

# Adding Parameters

We have tried to design the system so that adding parameters does not require knowledge of the whole program. This document will describe where any new parameters should be added and what data will be accessible when extracting and evaluating them.

## Adding a parameter

[Video walkthrough in Spyder](#)

1. Before you write the functions, you may want to review the truth table values for the new parameter and see if they are acceptable.
  - a. Keep the format of the truth table values in mind when writing the extraction and evaluation functions.
2. You will need to write a function in `code/parameters/extractor_functions.py` to extract the information you want from the dicom files.
  - a. Each function has access to the dataset (RTPLAN data), the case, and potentially the RTDOSE and RTSTRUCT corresponding to the RTPLAN.
3. Then, add it to the exported functions (`extractor_functions`) at the bottom of the file.
4. You will probably want to evaluate it. This means writing an evaluation function in `code/parameters/evaluator_functions.py`.
  - a. Each function will have access to the parameter value, the corresponding truth table value, as well as a group of arguments in "kwargs". This group contains the complete parameter values, complete truth table, and case.
5. That's it! You've added a new parameter to the output.

Note that this is for parameters that are listed but not implemented. To add a completely new parameter, you will need to add entries in `strings.py` for the variable "parameters", as well as the dictionaries at the bottom of `extractor_functions.py` and `evaluator_functions.py`

## Editing existing parameters

1. Edit one of the existing extraction functions.
2. Then, with regard to how the data format has changed, edit the corresponding evaluation function.
  - a. For example, `_extract_prescription_dose()` currently returns a string of the format "dose/fractions/unit", however if you wanted to split them up into separate parameters for whatever reason, you would have 3 extraction functions. Then the function `_evaluate_prescription_dose()` would likewise have to be split up into 3 corresponding functions.