

---

# **HADDOCK2.4 param file tools Documentation**

*Release 0.1alpha*

**Mikael Trellet**

**Nov 07, 2018**



---

## Contents

---

<b>1</b>	<b>API Reference</b>	<b>3</b>
<b>2</b>	<b>Scripts</b>	<b>5</b>
<b>3</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>



This document aims to provide some usage and info about the different ways provided by the CSB team to manipulate or visualise HADDOCK2.4 new parameter files. Those new parameter files are associated with all your HADDOCK runs performed with the portal and are named by default *job\_params.json*.

The tools are split in two categories:

- **scripts**

Standalone scripts that can be used at the command-line level and piped together to perform actions on a parameter file. They usually take a JSON parameter file as input and output the same file with the change required.

- **API - param\_to\_json**

This API lets you manipulate HADDOCK2.4 parameter files at the Python level. It allows you to perform changes on your parameters within your own Python scripts. It encapsulates the parameter file into a HADDOCKParam python class that ensures for the validity and integrity of your parameters.



## 1.1 The `param_to_json` Python module

Representation of a HADDOCK parameter file as a Python object.

This library allows to create, load, modify and write HADDOCK parameter files in JSON format.

Several functions are made available and are based on a common structure `HADDOCKParam` that needs to be initialized prior to doing anything.

```
class param_to_json.HADDOCKParamError (message)
    Exception class related to any error within the HADDOCKParam class

class param_to_json.HADDOCKParamFormatError (message, param=None)
    Exception class related to any format error within the HADDOCKParam class

class param_to_json.HADDOCKParam (verbose=True)
    Top-level class representing a complete HADDOCK parameter file.

        Parameters verbose (bool) – Get validation details and warnings

check_status ()
    Check if a HADDOCKParam is loaded and valid

get (param)
    Get value of a parameter using its name

        Parameters param – Name of the parameter

load (input, skip_validation=False)
    Load the parameter file in a HADDOCKParam object

        Parameters

            • input (str, file) – JSON file path or file-object

            • skip_validation (bool) – Flag to skip or not the validation step
```

**set** (*param, value*)

Get value of a parameter using its name

**Parameters** *param* – Name of the parameter

**validate** (*init=False*)

Validation of parameter file format and keys

**Param** *bool init*: Indicate if validation occurs upon loading file

**Returns** True/False

**Return type** bool

**Raise** HADDOCKParamFormatError



- **haddock\_param\_summary.py**

Get a quick summary of the job parameter file (JSON)

**usage:**

```
$> python haddock_param_summary.py <json file>
```

**example:**

```
$> python haddock_param_summary.py job_params.json
it0 1000
it1 20
itw 20
Partner1: Protein
Partner2: Protein
clust_meth: FCC
clust_cutoff: 0.6
```

This script is supposed to work with the new parameter files used in HADDOCK2.4 (JSON format).

- **haddock\_param\_extract\_pdb.py**

Extract the PDB input files from a job parameter file (JSON) in the current directory

**usage:**

```
$> python haddock_param_summary.py <json file>
```

**example:**

```
$> python haddock_param_summary.py job_params.json
partner1.pdb created
partner2.pdb created
```

This script is supposed to work with the new parameter files used in HADDOCK2.4 (JSON format).

- **haddock\_param\_validate.py**

Validate a HADDOCK parameter file (JSON) and returns any WARNING/ERRORS found.

**usage:**

```
$> python haddock_param_validate.py [-v/--verbose] <json file>
```

**example:**

```
$> python haddock_param_validate.py job_params.json
WARNING: No partner detected
ERROR: Wrong type for parameter 'amb_cool2'
...
```

This script is supposed to work with the new parameter files used in HADDOCK2.4 (JSON format).

- **haddock\_param\_replace.py**

Replace a parameter in a HADDOCK parameter file (JSON), returns the full parameter file with the modified parameter. Parameter must be int, float, string or boolean

**usage:**

```
$> python haddock_param_replace.py <param_name> <param_new_value> <json file>
```

**example:**

```
$> python haddock_param_replace.py amb_cool1 20 job_params.json
{
    'amb_cool1': 20,
    'amb_cool2': 50,
    ...
}
```

This script is supposed to work with the new parameter files used in HADDOCK2.4 (JSON format).

## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



**h**

haddock\_param\_extract\_pdb, 5  
haddock\_param\_replace, 6  
haddock\_param\_summary, 5  
haddock\_param\_validate, 6

**p**

param\_to\_json, 3



## C

check\_status() (param\_to\_json.HADDOCKParam method), 3

## G

get() (param\_to\_json.HADDOCKParam method), 3

## H

haddock\_param\_extract\_pdb (module), 5

haddock\_param\_replace (module), 6

haddock\_param\_summary (module), 5

haddock\_param\_validate (module), 6

HADDOCKParam (class in param\_to\_json), 3

HADDOCKParamError (class in param\_to\_json), 3

HADDOCKParamFormatError (class in param\_to\_json),  
3

## L

load() (param\_to\_json.HADDOCKParam method), 3

## P

param\_to\_json (module), 3

## S

set() (param\_to\_json.HADDOCKParam method), 3

## V

validate() (param\_to\_json.HADDOCKParam method), 4