

---

# **grg-grgdata Documentation**

***Release 0.2.3***

**Carleton Coffrin**

**Aug 09, 2019**



---

## Contents:

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview . . . . .	1
1.2	Installation . . . . .	1
1.3	Testing . . . . .	1
<b>2</b>	<b>grg-grgdata package</b>	<b>3</b>
2.1	grg_grgdata.io module . . . . .	3
2.2	grg_grgdata.cmd module . . . . .	3
2.3	grg_grgdata.exception module . . . . .	5
2.4	grg_grgdata.struct module . . . . .	5
2.5	Module contents . . . . .	5
<b>3</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



### 1.1 Overview

grg-grgdata is a minimalist python package for working with of **GRG** network data files.

The primary entry point of the library is `grg_grgdata.cmd` module, which contains the methods for GRG data validation.

### 1.2 Installation

Simply run:

```
pip install grg-grgdata
```

### 1.3 Testing

grg-grgdata is designed to be a library that supports other software. It is not immediately useful from the terminal. However, you can test the validation functionality from the command line with:

```
python -m grg_pssedata.cmd <path to GRG data file>
```

If this command is successful, you will see a message indicating that the given data is a valid GRG file printed to the terminal.



## 2.1 grg\_grgdata.io module

`grg_grgdata.io.abstract_value_to_string(abs_value)`

`grg_grgdata.io.build_cli_parser()`

`grg_grgdata.io.dict_to_list(name, data, keys)`

`grg_grgdata.io.main(args)`

reads a matpower or grg case files and processes them based on command line arguments.

**Parameters** `args` – an argparse data structure

`grg_grgdata.io.parse_grg_case_file(grg_file_name)`

opens the given path and parses it as json data

**Parameters** `grg_file_name` (*str*) – path to the a json data file

**Returns** a dictionary case

**Return type** Dict

`grg_grgdata.io.print_json(grg_data, selection=None)`

`grg_grgdata.io.print_json_selection(data, prefix, selection)`

`grg_grgdata.io.print_tabular_summary(grg_data)`

`grg_grgdata.io.print_tabular_summary_network(grg_data)`

`grg_grgdata.io.value_to_string(value)`

## 2.2 grg\_grgdata.cmd module

`grg_grgdata.cmd.active_voltage_points(grg_data, switch_status={})`

`grg_grgdata.cmd.apply_assignment(network_data, pointer_string, value)`

```
grg_grgdata.cmd.build_cli_parser()
grg_grgdata.cmd.bus_voltage_points(grg_data)
grg_grgdata.cmd.check_flow_limit_bound(identifier, ac_line_data, ad_lookup, vl_lookup,
                                         per_unit)
grg_grgdata.cmd.check_property(status, feedback)
grg_grgdata.cmd.check_voltage_level(identifier, ac_line_data, voltage_level_lookup)
grg_grgdata.cmd.collapse_voltage_points(grg_data, switch_status={})
grg_grgdata.cmd.components_by_type(grg_data)
grg_grgdata.cmd.flatten_network(grg_data, transformation_id)
grg_grgdata.cmd.isolated_voltage_points(grg_data, switch_status={})
grg_grgdata.cmd.lookup_network(grg_data, transformation_id)
grg_grgdata.cmd.lookup_pointer(grg_data, pointer_string)
grg_grgdata.cmd.main(args)
    reads a GRG case file and runs the GRG data validation and parameter checks

    Parameters args – an argparse data structure

grg_grgdata.cmd.print_err()
    print(value, ..., sep=' ', end='n', file=sys.stdout)

    Prints the values to a stream, or to sys.stdout by default. Optional keyword arguments: file: a file-like object
    (stream); defaults to the current sys.stdout. sep: string inserted between values, default a space. end: string
    appended after the last value, default a newline.

grg_grgdata.cmd.validate_grg(grg_data)
grg_grgdata.cmd.validate_grg_ac_line(identifier, ac_line_data, per_unit)
grg_grgdata.cmd.validate_grg_bus(identifier, bus_data, per_unit)
grg_grgdata.cmd.validate_grg_dc_line(identifier, dc_line_data)
grg_grgdata.cmd.validate_grg_flow_limit(identifier, limit_name, comp_data)
grg_grgdata.cmd.validate_grg_flow_limit_bound(identifier, limit_name, comp_data,
                                              limit_bound)
grg_grgdata.cmd.validate_grg_generator(identifier, gen_data)
grg_grgdata.cmd.validate_grg_intertie(identifier, intertie_data)
grg_grgdata.cmd.validate_grg_load(identifier, load_data)
grg_grgdata.cmd.validate_grg_parameters(grg_data)
grg_grgdata.cmd.validate_grg_shunt(identifier, shunt_data)
grg_grgdata.cmd.validate_grg_switch(identifier, switch_data)
grg_grgdata.cmd.validate_grg_synchronous_condenser(identifier, sync_cond_data)
grg_grgdata.cmd.validate_grg_three_winding_transformer(identifier, thwt_data)
grg_grgdata.cmd.validate_grg_two_winding_transformer(identifier, twt_data, per_unit)
grg_grgdata.cmd.validate_pointer(pointer_string, grg_data, component_lookup, context=[], as-
                                signment=False)
grg_grgdata.cmd.voltage_level_by_voltage_point(grg_data)
```



```
grg_grgdata.cmd.votlage_level_lookup(grg_data)
grg_grgdata.cmd.walk_assignments(grg_data)
grg_grgdata.cmd.walk_components(grg_data)
grg_grgdata.cmd.walk_fault_lists(grg_data)
grg_grgdata.cmd.walk_operation_constraints(grg_data)
grg_grgdata.cmd.walk_pointers(grg_data)
grg_grgdata.cmd.walk_time_series_assignments(grg_data)
grg_grgdata.cmd.walk_voltage_links(grg_data)
```

## 2.3 grg\_grgdata.exception module

a collection of all grg\_grgdata exception classes

**exception** grg\_grgdata.exception.GRGDataException

Bases: exceptions.Exception

root class for all GRGData Exceptions

**exception** grg\_grgdata.exception.GRGDataValidationError

Bases: *grg\_grgdata.exception.GRGDataException*

for errors that occur while attempting to validate the correctness of a parsed GRG data file

**exception** grg\_grgdata.exception.GRGDataWarning

Bases: exceptions.Warning

root class for all GRG data warnings

## 2.4 grg\_grgdata.struct module

## 2.5 Module contents

a package for reading and writing of grid data files



## CHAPTER 3

---

### Indices and tables

---

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



### g

- `grg_grgdata`, 5
- `grg_grgdata.cmd`, 3
- `grg_grgdata.exception`, 5
- `grg_grgdata.io`, 3
- `grg_grgdata.struct`, 5



## A

`abstract_value_to_string()` (in module `grg_grgdata.io`), 3  
`active_voltage_points()` (in module `grg_grgdata.cmd`), 3  
`apply_assignment()` (in module `grg_grgdata.cmd`), 3

## B

`build_cli_parser()` (in module `grg_grgdata.cmd`), 4  
`build_cli_parser()` (in module `grg_grgdata.io`), 3  
`bus_voltage_points()` (in module `grg_grgdata.cmd`), 4

## C

`check_flow_limit_bound()` (in module `grg_grgdata.cmd`), 4  
`check_property()` (in module `grg_grgdata.cmd`), 4  
`check_voltage_level()` (in module `grg_grgdata.cmd`), 4  
`collapse_voltage_points()` (in module `grg_grgdata.cmd`), 4  
`components_by_type()` (in module `grg_grgdata.cmd`), 4

## D

`dict_to_list()` (in module `grg_grgdata.io`), 3

## F

`flatten_network()` (in module `grg_grgdata.cmd`), 4

## G

`grg_grgdata` (module), 5  
`grg_grgdata.cmd` (module), 3  
`grg_grgdata.exception` (module), 5  
`grg_grgdata.io` (module), 3  
`grg_grgdata.struct` (module), 5

`GRGDataException`, 5  
`GRGDataValidationError`, 5  
`GRGDataWarning`, 5

## I

`isolated_voltage_points()` (in module `grg_grgdata.cmd`), 4

## L

`lookup_network()` (in module `grg_grgdata.cmd`), 4  
`lookup_pointer()` (in module `grg_grgdata.cmd`), 4

## M

`main()` (in module `grg_grgdata.cmd`), 4  
`main()` (in module `grg_grgdata.io`), 3

## P

`parse_grg_case_file()` (in module `grg_grgdata.io`), 3  
`print_err()` (in module `grg_grgdata.cmd`), 4  
`print_json()` (in module `grg_grgdata.io`), 3  
`print_json_selection()` (in module `grg_grgdata.io`), 3  
`print_tabular_summary()` (in module `grg_grgdata.io`), 3  
`print_tabular_summary_network()` (in module `grg_grgdata.io`), 3

## V

`validate_grg()` (in module `grg_grgdata.cmd`), 4  
`validate_grg_ac_line()` (in module `grg_grgdata.cmd`), 4  
`validate_grg_bus()` (in module `grg_grgdata.cmd`), 4  
`validate_grg_dc_line()` (in module `grg_grgdata.cmd`), 4  
`validate_grg_flow_limit()` (in module `grg_grgdata.cmd`), 4

`validate_grg_flow_limit_bound()` (*in module grg\_grgdata.cmd*), 4  
`validate_grg_generator()` (*in module grg\_grgdata.cmd*), 4  
`validate_grg_intertie()` (*in module grg\_grgdata.cmd*), 4  
`validate_grg_load()` (*in module grg\_grgdata.cmd*), 4  
`validate_grg_parameters()` (*in module grg\_grgdata.cmd*), 4  
`validate_grg_shunt()` (*in module grg\_grgdata.cmd*), 4  
`validate_grg_switch()` (*in module grg\_grgdata.cmd*), 4  
`validate_grg_synchronous_condenser()` (*in module grg\_grgdata.cmd*), 4  
`validate_grg_three_winding_transformer()` (*in module grg\_grgdata.cmd*), 4  
`validate_grg_two_winding_transformer()` (*in module grg\_grgdata.cmd*), 4  
`validate_pointer()` (*in module grg\_grgdata.cmd*), 4  
`value_to_string()` (*in module grg\_grgdata.io*), 3  
`voltage_level_by_voltage_point()` (*in module grg\_grgdata.cmd*), 4  
`voltage_level_lookup()` (*in module grg\_grgdata.cmd*), 4

## W

`walk_assignments()` (*in module grg\_grgdata.cmd*), 5  
`walk_components()` (*in module grg\_grgdata.cmd*), 5  
`walk_fault_lists()` (*in module grg\_grgdata.cmd*), 5  
`walk_operation_constraints()` (*in module grg\_grgdata.cmd*), 5  
`walk_pointers()` (*in module grg\_grgdata.cmd*), 5  
`walk_time_series_assignments()` (*in module grg\_grgdata.cmd*), 5  
`walk_voltage_links()` (*in module grg\_grgdata.cmd*), 5