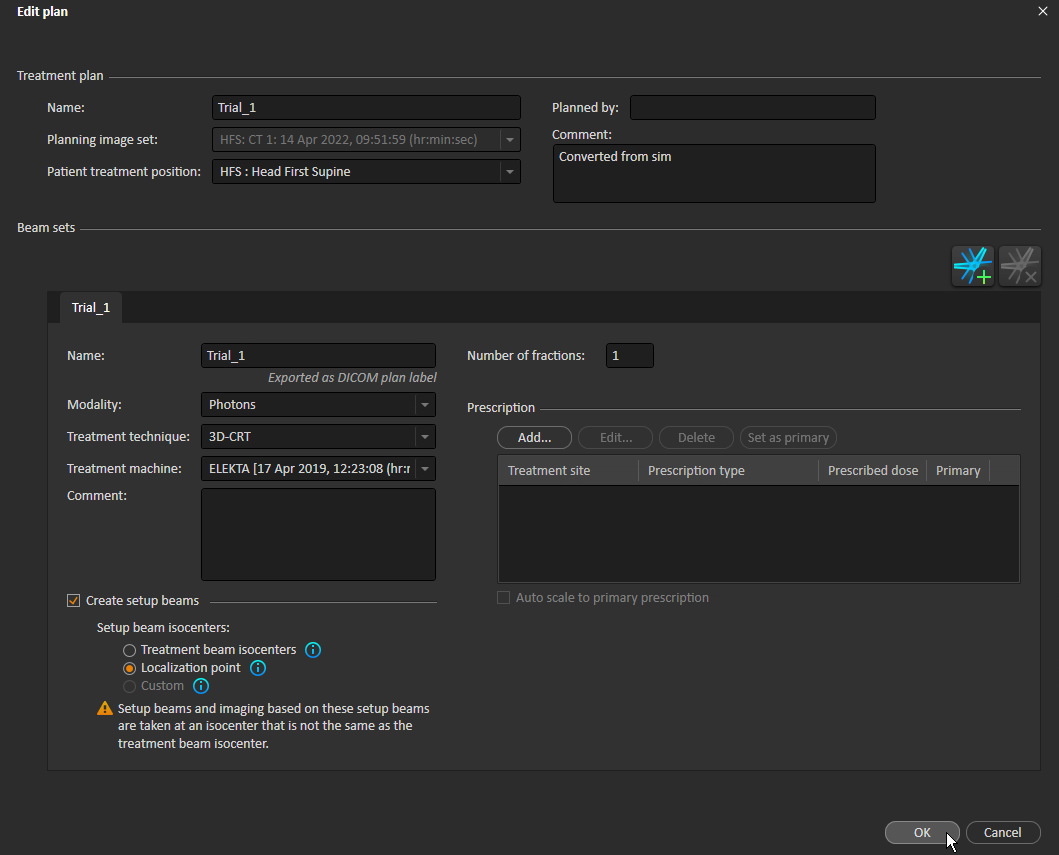
Before running script, we have a plan that we imported from sim:

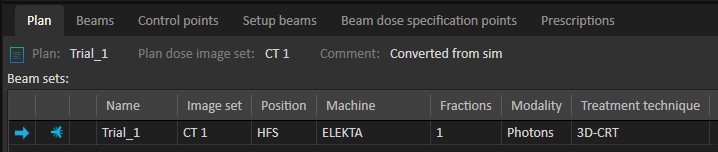


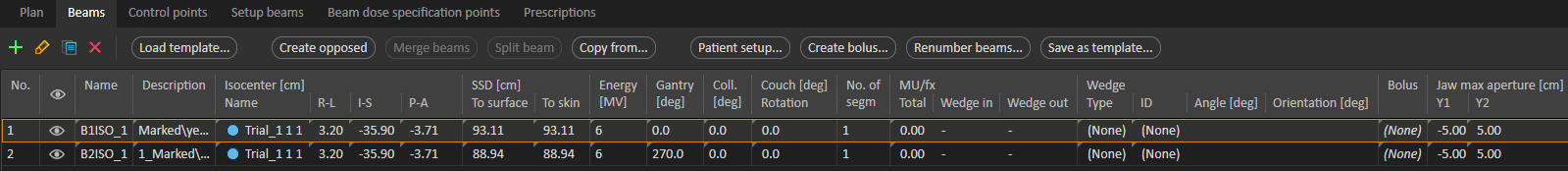




After running the script, note the differences between the plan from the sim, and the converted plan.







## Copy Beam Set

Copies the current beam set. Supports imported doses.

Objectives and constraints, select optimization parameters, and prescriptions are also copied. See [Copy Opt Stuff](#_Copy_Opt_Stuff) for an explanation of how objectives, constraints, and optimization parameters are copied.

### Assumptions

* A patient is open.
* A plan is open.
* The open plan has at least one beam set.
* The current beam set is either photons or electrons.

### Approved Plans

If the current plan is approved, the script offers to copy to a copy of the plan. See [Copy Plan without Changes](#_Copy_Plan_without) for details on how the plan is copied.

### Naming and Numbering Conventions

The new beam set has the same name as the old, made unique across all beam sets in all cases in the current patient. The new beam set name is truncated to 16 characters due to DICOM-imposed length limitations.

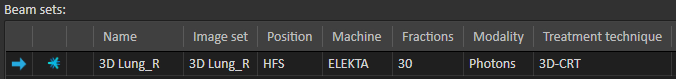
Consider an old beam set called SBRT Lung. If the patient’s other beam sets are 3D Lung\_R and Beam set 1, then the new beam set will be SBRT Lung (1). If SBRT Lung (1) already existed, the new beam set would be SBRT Lung (2). If the old beam set had a long name like Really Long Name and needed to be named Really Long Name (2), the new name would be truncated to Really Long (2) to meet length constraints.

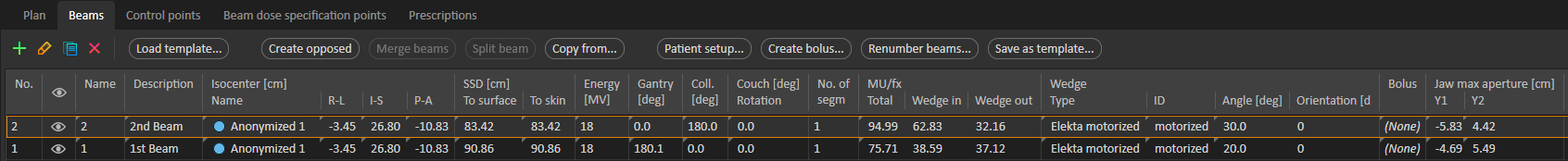
Beam names are treated similarly, but without the 16-character constraint.

Beam numbers are unique across all the patient’s beams and setup beams. The new beam numbers start with a number one greater than the maximum beam or setup beam number in the patient, and continue consecutively from there. New setup beams continue consecutively after the new beams.

### Example

Copy the beam set 3D Lung\_R in the plan 3D Lung\_R. This beam set looks like this:





## Copy Opt Stuff

Copies clinical goals, objectives and constraints, and/or optimization parameters from other plans to the current plan.

### Assumptions

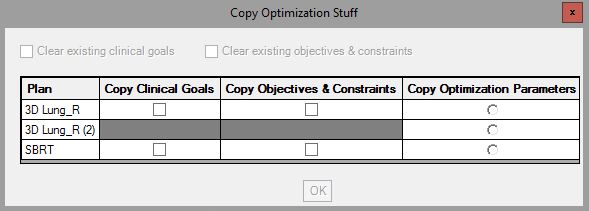
* A patient is open.
* A case is open.
* There is at least one plan in the current case.
* There are at least two plans in the current case.

### Approved Plans

If the plan to copy to is approved, the script offers to copy to a copy of the plan. See [Copy Plan without Changes](#_Copy_Plan_without) for details on how the plan is copied.

### Example

The script launches this window:



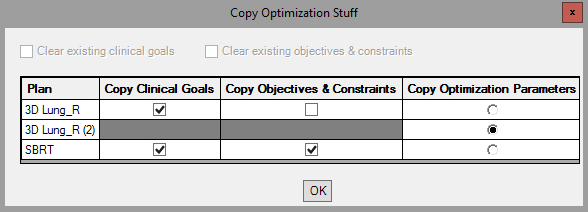
The checkboxes at the top are disabled if the current plan has no goals or objectives/constraints, respectively.

The table allows you to select no, one, or multiple plans from which to copy goals; no, one, or multiple plans from which to copy objectives/constraints; and no or one plan from which to copy optimization parameters. Gray cells indicate plans that don’t have any goals or objectives/constraints. For example, 3D Lung\_R (2) has neither goals nor objectives/constraints.

The **OK** button is enabled only if at least one checkbox or radio button is selected.

The below selections copy the following to the current plan:

* Clinical goals from 3D Lung\_R and SBRT
* Objectives and constraints from SBRT
* Optimization parameters from 3D Lung\_R (2)



### Coming Soon

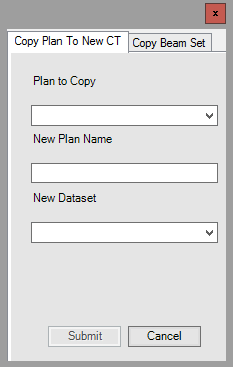
* Support for multi-beam set plans. Currently, script assumes, without warning the user, that the current plan (to copy to) and each plan to copy from contains just one beam set.
* Don’t add duplicate objectives and constraints.
* Add visualization priority to copied clinical goals so the user can see which plan each goal comes from.

## Copy Plan to New CT or Merge Beam Sets

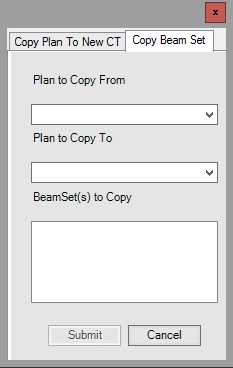
Copies a photon or electron plan to another exam or merges beam sets for plans on the same exam. The new exam must have an external geometry. I modified this script from one provided by RaySearch support. My version works with RayStation 11A and CPython and better adheres to programming best practices and Python style guidelines.

The script presents two tabs:

* **Copy Plan To New CT**

****

* **Copy Beam Set**



### Not a “Real” Copy

The “copying” is less comprehensive than we’d prefer, but this script is the closest that RaySearch could provide. The problem is that we can’t directly copy control points. We must first optimize. But only IMRT plans need optimization. Even for IMRT plans, since optimization is stochastic, we can’t expect to copy the old beam set’s goals, objectives and constraints, and optimization parameters to the new beam set and get the same results. Therefore, before manually copying the control points from the old beam set to the new, the script performs a dummy optimization with a dummy PTV. There is a single objective based on the beam set’s primary prescription:



Another problem is that structures are not copied to the new exam.

### How It Works

1. If copying plan, create a new plan on the new CT. The new plan name is the same as the old, with a copy number for uniqueness.
2. For each old beam set, add a new beam set in the plan that we are copying to. The new beam set names are the same as the old beam set names, with a copy number for uniqueness within the case. The new beam set comments are From <old beam set>, where <old beam set> is the old beam set name. The new beam set has the default dose with the same voxel size as the old dose grid.
3. For each new beam set, add the beams. If copying a plan, the new isocenters are registered between the old and new planning exams. For all beams, beam quality, name, gantry and couch angles match.
   1. For electron beams, the applicator angle and name, insert name and contour, and MU also match.
   2. For non-VMAT photon beams, the collimator angle also matches, and each segment is copied by adding a rectangular field with the same leaf and jaw positions, MU, and relative weight as the old segment.
   3. For VMAT beams, the end gantry angle and arc direction also match. Control points cannot be manually created, so they are added via a dummy optimization. A dummy sphere PTV with radius 2 cm and center at the isocenter has a single optimization objective: uniform dose (the beam set’s primary prescription) with weight 90. The optimization has no preparatory iterations and a single optimization iteration. The beams have the same final arc gantry spacing optimization parameter as the old beams. After the dummy optimization, the beam MU is copied, and control points are copied by copying each segment leaf and jaw positions, dose rate, and relative weight.
4. Compute each new photon beam set. For each new electron beam set, alert the user that histories and prescription must be set before computing dose.

### Assumptions

* A patient is open.
* A case is open.
* If copying (as opposed to merging):
  + There is an external geometry on the planning exam of the plan that is being copied.
  + The plan to be copied has dose.
  + The exam of the plan to be copied, and the exam to copy the plan to, are registered.

### Warnings

* Beam sets that could not be copied/merged because they are neither electrons nor photons

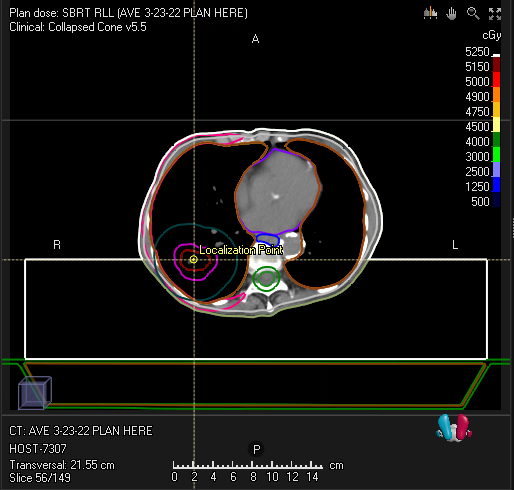
### Coming Soon

* For copying plans, register the old and new planning exams if a registration does not exist.

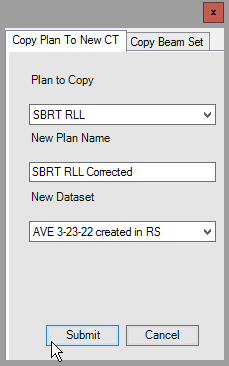
### Example

Here, we copy the plan SBRT RLL from the exam AVE 3-23-22 PLAN HERE to the exam AVE 3-23-22 created in RS.

The old plan:



Set the appropriate values in the script:



A new plan called SBRT RLL Corrected will be created on the exam AVE 3-23-22 created in RS.

## 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.

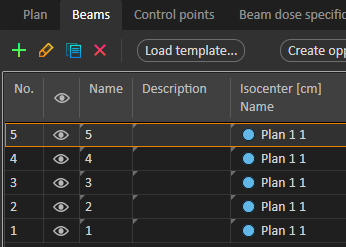
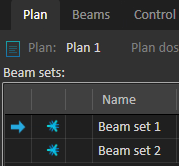
* Name beams the same as their numbers.

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.

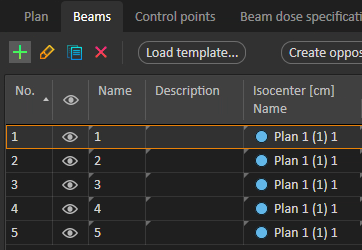
The new plan name is the same as the old plan name plus a copy number (e.g., R Breast (1)).

### Example

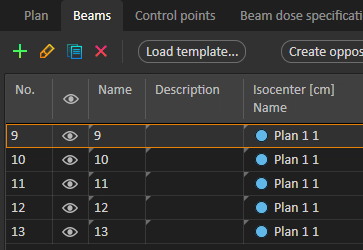
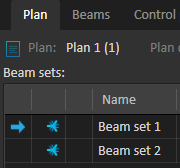
Given a plan Plan 1:



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



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:



## Copy Beam Set

[Coming soon]

## Copy Exam

[Coming soon]

## Create QA Plan

[Coming soon]

## Delete Empty ROIs

[Coming soon]

## Dose Grid Box

Adds a box ROI with geometry that outlines the dose grid of the current beam set.

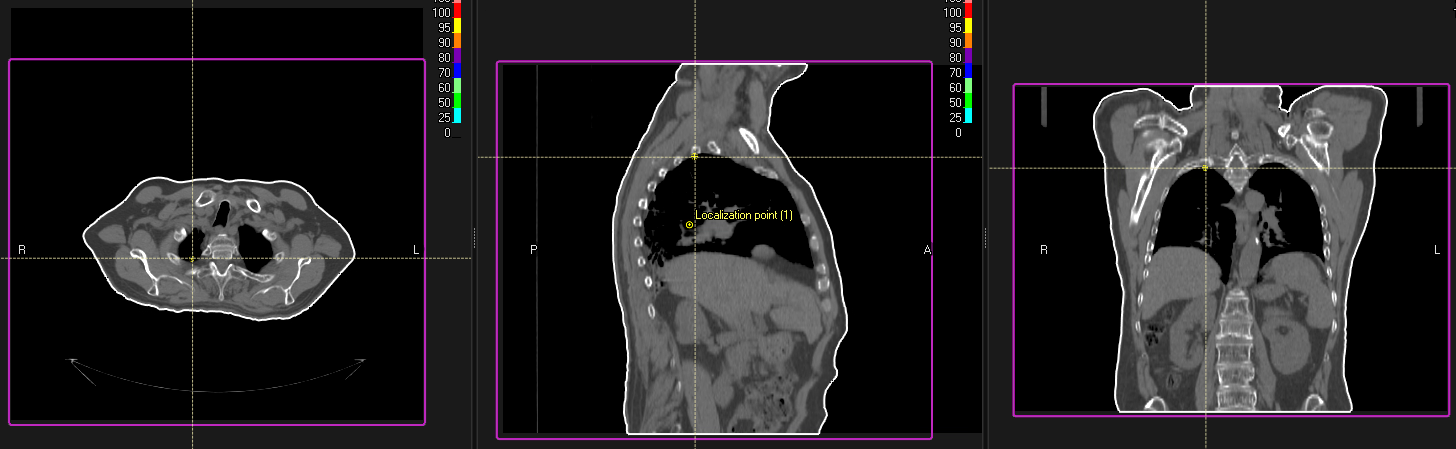
The dose grid outline in RayStation is (sensibly!) faint. When we want the dose grid to be more visually obvious, we outline it with a box ROI geometry.

### Example

The dose grid is faint:



But the purple box is prominent!



## Exclude from MOSAIQ Export

[Coming soon]

## Export DVH Curves

[Coming soon]

## Export Line Doses

[Coming soon]

## Extend Exam

[Coming soon]

## Generate Shifts Comments

[Coming soon]

## Last Modified Times

[Coming soon]

## List Patients

[Coming soon]

## Localize to Beam Iso

[Coming soon]

## Parse Logfiles

Writes information from RayStation logfiles to a table in an Excel spreadsheet.

### Logfile Types

RayStation’s logfiles are a useful troubleshooting tool; RaySearch support often references them when responding to tickets. These logfiles, at least for CRMC’s RayStation installation, are located at C:\ProgramData\RaySearch. There are two types of logfiles:

* Exceptions. These files contain information on a single exception that was thrown. Their file extension is .exn.log.
* General. These files contain information on all logged events, including exceptions. Their file extension is .log. Because these logfiles include exceptions, the script ignores .exn.log files.

### Filepaths

There are two constants in the script that you should change according to your needs:

* LOGFILE\_PATH: The absolute filepath to the RayStation logfiles
* OUTPUT\_PATH: The absolute filepath that the spreadsheet will be in. This path does not have to already exist.

### Spreadsheet Name

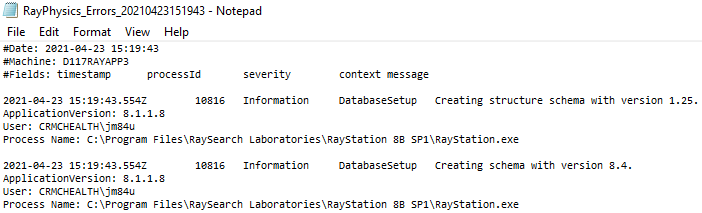
The generated spreadsheet is named Parsed Logfiles YYYY-MM-DD HH\_MM\_SS.xlsx (e.g., Parsed Logfiles 2022-04-22 02\_05\_05.xlsx).

### Data Dictionary

Each row in the table represents a log event. The table contains the following fields:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Source** | **Example Value** |
| Date | The datetime of the log event, in the format M/D/YYYY H:MM AM/PM. | The timestamp field of the event. | 4/22/2022 3:12 PM |
| App | The RayStation app that generated the event. | The first part of the logfile name. | RayStationDoctor |
| App Version | The RayStation version that generated the event.  Is not present in RayStation\_Storage\_Tool events. | The ApplicationVersion field of the event. | 8.1.1.8 |
| User | The username of the user who generated the event  Is not present in RayStation\_Storage\_Tool events. | The User field of the event. | CRMCHEALTH\jm84u |
| Machine | The computer that generated the event. | The Machine header field in the logfile. | D117RAYAPP1 |
| Severity | The severity level of the event. | The severity field of the event. | Information |
| Type | The type of event. | The context field of the event. | General |
| Exception | The exception that the event was logged for.  Does not apply to all events. | Part of the message field of the event. | RaySearch.CoreUtilities.LogicalException |
| Exception Name | The user-friendly name of the exception. Only present for exceptions that involve a BROKEN RULE. | The Name field of an exception’s BROKEN RULE. | Validate isocenter name |
| Message/Description | The message associated with the event.  For RayStation Scripting Log events, this is always Using scripting environment ‘\_\_\_’. Because Scripting Env is another field, Message/Description is not listed for RayStation Scripting Log events. | Part or all of the message field of the event. For exceptions, also includes the Description field of the event’s BROKEN RULE, if present. | No licenses available; Error during license checkout of feature: 11\_0\_1-CLI-rayArc-4b79ac7aadf54e1e87f9ac345ecda85e |
| Server | The SQL server on which the event occurred. Only applies to events that raised a SqlException. | The Server field of the event. | D117RAYSQL\raystation |
| Scripting Env | The scripting environment in which the event occurred. Only applies to RayStation Scripting Log events. | The scripting environment name in the event’s message field. | CPython 3.6 (64-bit) |
| Script | The Python script that generated the event. Only applies to RayStation Scripting Log events. | The script name in the event line Running ‘\_\_\_’ from database. | CopyPlanandBeamSet11A |
| Pt | The current patient when the event was generated. Only applies to RayStation Scripting Log events. | The patient name in the event line Current patient is patient with name: \_\_\_ | Test^Master |
| Filename | The absolute filepath of the logfile. |  | C:\ProgramData\RaySearch\RayStationPlanning\_Errors\_20220117210633.log |
| Full Text | The full text of the event (from which most other fields were extracted). |  | "2022-03-24 18:28:10.969Z 236 Information RayStation Scripting Log Using scripting environment 'CPython 3.6 (64-bit)'  ⋮  Patient UUID: 6c3943df-26b6-44ee-ba55-b3e38dfb0c70" |

### Example Events



**App**

**Machine**

**Date**

**Severity**

**Type**

**Message/Description**

**App Version**

**User**

**Event**

**Event**

**App**

**Date**

**Machine**

**Severity**

**Type**

**Scripting Env**

**Script**

**Pt**

**App Version**

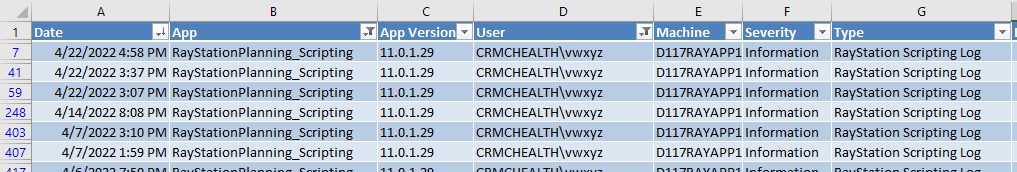
**User**

**Event**



### Example

Suppose user vwxyz says that the RayStation Planning scripts keep erroring out on her today. I can run Parse Logfiles and sort the resulting table by newest to oldest Dateand then filter by App (RayStationPlanning\_Scripting) and User (abcde).

 ⋮

…

## Plan Check

[Coming soon]

## Plan Quality Metrics

[Coming soon]

## Prepare Exams

[Coming soon]

## Print Report

[Coming soon]

## QACT Adaptive Analysis

[Coming soon]

## TG-263 Names & CRMC Colors

[Coming soon]

## Scale Objectives & Constraints

[Coming soon]

## Show Non-Empty Geometries

[Coming soon]

## Update Clinical Goals Templates

[Coming soon]

## Update Objectives & Constraints Templates

[Coming soon]

# 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](https://superuser.com/questions/15596/automatically-run-a-script-when-i-log-on-to-windows), [Mark McClelland](https://poetengineer.postach.io/), and [Christoph Kolbicz](https://kolbi.cz/).

### Dependencies

* The browser you want to set as the default. To use the solution as is, this is Chrome.
* Christoph Kolbicz's [SetDefaultBrowser.exe](https://kolbi.cz/blog/2017/11/10/setdefaultbrowser-set-the-default-browser-per-user-on-windows-10-and-server-2016-build-1607)
* 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:



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.

# Appendix: Common RayStation Script Error Messages

When the script cannot continue for some reason, you will be prompted to end the script or take some other action. These messages do not mean that the script tried to do something and messed up, but that because a certain assumption is violated, it will not try. Here are some “error” messages, why they occur, where to check them in RayStation, and how to fix them if possible.

## Incorrect Modality

