# $\mathbf{py}_{d}ss_{i}nterface$ Release 1.0.2

# Contents

1	Over 1.1 1.2 1.3 1.4	Disclair Installat Docume Thanks	tion entati	on .	 			 	 	  	 			 	 		  		1 1 1 2 2
2	Insta	llation																	3
3	How 3.1 3.2	to Use p py-dss-i Simple	interf	ace Py	ythor	n pac	kag	e.	 										<b>5</b> 5 5
4	<b>src</b> 4.1	py_dss_4.1.1 4.1.2 4.1.3 4.1.4 4.1.5 4.1.6 4.1.7 4.1.8 4.1.9 4.1.10 4.1.11 4.1.12 4.1.13 4.1.14 4.1.15 4.1.16 4.1.17 4.1.18 4.1.19 4.1.20 4.1.21 4.1.22 4.1.23	Acti Bus Cap Cap Circ Cktll CMI Ctrll DSS DSS DSS DSS Erro Gen ISou Line Load Met	acitor Contrauit ElementathLib Queue SElem SExect SInterf SProper orInter es erator irces eCode	ass														7 7 7 7 8 11 13 16 20 23 25 26 26 26 28 29 30 32 35 36 39 42 48 50 54

		4.1.24 4.1.25 4.1.26 4.1.27 4.1.28 4.1.29 4.1.30 4.1.31 4.1.32 4.1.33 4.1.34 4.1.35 4.1.36 4.1.37 4.1.38 4.1.39	Pa PI PV Rec Rec See See So Tee Tr	DEldySygeclovescope of the second sec	el r eme ster sers ontr s on ontr ogy form rees urve	rols rols rols	33								 									56 58 60 61 63 65 68 70 72 74 80 82 82 83 87 88
<ul><li>5</li><li>6</li></ul>	5.1 5.2 5.3 Auth	Bug rep Docume Feature	port enta	atio	n im	npro	ove	mei	nts	 			 							 				91 91 91 91
7	7.1 7.2 7.3 7.4 7.5 7.6 7.7 7.8 7.9	1.1.0 (2 1.0.0 (2 0.1.0 (2 0.0.9 (2 0.0.8 (2 0.0.7 (2 0.0.4 (2 0.0.1 (2 0.0.0 (2	2021 2021 2020 2020 2020 2020	-01  -01  -01  -12  -10  -08	-21 -21 -21 -10 -22 -17 5-12						    	    	· · · · · · · · · · · · · · · · · · ·	 · · · · · · · ·	 · · · · · · · · · ·		 	· · · · · · · · · · · · · · · · · · ·		 	· · · · · · · · · · · · · · · · · · ·	 · · · · · · · · · · · · · · · · · · ·		95 95 95 95 96 96 96 96
8	Indic	ces and ta	abl	es																				97
Py	thon I	Module I	Ind	ex																				99
Inc	dex																						1	01

				4
CH	ΔF	TF	R	

Overview

doop	
docs	
docs tests	
	Irequires
package	

py-dss-interface is a Windows Python package providing access to OpenDSS direct dll version of OpenDSS - Version 9.2.0.1 (64-bit build); License Status: Open and Version 9.2.0.1 (32-bit build); License Status: Open.

• Free software: MIT license

# 1.1 Disclaimer

This Python Package is purely responsibility of Paulo Radatz and not his employer. Use this package at your own risk.

### 1.2 Installation

pip install py-dss-interface

# 1.3 Documentation

You can access the documentation through:

1 - The Read the Docs.

https://py\_dss\_interface.readthedocs.io/

2 - Another good resource is the OpenDSS\_Direct\_DLL.pdf doc created by Davis Montenegro. The package has been done based on this documentation.

https://sourceforge.net/p/electricdss/code/HEAD/tree/trunk/Version8/Distrib/Doc/OpenDSS\_Direct\_DLL.pdf

#### 1.4 Thanks

I want to thank Ênio Viana and Rodolfo Pilar Londero for all their contribution to the new version of the tool.

_			$\mathbf{O}$
CHA	١P	FF	$\mathbf{R}$

Installation

#### At the command line:

pip install py-dss-interface

# CHAPTER 3

## How to Use py-dss-interface Python package

## 3.1 py-dss-interface Python package

py-dss-interface is a Windows Python package providing access to OpenDSS direct dll.

## 3.2 Simple Usage

First import the Package

Creates an OpenDSS object

```
OpenDSS Started successfully!
OpenDSS Version 9.3.0.1 (64-bit build); License Status: Open
```

If you want to use your OpenDSS, you will need to pass the OpenDSS path as argument to the DSSDLL class, as can be seen below:

Creates a dss object with your OpenDSS

```
OpenDSS Started successfully!
OpenDSS Version 9.2.0.1 (64-bit build); License Status: Open
```

Select the DSS model

Compile

Solve - You can use the text interface as well: dss.text("solve")

```
0
```

Show Voltage Report

1.1

#### Get all buses voltages

```
[57499.9999611886, 33197.64035949601, -0.00013449617643016606, -66395.28088703856, -
→57499.99982669239, 33197.64054049976, 2521.7954836507547, -0.11910264415748555, -
→1245.983653171188, -2157.930119652756, -1260.782309001901, 2184.0080773974814, 2461.
→932156820345, -76.36035906329018, -1284.5460767592406, -2134.6137221915546, -1162.
→2387444138396, 2149.367176421046, 2447.622527822158, -105.33323048056116, -1292.
→0799754959983, -2145.0014944867958, -1135.8268535116601, 2139.8329412956255, 2416.
 \hspace{2.5cm} \boldsymbol{\to} 0125196180225, \hspace{0.5cm} -163.36016245424702, \hspace{0.5cm} -1309.7330667771344, \hspace{0.5cm} -2170.4745203525963, \hspace{0.5cm} -1094. 
→9726480270429, 2123.26458714393, 2416.0126942531097, -163.3602301351151, -1309.
→733210696227, −2170.474634870388, −1094.9726875020344, 2123.2647662400955, 2456.
→6075093447926, -77.22909111556126, -1283.252068702874, -2130.2151765650274, -1159.
→8081395939932, 2145.242676822373, -1161.1768654509938, 2145.138477965435, -1280.
→4300435279195, -2113.915974006277, -1160.6073109958113, 2140.403981134188, -1281.
→0373135073385, -2109.180381738326, 2415.769857096313, -163.5069028811886, -1309.
→832637909631, -2170.4632089820057, -1094.7636605550108, 2123.186733078666, 2401.
→861193733008, -172.353118727211, -1318.579879780884, -2171.2963607061065, -1093.
→1900162376958, 2120.0426886068794, 2411.6393197737566, -163.81660155967867, -1089.
→5319085584983, 2121.576761353844, -1082.6282761181412, 2120.6709717842764, 2399.
→6753950529396, -160.25426417938482, 278.27148663424725, -10.493493912035552, -146.
→96678765905787, -241.22429003185408, -130.42325689990824, 244.48803229938719, 206.
→94129996491046, -35.38505415088836, -142.73287289392556, -183.59574244679482, -91.
→6887699449892, 207.03653181237675, 2401.7765548121, -0.0011773940893547467, -1200.
→8892060353498, -2079.9993861623666, -1200.8871497819646, 2080.0002010072703, 278.
\rightarrow 34299926170385, 244.99208111611046]
```

# CHAPTER 4

src

## 4.1 py\_dss\_interface package

#### 4.1.1 ActiveClass

Created by eniocc at 11/10/2020

This interface implements the ActiveClass (IActiveClass) interface of OpenDSS by declaring 3 procedures for accessing the different properties included in this interface: ActiveClassS, ActiveClassI, ActiveClassV. In the original paper Davis cited that are 4 procedures, but only 3 were described.

This interface can be used to read/modify the properties of the ActiveClass Class where the values are integers.

The structure of the interface is as follows: int32\_t ActiveClassI(int32\_t Parameter,int32\_t argument).

This interface returns an integer (signed 32 bits), the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
active\_class\_count() \rightarrow int
```

Gets the number of elements in this class. Same as NumElements Property.

```
\textbf{active\_class\_first} \; () \; \rightarrow int
```

Sets first element in the active class to be the active DSS object. If object is a CktElement, ActiveCktElement also points to this element. Returns 0 if none.

```
active class next() \rightarrow int
```

Sets next element in the active class to be the active DSS object. If object is a CktElement, ActiveCktElement also points to this element. Returns 0 if none.

```
active\_class\_num\_elements() \rightarrow int
```

Gets the number of elements in this class. Same as Count Property.

```
class py_dss_interface.models.ActiveClass.ActiveClass.ActiveClassS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the ActiveClass Class where the values are strings.

The structure of the interface is as follows: CStr ActiveClassI(int32\_t Parameter, CStr argument)

This interface returns a string, the first parameter is used to specify the property of the class to be used and the second parameter can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
active\_class\_get\_class\_name() \rightarrow str
```

Sets the name of the active Element of the Active class.

```
active\_class\_get\_name() \rightarrow str
```

Gets the name of the active Element of the Active class.

```
active\_class\_parent\_class\_name() \rightarrow str
```

Gets the name of the Parent Element of the Active class.

```
active\_class\_write\_name(argument) \rightarrow str
```

Sets the name of the active Element of the Active class.

```
class py_dss_interface.models.ActiveClass.ActiveClass.ActiveClassV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the ActiveClass Class where the values are Variants.

The structure of the interface is as follows: void ActiveClassI(int32\_t Parameter, VARIANT \*Argument)

This interface returns a Variant, the first parameter is used to specify the property of the class to be used and the second parameter can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
active\_class\_all\_names() \rightarrow List[str]
```

Gets a variant array of strings consisting of all element names in the active Class.

#### 4.1.2 Bus

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.Bus.Bus(obj_dss)
Bases: py_dss_interface.models.Bus.BusS.BusS, py_dss_interface.models.Bus.
BusI.BusI, py_dss_interface.models.Bus.BusV.BusV, py_dss_interface.models.
Bus.BusF.BusF
```

This interface implements the Bus (IBus) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: BusS, BusI, BusV, BusF.

```
class py_dss_interface.models.Bus.Bus.BusF(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double BUSF(int32 t Parameter, double Argument)

This interface returns a floating point number (64 bits) according to the number sent in the variable "parameter". The parameter can be one of the following.

```
bus\_distance() \rightarrow float
```

Returns the distance from the energymeter (if non-zero).

#### ${\tt bus\_interruptions\_avg\_duration} \ () \ \to float$

Returns the average interruption duration in hours.

#### $bus\_interruptions\_num() \rightarrow float$

Returns the number of interruptions this bus per year.

#### bus\_interruptions\_total\_customers() → float

Returns the annual number of customer interruptions from this bus.

```
bus_kv_base() \rightarrow float
```

Returns the base voltage at bus in kV.

```
bus_lambda() \rightarrow float
```

Returns the accumulated failure rate downstream from this bus, faults per year.

```
bus\_line\_total\_miles() \rightarrow float
```

Returns the total length of line downline from this bus, in miles. For recloser siting algorithm.

```
bus\_outage\_customer\_accum\_duration() \rightarrow float
```

Returns the accumulated customer outage durations.

```
{\tt bus\_read\_latitude}\,(\,)\,\to float
```

This parameter returns the GIS latitude assigned to the active bus (if any).

```
{\tt bus\_read\_longitude}\,(\,)\,\to float
```

This parameter returns the GIS longitude assigned to the active bus (if any).

```
bus\_read\_x() \rightarrow float
```

Returns the X coordinate for the bus.

```
bus\_read\_y() \rightarrow float
```

Returns the X coordinate for the bus.

```
bus\_write\_latitude(latitude\_param: float) \rightarrow float
```

This parameter sets the GIS latitude to the active bus using the value given at the argument..

```
bus\_write\_longitude (longitude\_param: float) \rightarrow float
```

This parameter sets the GIS longitude to the active bus using the value given at the argument..

```
bus_write_x (param_coordinate) → int
```

Allows to write the X coordinate for the bus. Returns 0. :param param\_coordinate: The X coordinate, if it's None, X will be 0.0

```
bus\_write\_y(param\_coordinate: float) \rightarrow int
```

Allows to write the Y coordinate for the bus. Returns 0. :param param\_coordinate: The Y coordinate, if it's None, Y will be 0.0

```
class py_dss_interface.models.Bus.Bus.BusI(obj_dss)
```

```
Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32 t BUSI(int32 t Parameter, int32 t Argument)

This interface returns an integer according to the number sent in the variable "parameter". The parameter can be one of the following.

```
{\tt bus\_coord\_defined}\,(\,)\,\to int
```

Returns 1 if a coordinate has been defined for this bus; otherwise, it will return 0.

```
bus_get_unique_node_number (start_number: int = 1) \rightarrow int
```

Returns a unique node number at the active bus to avoid node collisions and adds it to the node list for the bus. The start number can be specified in the second parameter.

Parameters start\_number - The first number corresponding the initial bus node number

Returns int

Return type int

```
bus_num_nodes() \rightarrow int
```

Returns the number of nodes of this bus.

```
bus\_section\_id() \rightarrow int
```

Returns the integer ID of the feeder section in which this bus is located.

```
bus\_total\_customers() \rightarrow int
```

Returns returns the total number of customers served down line from this bus.

```
bus\_zsc\_refresh() \rightarrow int
```

Recomputes Zsc for active bus for present circuit configuration. Return 1 if the procedure was successful.

```
class py_dss_interface.models.Bus.Bus.BusS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr BUSS(int32\_t Parameter, CStr Argument)

This interface returns a string according to the number sent in the variable "parameter". The parameter can be one of the following.

```
bus_name() \rightarrow str
```

Returns the name of the active bus.

```
class py_dss_interface.models.Bus.Bus.BusV(obj_dss)
Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void BUSV(int32\_t Parameter, VARIANT \*Argument)

This interface returns a variant according to the number sent in the variable "parameter". The parameter can be one of the following.

```
bus_all_pce_active_bus() → List[str]
```

Returns an array with the names of all PCE connected to the active bus.

```
\textbf{bus\_all\_pde\_active\_bus}\,(\,)\,\to List[str]
```

Returns an array with the names of all PDE connected to the active bus.

```
bus_axc_012_matrix()
```

Variant array of doubles (complex) containing the complete 012 Zsc matrix.

```
bus_cplx_sequence_voltages()
```

Returns the complex double array of sequence voltages (0, 1, 2) at this bus.

```
bus isc()
```

Returns the short circuit current as complex array.

#### bus line list()

This parameter returns a variant array of strings containing the names of the lines connected to the active bus. The names of the lines include the class name 'Line.'

#### bus load list()

This parameter returns a variant array of strings containing the names of the loads connected to the active bus. The names of the lines include the class name 'Load.'.

#### bus nodes() $\rightarrow$ List[int]

Returns an integer array of node numbers defined at the bus in same order as the voltages.

#### bus\_pu\_vll()

Returns a variant array of complex numbers representing L-L voltages in per unit. Returns -1.0 for 1-phase bus. If more than 3 phases, returns only first 3..

#### bus\_pu\_vmag\_angle()

Returns a variant array of doubles containing voltages in per unit and angles in degrees.

#### bus\_pu\_voltages()

Returns the voltages in per unit at bus as complex array.

#### bus\_seq\_voltages()

Returns a complex array of Sequence voltages at this bus.

#### bus\_vll()

For 2 and 3 phase buses, returns a variant array of complex numbers representing L-L voltages in volts. Returns -1.0 for 1-phase bus. If more than 3 phases, returns only first 3.

#### bus\_vmag\_angle()

Returns a variant array of doubles containing voltages in magnitude (VLN), angle (deg).

#### bus\_voc()

Returns the open circuit voltage as complex array.

#### bus\_voltages()

Returns a complex array of voltages at this bus.

#### bus\_ysc\_matrix()

Returns the complex array of Ysc matrix at bus, column by column.

#### bus\_zsc0()

Returns the complex zero-sequence short circuit impedance at bus.

#### bus\_zsc1()

Returns the complex array of Zsc matrix at bus, column by column.

#### bus\_zsc\_matrix()

Returns the complex array of Zsc matrix at bus, column by column.

#### 4.1.3 Capacitors

Created by eniocc at 11/10/2020

This interface implements the Capacitors (ICapacitors) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: CapacitorsF, CapacitorsI, CapacitorsS, CapacitorsV.

```
class py_dss_interface.models.Capacitors.Capacitors.CapacitorsF(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the Capacitors Class where the values are doubles.

The structure of the interface is as follows: double CapacitorsF(int32\_t Parameter, double argument)

This interface returns a floating point number (64 bits), the first parameter is used to specify the property of the class to be used and second parameter can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
capacitors_read_kv() \rightarrow float
```

Gets the bank rating. Use LL for 2 or 3 phases, or actual can rating for 1 phase.

```
capacitors\_read\_kvar() \rightarrow float
```

Gets the total bank kvar, distributed equally among phases and steps.

```
capacitors_write_kv (argument: float)
```

Sets the bank rating. Use LL for 2 or 3 phases, or actual can rating for 1 phase. There is not a explicit return type in the oficial documentation, because of this we choose not put a explicit return too.

```
capacitors_write_kvar (argument: float)
```

Sets the total bank kvar, distributed equally among phases and steps. There is not a explicit return type in the oficial documentation, because of this we choose not put a explicit return too.

```
class py_dss_interface.models.Capacitors.Capacitors.CapacitorsI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the Capacitors Class where the values are integers.

```
The structure of the interface is as follows: int32_t CapacitorsI(int32_t Parameter, int32_t argument)
```

This interface returns an integer (signed 32 bits), the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
capacitors\_add\_step() \rightarrow int
```

Adds one step of the capacitor if available. If successful returns 1.

```
\textbf{capacitors\_available\_steps} \; (\;) \; \rightarrow int
```

Gets the number of steps available in cap bank to be switched ON.

```
\textbf{capacitors\_close}\,(\,)\,\to int
```

Closes all steps, all phases of the capacitor.

```
capacitors count () \rightarrow int
```

Gets the number of capacitor objects in active circuit.

```
capacitors_first() \rightarrow int
```

Sets the first capacitor active. Returns 0 if no more.

```
capacitors_next() \rightarrow int
```

Sets the next capacitor active. Returns 0 if no more.

```
capacitors_open() \rightarrow int
```

Opens all steps, all phases of the capacitor.

```
{\tt capacitors\_read\_is\_delta()} \, \to int
```

Gets 1 if delta connection, otherwise will return 0 for distributing and switching the total kvar.

```
capacitors read num steps() \rightarrow int
```

Gets the number of steps (defaults 1) for distributing and switching the total bank kvar.

```
capacitors\_subtract\_step() \rightarrow int
```

Subtracts one step of the capacitor if available. If no more steps, returns 0.

```
capacitors_write_is_delta(argument: int = 1) \rightarrow int
```

Sets (Argument) 1 if delta connection, otherwise will return 0 for distributing and switching the total kvar.

```
capacitors write num steps (argument: int) \rightarrow int
```

Sets the number of steps (defaults 1) for distributing and switching the total bank kvar.

```
class py_dss_interface.models.Capacitors.Capacitors.CapacitorsS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the Capacitors Class where the values are Strings.

The structure of the interface is as follows: CStr CapacitorsS(int32\_t Parameter, CStr capacitor\_name)

This interface returns a string, the first parameter is used to specify the property of the class to be used and the second parameter can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
\textbf{capacitors\_read\_name} \, (\,) \, \to str
```

Gets the name of the active Capacitor element.

```
capacitors_write_name (capacitor\_name: str) \rightarrow str
```

Sets the name of the Capacitor element to set it active. There is not a explicit return type in the oficial documentation, because of this we choose not put a explicit return too. :param capacitor\_name: the intended name to the capacitor

```
class py_dss_interface.models.Capacitors.Capacitors.CapacitorsV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the Capacitors Class where the values are Variants.

The structure of the interface is as follows: void Capacitors V(int32\_t Parameter, VARIANT \*Argument)

This interface returns a Variant, the first parameter is used to specify the property of the class to be used and the second parameter can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
capacitors_all_names() \rightarrow List[str]
```

Gets a variant array of strings with all Capacitor names in the circuit.

```
\texttt{capacitors\_read\_states} \hspace{0.1cm} \textbf{()} \hspace{0.1cm} \rightarrow List[int]
```

Gets a variant array of integers [0..numsteps-1] indicating the state of each step. If value is -1 and error has occurred.

```
capacitors\_write\_states(dss\_, argument: list) \rightarrow int
```

Sets a variant array of integers [0..numsteps-1] indicating the state of each step. If value is -1 and error has occurred. :param dss\_: an instance of dss object :param argument: list with status of Capacitor states

#### 4.1.4 CapControls

Created by eniocc at 11/10/2020

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double CapControlsF(int32\_t Parameter, double Argument);

This interface returns a floating point number (64 bits) with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
{\tt capcontrols\_read\_ct\_ratio}\,(\,)\,\to float
```

Gets the transducer ratio current to control current.

```
{\tt capcontrols\_read\_dead\_time}\,(\,)\,\to float
```

Gets the time delay [s] after switching off a step. Control may reset before actually switching.

```
capcontrols\_read\_delay() \rightarrow float
```

Gets the time delay [s] to switch on after arming. Control may reset before actually switching.

```
capcontrols read delay off() \rightarrow float
```

Gets the time delay [s] before switching off a step. Control may reset before actually switching.

```
capcontrols\_read\_off\_setting() \rightarrow float
```

Gets the threshold to switch off a step. See Mode for Units.

```
{\tt capcontrols\_read\_on\_setting}\,(\,)\,\to float
```

Gets the threshold to arm or switch on a step. See Mode for Units.

```
capcontrols\_read\_pt\_ratio() \rightarrow float
```

Gets the transducer ratio from primary feeder to control voltage.

```
capcontrols\_read\_vmax() \rightarrow float
```

Gets the Vmax, this reference with VoltOverride, switch off whenever PT voltage exceeds this level.

```
{\tt capcontrols\_read\_vmin}\,(\,)\,\to float
```

Gets the Vmin, this reference with VoltOverride, switch ON whenever PT voltage drops below this level.

```
capcontrols write ct ratio (argument: float) \rightarrow float
```

Sets the transducer ratio current to control current.

```
capcontrols\_write\_dead\_time(argument: float) \rightarrow float
```

Sets the time delay [s] after switching off a step. Control may reset before actually switching..

```
{\tt capcontrols\_write\_delay} \ (\textit{argument: float}) \ \to \text{float}
```

Sets the time delay [s] to switch on after arming. Control may reset before actually switching.

```
capcontrols\_write\_delay\_off(argument: float) \rightarrow float
```

Sets the time delay [s] before switching off a step. Control may reset before actually switching.

```
capcontrols\_write\_off\_setting(argument: float) \rightarrow float
```

Sets the threshold to switch off a step. See Mode for Units.

#### $capcontrols\_write\_on\_setting(argument: float) \rightarrow float$

Sets the threshold to arm or switch on a step. See Mode for Units.

#### capcontrols write pt ratio (argument: float) $\rightarrow$ float

Sets the transducer ratio from primary feeder to control voltage.

#### $capcontrols\_write\_vmax(argument: float) \rightarrow float$

Sets the Vmax, this reference with VoltOverride, switch off whenever PT voltage exceeds this level.

#### capcontrols write vmin (argument: float) $\rightarrow$ float

Sets the Vmin, this reference with VoltOverride, switch ON whenever PT voltage drops below this level.

```
class py_dss_interface.models.CapControls.CapControls.CapControlsI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t CapControlsI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following

```
capcontrols\_count() \rightarrow int
```

Gets the number of CapControls in Active Circuit.

```
\textbf{capcontrols\_first} \; () \; \rightarrow int
```

Sets the first CapControl active. Returns 0 if no more.

```
capcontrols_next() \rightarrow int
```

Sets the next CapControl active. Returns 0 if no more.

```
capcontrols read mode () \rightarrow int
```

Gets the type of automatic controller (see manual for details). CURRENTCONTROL: Result := 0; VOLT-AGECONTROL: Result := 1; VARCONTROL: Result := 2; TIMECONTROL: Result := 3; PFCONTROL: Result := 4; USERCONTROL: Result := 4;

#### $\verb|capcontrols_read_monitored_term|()| \rightarrow int$

Gets the terminal number on the element that PT and CT are connected to.

```
capcontrols\_read\_use\_volt\_override() \rightarrow int
```

Gets if Vmin and Vmax are enabled to override the control Mode. There is not a explicit return type in the oficial documentation, because of this we choose not put a explicit return too." return self.dss\_obj.CapControlsI(6, 0).

```
capcontrols\_write\_mode(argument: int) \rightarrow int
```

Sets the type of automatic controller (see manual for details). 0: elem.CapControlType := CURRENT-CONTROL; 1: elem.CapControlType := VOLTAGECONTROL; 2: elem.CapControlType := KVARCONTROL; 3: elem.CapControlType := TIMECONTROL; 4: elem.CapControlType := PFCONTROL;

```
capcontrols\_write\_monitored\_term (dss, argument: int) \rightarrow int
```

Sets the terminal number on the element that PT and CT are connected to. There is not a explicit return type in the oficial documentation, because of this we choose not put a explicit return too.

```
capcontrols\_write\_use\_volt\_override(dss, argument: int) \rightarrow int
```

Sets if enables Vmin and Vmax to override the control Mode. There is not a explicit return type in the oficial documentation, because of this we choose not put a explicit return too.

```
class py_dss_interface.models.CapControls.CapControls.CapControlsS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr CapControlsS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
\label{eq:capacitor} \textbf{Capacitor} \ () \ \to str Gets the name of the capacitor that is controlled.
```

#### ${\tt capcontrols\_read\_monitored\_obj()} \to str$

Gets the full name of the element that PT and CT are connected to.

```
{\tt capcontrols\_read\_name}\:(\:)\:\to str
```

Gets the name of the active CapControl.

#### capcontrols\_write\_capacitor (argument: str) $\rightarrow$ str

Sets the name of the capacitor that is controlled.

#### $capcontrols\_write\_monitored\_obj(argument: str) \rightarrow str$

Sets the full name of the element that PT and CT are connected to.

```
capcontrols_write_name (argument: str) \rightarrow str
```

Sets a CapControl active by name.

```
class py_dss_interface.models.CapControls.CapControls.CapControls(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void CapControlsV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
capcontrols_all_names() \rightarrow List[str]
```

#### 4.1.5 Circuit

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.Circuit.Circuit.Circuit(obj_dss)
    Bases: py_dss_interface.models.Circuit.CircuitI.CircuitI, py_dss_interface.
    models.Circuit.CircuitS.CircuitS, py_dss_interface.models.Circuit.CircuitF.
    CircuitF, py_dss_interface.models.Circuit.CircuitV.CircuitV
```

This interface implements the Circuit (ICIrcuit) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: CircuitI, CircuitS, CircuitF, CircuitV

```
class py_dss_interface.models.Circuit.Circuit.CircuitF(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

**The structure of the interface is as follows:** double CircuitF(int32\_t Parameter, double Argument1, double Argument2);

This interface returns a floating point number (IEEE754 64 bits) according to the number sent in the variable "parameter". The parameter can be one of the following.

```
circuit\_capacity\_(capacity\_start=0, capacity\_increment=0.1) \rightarrow float
```

Returns the total capacity of the active circuit. Or this parameter it is necessary to specify the start and increment of the capacity in the arguments argument1 and argument2 respectively.

```
circuit_float(first, second, third) \rightarrow float
```

```
class py_dss_interface.models.Circuit.Circuit.CircuitI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

#### The structure of the interface is as follows: int32\_t CircuitI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer according to the number sent in the variable "parameter". The parameter can be one of the following.

#### ${\tt circuit\_end\_of\_time\_step\_update}\:(\:)\:\to int$

Calls end of time step cleanup routine in solutionalgs.pas. Returns 0.

#### $\textbf{circuit\_first\_element} \; () \; \rightarrow int$

Sets the first Element of the active class to be the active Element, as a result, this parameter will deliver the index of the active Element (0 if none).

#### $circuit\_first\_pc\_element() \rightarrow int$

Sets the first PCElement to be the active PCElement, as a result, this parameter will deliver the index of the active PCElement (ideally 1).

#### circuit\_first\_pd\_element() → int

Sets the first PDElement to be the active PDElement, as a result, this parameter will deliver the index of the active PDElement (ideally 1).

```
circuit\_integer (first: int, second: int) \rightarrow int
```

```
circuit_next_element() → int
```

Sets the next Element of the active class to be the active Element, as a result, this parameter will deliver the index of the active Element (0 if none).

```
{\tt circuit\_next\_pc\_element}\:(\:)\:\to int
```

Sets the next PCElement to be the active PCElement, as a result, this parameter will deliver the index of the active PCElement (if there is no more it will return a 0).

#### ${\tt circuit\_next\_pd\_element}\:(\:)\:\to int$

Sets the next PDElement to be the active PDElement, as a result, this parameter will deliver the index of the active PDElement (if there is no more it will return a 0).

#### ${\tt circuit\_num\_buses} \ () \ \to int$

Will deliver the number of buses included in the active circuit.

#### $circuit\_num\_ckt\_elements() \rightarrow int$

Will deliver the number of CktElements included in the active circuit.

#### $circuit\_num\_nodes() \rightarrow int$

Will deliver the number of nodes included in the active circuit.

#### ${\tt circuit\_parent\_pd\_element}\:(\:)\:\to int$

Sets parent PD Element, if any, to be the active circuit element and returns index > 0 if it fails or not applicable.

#### circuit sample() $\rightarrow$ int

Forces all meters and monitors to take a sample, returns 0.

#### ${\tt circuit\_save\_sample}\,(\,)\,\to int$

Forces all meters and monitors to save their sample buffers, returns 0.

#### $circuit\_set\_active\_bus\_i(i:int) \rightarrow int$

Sets active the bus specified by index, which is compatible with the index delivered by AllBusNames, returns 0 it everything ok.

#### circuit\_update\_storage\_t()

Forces all storage classes to update. Typically done after a solution.

```
{\tt class} \ {\tt py\_dss\_interface.models.Circuit.Circuit.CircuitS} \ (obj\_dss)
```

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr CircuitS(int32 t Parameter, CStr Argument);

This interface returns a string according to the number sent in the variable "parameter". The parameter can be one of the following.

```
circuit_disable() \rightarrow str
```

Allows to disable an element of the active circuit, the element must be specified by name. As a result, this parameter will deliver the string ?Ok?.

```
circuit\_enable() \rightarrow str
```

Allows to enable an element of the active circuit, the element must be specified by name. As a result, this parameter will deliver the string ?Ok?.

```
circuit_name() \rightarrow str
```

Returns the name of the active circuit.

```
circuit\_set\_active\_bus (argument: str) \rightarrow str
```

Allows to activate a bus of the active circuit, the bus must be specified by name. As a result, this parameter will deliver a string with the index of the active Bus.

```
circuit\_set\_active\_class(argument: str) \rightarrow str
```

Allows to activate a Class of the active circuit, the Class must be specified by name. As a result, this parameter will deliver a string with the index of the active Class.

```
circuit\_set\_active\_element(argument: str) \rightarrow str
```

Allows to activate an element of the active circuit, the element must be specified by name. As a result, this parameter will deliver a string with the index of the active element.

```
class py_dss_interface.models.Circuit.Circuit.CircuitV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void CircuitV(int32\_t Parameter, VARIANT \*Argument, int32\_t Argument2);

This interface returns a variant according to the number sent in the variable "parameter". The parameter can be one of the following.

```
circuit\_all\_bus\_distances() \rightarrow List[float]
```

Returns distance from each bus to parent EnergyMeter. Corresponds to sequence in AllBusNames. Argument2 must be 0.

```
circuit\_all\_bus\_names() \rightarrow List[str]
```

Returns an array of strings with the names of all the Buses of the active circuit (See AllNodeNames). Argument2 must be 0.

```
circuit_all_bus_vmag() → List[float]
```

Returns an array of doubles (magnitude) with the node voltages from the most recent solution. Argument2 must be 0.

```
circuit_all_bus_vmag_pu() → List[float]
```

Returns an array of doubles with the voltages in per unit of the most recent solution of the active circuit. Argument2 must be 0.

```
{\tt circuit\_all\_bus\_volts}\,(\,)\,\to List[float]
```

Returns an array of doubles (two doubles for representing a complex number) with the node voltages from the most recent solution. Argument2 must be 0.

#### circuit all element losses() → List[float]

Returns an array of doubles (two doubles for representing a complex number) with the losses in each element of the active circuit. Argument2 must be 0.

#### $circuit\_all\_element\_names() \rightarrow List[str]$

Returns an array of strings with the names of all the elements of the active circuit. Argument2 must be 0.

#### circuit all node distances() → List[float]

Returns distance from each Node to parent EnergyMeter. Corresponds to sequence in AllBusVmag. Argument2 must be 0.

#### $circuit_all_node_distances_by_phase (argument: int = 1) \rightarrow List[float]$

Returns array of doubles representing the distances to parent EnergyMeter. Sequence of array corresponds to other node ByPhase properties. Argument2 must contain the number of the phase to return.

#### circuit\_all\_node\_names() → List[str]

Returns an array of strings containing full name of each node in system in same order as returned by AllBusVolts, etc. Argument2 must be 0.

#### $circuit\_all\_node\_names\_by\_phase(argument: int = 1) \rightarrow List[str]$

Returns array of strings of the node names by Phase criteria. Sequence corresponds to other ByPhase properties. Argument2 must contain the number of the phase to return.

#### $circuit_all_node_vmag_by_phase (argument: int = 1) \rightarrow List[float]$

Returns array of doubles representing the voltage magnitudes for nodes on the specified phase. The phase must be specified in the Argument2.

#### $circuit_all_node_vmag_pu_by_phase(argument: int = 1) \rightarrow List[float]$

Returns array of doubles representing the voltage magnitudes (in per unit) for nodes on the specified phase. The phase must be specified in the Argument2.

#### $circuit\_line\_losses() \rightarrow List[float]$

Returns an array of doubles (two doubles for representing a complex number) with the total Line losses of the active circuit. Argument2 must be 0.

#### circuit\_losses() → List[float]

Returns an array of doubles (two doubles for representing a complex number) with the total losses of the active circuit. Argument2 must be 0.

#### circuit\_substation\_losses() → List[float]

Returns an array of doubles (two doubles for representing a complex number) with the total transformer losses of the active circuit. Argument2 must be 0.

#### circuit\_system\_y() → List[float]

Returns an array of doubles (two doubles for representing a complex number) containing the Y Bus Matrix of the system (after a solution has been performed). Argument2 must be 0.

#### $\textbf{circuit\_total\_power}\,(\,)\,\to List[float]$

Returns an array of doubles (two doubles for representing a complex number) with the total power in watts delivered to the active circuit. Argument2 must be 0.

#### circuit\_y\_currents()

Returns a variant array of doubles containing complex injection currents for the present solution. It is the "I" vector of I=YV. Argument2 must be 0.

#### $circuit_y_node_order() \rightarrow List[str]$

Returns a variant array of strings containing the names of the nodes in the same order as the Y Matrix. Argument2 must be 0.

#### circuit y node varray()

Returns a complex array of actual node voltages in same order as SystemY Matrix. Argument2 must be 0.

#### 4.1.6 CktElement

Created by eniocc at 11/10/2020

This interface implements the CktElement interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: CktElementI, CktElementF, CktElementV

```
class py_dss_interface.models.CktElement.CktElement.CktElementF(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

```
cktelement\_read\_emerg\_amps() \rightarrow float
```

Deliver the Emergency ampere rating for the active PDElement.

```
cktelement_read_norm_amps() → float
```

Deliver the normal ampere rating for the active PDElement.

```
cktelement_variable_i (argument: float) → float
```

Delivers get the value of a variable by index for the active PCElement.

```
cktelement\_write\_emerg\_amps(argument: float) \rightarrow float
```

Allows to fix the Emergency ampere rating for the active PDElement. The new value must be defined in the variable ?Argument?.

```
cktelement\_write\_norm\_amps(argument: float) \rightarrow float
```

Allows to fix the normal ampere rating for the active PDElement.

```
class py_dss_interface.models.CktElement.CktElementI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

```
The structure of the interface is as follows: int32_t CktElementI(int32_t Parameter, int32_t Argument);
```

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
cktelement_close (argument: int) \rightarrow int
```

Close the specified terminal (Argument) of the active DSS object.

```
{\tt cktelement\_has\_switch\_control}\;()\;\to int
```

Returns 1 if the active DSS object has a Switch Control linked; otherwise, it will return 0.

```
cktelement_has\_volt\_control() \rightarrow int
```

Returns 1 if the active DSS object has a Volt Control linked; otherwise, it will return 0.

```
cktelement_is_open() \rightarrow int
```

Return a 1 if any terminal of the active DSS object is open, otherwise, it will return a 0.

```
cktelement num conductors () \rightarrow int
```

Deliver the number of conductors of the active DSS object.

```
\textbf{cktelement\_num\_controls}\,(\,)\,\rightarrow int
```

Returns number of controls linked to the active DSS object.

```
cktelement_num\_phases() \rightarrow int
```

Delivers the number of phases of the active DSS object.

#### cktelement num properties () $\rightarrow$ int

Return the number of properties of the active DSS object.

#### ${\tt cktelement\_num\_terminals}\,(\,)\,\to int$

Deliver the number of terminals of the active DSS object.

#### $cktelement\_ocp\_dev\_index() \rightarrow int$

Returns the Index into Controller list of OCP Device controlling the active DSS object.

#### $cktelement\_ocp\_dev\_type() \rightarrow int$

Returns one of the following values: 0=none; 1=Fuse; 2=Recloser; 3=Relay according to the type of active control.

#### **cktelement\_open** ( $argument: int) \rightarrow int$

Open the specified terminal (Argument) of the active DSS object.

#### $\textbf{cktelement\_read\_enabled}\,(\,)\,\rightarrow int$

Returns one of the following values: 0 if the active element is disabled or 1 if the active element is enabled.

```
cktelement\_write\_enabled(argument: int) \rightarrow int
```

Returns one of the following values: 0 if the active element is disabled or 1 if the active element is enabled.

```
class py_dss_interface.models.CktElement.CktElement.CktElementS(obj_dss)
     Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr CktElementS(int32\_t Parameter, CStr Argument);

This interface returns a string (pAnsiChar) with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
cktelement_controller (argument: str) \rightarrow str
```

Delivers the Full name of the i-th controller attached to the active circuit element. The i-th controller index must be specified in the argument arg. Ex: Str = Controller(2). See NumControls to determine valid index range.

#### $\textbf{cktelement\_energymeter} \hspace{0.1cm} \textbf{()} \hspace{0.1cm} \rightarrow str$

Delivers the name of the EnergyMeter linked to the active circuit element.

#### $cktelement\_guid() \rightarrow str$

Delivers the unique name for the active circuit element.

```
{\tt cktelement\_name} () \to {\tt str}
```

Delivers the full name of the active circuit element.

```
{\tt cktelement\_read\_display}\,()\,\to str
```

Displays the name of the active circuit element (not necessarily unique).

```
cktelement_write_display (argument: str) → str
```

Allows to modify the name of the active circuit element (not necessarily unique).

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void CktElementV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a variant (the format depends on the parameter) with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
cktelement\_all\_property\_names() \rightarrow List[str]
```

Delivers an array of strings with the names of all the properties of the active circuit element.

#### cktelement all variables names() → List[str]

Delivers a Variant array of strings listing all the published state variable names, if the active circuit element is a PCElement. Otherwise, null string.

#### $cktelement\_all\_variables\_values() \rightarrow List[float]$

Delivers a Variant array of doubles listing all the values of the state variables, if the active circuit element is a PCElement. Otherwise, null string.

#### **cktelement\_cplx\_seq\_currents**() → List[float]

Delivers an array of doubles with the complex of sequence currents for all terminals of the active circuit element.

#### cktelement\_cplx\_seq\_voltages() → List[float]

Delivers an array of doubles with the complex of sequence voltages for all terminals of the active circuit element.

#### **cktelement\_currents**() → List[float]

Delivers an array of doubles with the currents at terminals of the active circuit element.

#### **cktelement\_currents\_mag\_ang()** → List[float]

Delivers the currents in magnitude, angle format as a variant array of doubles of the active circuit element.

#### $cktelement\_losses() \rightarrow List[float]$

Delivers an array of doubles with the Losses at terminals of the active circuit element.

#### $\textbf{cktelement\_node\_order} () \rightarrow List[int]$

Delivers a Variant array integers variant array of integer containing the node numbers (representing phases, for Example) for each conductor of each terminal.

#### $cktelement\_phase\_losses() \rightarrow List[float]$

Delivers an array of doubles with the Losses per phase at the terminals of the active circuit element.

#### $cktelement\_powers() \rightarrow List[float]$

Delivers an array of doubles with the powers at terminals of the active circuit element.

#### $cktelement\_read\_bus\_names() \rightarrow List[str]$

Delivers an array of strings with the names of all the buses connected to the active circuit element.

#### **cktelement\_residuals**() → List[float]

Delivers an array of doubles with the residual currents (magnitude, angle) in all the nodes of the active circuit element.

#### **cktelement** seg currents() → List[float]

Delivers an array of doubles with the symmetrical component Currents per phase at the terminals of the active circuit element.

#### **cktelement** seq powers() → List[float]

Delivers an array of doubles with the symmetrical component powers per phase at the terminals of the active circuit element.

#### $\textbf{cktelement\_seq\_voltages} \; () \; \rightarrow List[float]$

Delivers an array of doubles with the symmetrical component voltages per phase at the terminals of the active circuit element.

#### **cktelement\_voltages**() → List[float]

Delivers an array of doubles with the voltages at terminals of the active circuit element.

#### **cktelement\_voltages\_mag\_ang()** → List[float]

Delivers the voltages in magnitude, angle format as a variant array of doubles of the active circuit element.

#### **cktelement write bus names** (dss, argument: List[str]) $\rightarrow$ str

Allows to fix an array of strings with the names of all the buses connected to the active circuit element.

#### **cktelement\_y\_prim**() → List[float]

Delivers an array of doubles with the Y primitive matrix (complex) of the active circuit element.

#### 4.1.7 CMathLib

Created by eniocc at 11/10/2020

This interface implements the CmathLib (ICmathLib) interface of OpenDSS by declaring 2 procedures for accessing the different properties included in this interface: CMathLibF, CMathLibV.

```
class py_dss_interface.models.CMathLib.CMathLib.CMathLibF (obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

**The structure of the interface is as follows:** double CmathLibF(int32\_t Parameter, double Argument1, double Argument2);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
cmathlib_cabs (arg_real: float, arg_imaginary: float) → float
Returns the absolute value of complex number given in real (Argument1) and imaginary (Argument2) doubles.
```

cmathlib\_cdang (arg\_real: float, arg\_imaginary: float) → float
Returns the angle, in degrees, of a complex number specified as two doubles: Real part (Argument1) and imaginary part (Argument2).

```
class py_dss_interface.models.CMathLib.CMathLib.CMathLibV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

**The structure of the interface is as follows:** void CmathLibV(int32\_t Parameter, double Argument1, Argument2, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
cmathlib_cmplx (real\_part: float, imag\_part: float) \rightarrow complex Convert real (Argument1) and imaginary (Argument1) doubles to variant array of doubles.
```

```
cmathlib_ctopolardeg (double_real: float, double_imag: float) → Tuple[float, float]
Convert complex number (Argument1 and Argument2) to magnitude and angle, degrees. Returns variant array of two doubles.
```

```
cmathlib_pdegtocomplex (double_real: float, double_imag: float) → complex Convert magnitude, angle in degrees (Argument1 and Argument2) to a complex number. Returns variant array of two doubles.
```

#### 4.1.8 CtrlQueue

Created by eniocc at 11/10/2020

This interface implements the CtrlQueue (ICtrlQueue) interface of OpenDSS by declaring 2 procedures for accessing the different properties included in this interface: CtrlQueueI, CtrlQueueV

```
class py_dss_interface.models.CtrlQueue.CtrlQueue.CtrlQueueI(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the CtrlQueue Class where the values are integers.

```
The structure of the interface is as follows: int32_t CtrlQueueI(int32_t Parameter, int32_t argument)
```

This interface returns an integer (signed 32 bits), the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
ctrlqueue\_action(argument: int) \rightarrow int
```

Sets the active action by index (argument).

```
ctrlqueue\_action\_code() \rightarrow int
```

Gets the code for the active action. Long integer code to tell the control device what to do.

```
{\tt ctrlqueue\_clear\_actions}\,(\,)\,\to int
```

Clears the action list.

```
ctrlqueue\_clear\_queue() \rightarrow int
```

Clears the control queue.

```
ctrlqueue\_delete(argument: int) \rightarrow int
```

Deletes a control action from the DSS control queue by referencing the handle of the action (Argument).

```
\textbf{ctrlqueue\_device\_handle} \, (\,) \, \to int
```

Gets the handle (user defined) to device that must act on the pending action.

```
{\tt ctrlqueue\_do\_all\_queue}\,(\,)\,\to int
```

Forces the execution of all control actions stored at the control queue. Returns 0.

```
ctrlqueue\_num\_actions() \rightarrow int
```

Gets the number of actions on the current action list (that have been popped off the control queue by CheckControlActions).

```
\textbf{ctrlqueue\_pop\_action} \; () \; \rightarrow int
```

Pops next action off the action list and makes it the active action. Returns zero if none.

```
\mathtt{ctrlqueue\_push}() \rightarrow \mathtt{int}
```

Pushes a control action onto the DSS control queue by time, action code, and device handle. Returns Control Queue handle.

```
\textbf{ctrlqueue\_queue\_size} \hspace{0.1cm} \textbf{()} \hspace{0.1cm} \rightarrow int
```

Delivers the size of the current control queue. Returns zero if none.

```
ctrlqueue\_show() \rightarrow int
```

Shows the entire control queue in CSV format.

```
class py_dss_interface.models.CtrlQueue.CtrlQueue.CtrlQueueV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the CtrlQueue Class where the values are Variants.

The structure of the interface is as follows: void CtrlQueueV(int32 t Parameter, , VARIANT \*Argument);

This interface returns a Variant, the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
ctrlqueue () \rightarrow str
```

Delivers the control actions contained in the CtrlQueue after the latest solve command.

#### 4.1.9 DSSElement

Created by eniocc at 11/10/2020

This interface implements the DSSElement (IDSSElement) interface of OpenDSS by declaring 3 procedures for accessing the different properties included in this interface: DSSElementI, DSSElementS, DSSElementV

```
class py_dss_interface.models.DSSElement.DSSElement.DSSElementI(obj_dss)
     Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the DSSElement Class where the values are integers.

```
The structure of the interface is as follows: int32_t DSSElementI(int32_t Parameter, int32_t argument);
```

This interface returns an integer (signed 32 bits), the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
dsselement\_num\_properties() \rightarrow int
```

Gets the number of properties for the active DSS object.

This interface can be used to read/modify the properties of the DSSElement Class where the values are Strings.

```
The structure of the interface is as follows: CStr DSSElementS(int32 t Parameter, CStr argument);
```

This interface returns a string, the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
dsselement name() \rightarrow str
```

Gets the full name of the active DSS object (general element or circuit element).

This interface can be used to read/modify the properties of the DSSElement Class where the values are Variants.

```
The structure of the interface is as follows: void DSSElementV(int32_t Parameter VARIANT *Argument);
```

This interface returns a Variant, the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
dsselement\_all\_property\_names() \rightarrow List[str]
```

Gets a variant array of strings containing the names of all properties for the active DSS object.

#### 4.1.10 DSSExecutive

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.DSSExecutive.DSSExecutive.DSSExecutive(obj_dss)
    Bases: py_dss_interface.models.DSSExecutive.DSSExecutiveS.DSSExecutiveS,
    py_dss_interface.models.DSSExecutive.DSSExecutiveI.DSSExecutiveI
```

This interface implements the DSS\_Executive (IDSS\_Executive) interface of OpenDSS by declaring 2 procedures for accessing the different properties included in this interface: DSSExecutiveS, DSSExecutiveI

```
class py_dss_interface.models.DSSExecutive.DSSExecutive.DSSExecutiveI(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

```
The structure of the interface is as follows: int32_t DSSExecutiveI(int32_t Parameter, int32_t Argument);
```

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
\textbf{executive\_num\_commands} \; (\;) \; \rightarrow int
```

Gets the number of DSS Executive Commands.

```
executive_num_options() \rightarrow int
```

Gets the number of DSS Executive Options.

```
class py_dss_interface.models.DSSExecutive.DSSExecutive.DSSExecutiveS(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr DSSExecutiveS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
executive_command (arg: str) \rightarrow str
```

Gets i-th command (specified in the argument as string).

```
executive\_command\_help(arg: str) \rightarrow str
```

Gets help string for i-th command (specified in the argument as string).

```
executive_option (arg: str) \rightarrow str
```

Gets i-th option (specified in the argument as string).

```
executive_option_help (arg: str) \rightarrow str
```

Gets help string for i-th option (specified in the argument as string).

```
executive_option_value (arg: str) \rightarrow str
```

Gets present value for i-th option (specified in the argument as string).

#### 4.1.11 DSSInterface

Created by eniocc at 11/10/2020

This interface implements the DSS interface (IDSS - DDSS.pas) of OpenDSS by declaring 3 procedures for accessing the different properties included in this interface: DSSInterfaceS, DSSInterfaceI, DSSInterfaceV

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t DSSI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

#### dss\_clear\_all()

Clears all circuit definitions.

#### dss num circuits() $\rightarrow$ int

Gets the number of circuits currently defined.

#### ${\tt dss\_num\_classes}\,(\,)\,\to int$

Gets the number of DSS intrinsic classes.

#### dss num user classes() $\rightarrow$ int

Gets the number of user-defined classes.

#### dss read allow forms()

Gets if the DSS allows forms (1) or not (0), default (1).

#### dss\_reset()

Resets DSS initialization for restarts, etc. from applets. Nothing implemented in the OpenDSS Original Source Code

#### dss\_show\_panel()

Shows non-MDI child form of the Main DSS Edit form.

```
dss\_start() \rightarrow int
```

Validates the user and starts the DSS. Returns TRUE (1) if successful.

```
dss write allow forms(argument: int)
```

Sets if the DSS allows forms (1) or not (0), default (1). PAY ATTENTION: If arg=0 Then NoFormsAllowed := TRUE (Only set to False) else NoFormsAllowed := FALSE;

```
class py_dss_interface.models.DSSInterface.DSSInterface.DSSInterfaceS(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr DSSS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
dss default editor() \rightarrow str
```

Gets the path name for the default text editor.

```
dss_new_circuit (argument: str) \rightarrow str
```

Makes a new circuit, the name of the circuit must be specified in the Argument.

```
dss read datapath() \rightarrow str
           Gets the Data File Path. Default for reports, etc. from DSS.
     {\tt dss\_version}\,() \to {\rm str}
           Gets the version string for the DSS.
     dss write datapath (argument: str) \rightarrow str
           Sets the Data File Path. Default for reports, etc. from DSS.
class py dss interface.models.DSSInterface.DSSInterface.DSSInterfaceV(obj dss)
     Bases: py_dss_interface.models.Base.Base
     This interface can be used to read/write certain properties of the active DSS object.
     The structure of the interface is as follows: void DSSV(int32 t Parameter, VARIANT *Argument);
     This interface returns a Variant with the result of the query according to the value of the variable Parameter,
     which can be one of the following.
     \texttt{dss\_classes}\,(\,)\,\to List[str]
```

Gets the list of DSS intrinsic classes (names of the classes).

```
dss\_user\_classes() \rightarrow List[str]
```

Gets list of user-defined classes (names of the classes).

#### 4.1.12 DSSProgress

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.DSSProgress.DSSProgress.DSSProgress(obj_dss)
                 py_dss_interface.models.DSSProgress.DSSProgressI.DSSProgressI,
    py_dss_interface.models.DSSProgress.DSSProgressS.DSSProgressS
```

This interface implements the DSSProgress (IDSSProgress) interface of OpenDSS by declaring 2 procedures for accessing the different properties included in this interface: DSSProgressI, DSSProgressS

```
class py_dss_interface.models.DSSProgress.DSSProgress.DSSProgressI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32 t DSSProgressI(int32 t Parameter, int32 t Argument);

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
dssprogress\_close() \rightarrow int
     Closes (hides) DSS Progress form.
dssprogress\_pct\_progress(arg: float) \rightarrow int
     Sets the percent progress to indicate [0..100].
```

```
dssprogress\_show() \rightarrow int
```

Shows progress form with null caption and progress set to zero.

```
class py_dss_interface.models.DSSProgress.DSSProgress.DSSProgressS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

TThis interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr DSSProgressS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

#### **dssprogress** caption (arg: str) $\rightarrow str$

Sets the caption to appear on the bottom of the DSS Progress form.

#### 4.1.13 DSSProperties

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.DSSProperties.DSSProperties.DSSProperties(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface implements the DSSproperties (IDSSProperties) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface.

This interface can be used to read/write certain properties of DSS objects.

The structure of the interface is as follows: CStr DSSProperties(int32\_t Parameter, CStr Argument);

This interface returns a string pointer (ANSI) with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
dssproperties\_description (argument: str) \rightarrow str
```

This parameter will deliver the description of the active property. This parameter will deliver the name of the active property. The index of the property must be specified in the argument. The index minimum value is 1. This value must be entered as string.

```
dssproperties\_name(argument: str) \rightarrow str
```

Delivers the name of the active property. The index of the property must be specified in the argument. The index minimum value is 1. This value must be entered as string.

```
dssproperties\_read\_value(argument: str) \rightarrow str
```

This parameter will deliver the value of the active property. This parameter will deliver the name of the active property. The index of the property must be specified in the argument. The index minimum value is 1. This value must be entered as string.

```
dssproperties\_write\_value(argument: str) \rightarrow str
```

This parameter will allow to set the value of the active property. The new value must be specified in the variable "argument" as string. This parameter will deliver the name of the active property. The index of the property must be specified in the argument. The index minimum value is 1. This value must be entered as string.

#### 4.1.14 ErrorInterface

Created by eniocc at 11/05/2021

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t ErrorCode(void);

```
error\_code() \rightarrow int
```

"This interface returns an integer with latest error code delivered by OpenDSS.

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr ErrorDesc(void );

This interface returns a string with description of the latest error code delivered by OpenDSS.

```
error_desc() \rightarrow str
```

"This interface returns a string with description of the latest error code delivered by OpenDSS.

#### 4.1.15 Fuses

Created by eniocc at 11/10/2020

"This interface implements the Fuses (IFuses) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: FusesI, FusesS, FusesF, FusesV

```
class py_dss_interface.models.Fuses.Fuses.Fusesf(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double FusesF(int32 t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
fuses read delay() \rightarrow float
```

Gets the fixed delay time in seconds added to the fuse blowing time determined by the TCC curve. Default is 0.

```
fuses_read_rated_current() \rightarrow float
```

Gets the multiplier or actual amps for the TCCcurve object. Defaults to 1.0, Multiply current values of TCC curve by this to get actual amps.

```
fuses_write_delay (argument: float) \rightarrow float
```

Sets the fixed delay time in seconds added to the fuse blowing time determined by the TCC curve. Default is 0.

```
fuses\_write\_rated\_current(argument: float) \rightarrow float
```

Sets the multiplier or actual amps for the TCCcurve object. Defaults to 1.0, Multiply current values of TCC curve by this to get actual amps.

```
class py_dss_interface.models.Fuses.Fuses.FusesI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t FusesI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
fuses_close() \rightarrow int Closing of fuse.

fuses count() \rightarrow int
```

Returns the number of Fuses objects currently defined in the active circuit.

```
fuses first() \rightarrow int
```

Sets the first Fuse to be the active Fuse. Returns 0 if none.

#### fuses is blown() $\rightarrow$ int

Returns the current state of the fuses. TRUE (1) if any on any phase is blown. Else FALSE (0).

#### fuses\_next() $\rightarrow$ int

Sets the next Fuse to be the active Fuse. Returns 0 if none.

#### ${\tt fuses\_num\_phases}\,(\,)\,\to int$

Gets the number of phases of the active fuse.

#### fuses\_open() $\rightarrow$ int

Opening of fuse.

#### $fuses\_read\_idx() \rightarrow int$

Gets the active fuse by index into the list of fuses. 1 based: 1..count.

#### fuses\_read\_monitored\_term() $\rightarrow$ int

Gets the terminal number to switch the fuse is connected.

#### fuses\_read\_switched\_term() → int

Gets the terminal number of the terminal containing the switch controlled by the fuse.

#### fuses reset()

Resets the state of the fuse object to the normal state.

#### **fuses\_write\_idx** ( $argument: int) \rightarrow int$

Sets the active fuse by index into the list of fuses. 1 based: 1..count.

#### fuses\_write\_monitored\_term (argument: int) $\rightarrow$ int

Sets the terminal number to switch the fuse is connected.

#### fuses\_write\_switched\_term (argument: int) $\rightarrow$ int

Sets the terminal number of the terminal containing the switch controlled by the fuse.

```
class py_dss_interface.models.Fuses.Fuses.Fuses(obj_dss)
```

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

#### The structure of the interface is as follows: CStr FusesS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

#### ${\tt fuses\_read\_monitored\_obj()} \to {\rm str}$

Gets the name of the Monitored Object by the active fuse.

#### fuses read name() $\rightarrow$ str

Gets the name of the active fuse.

#### fuses\_read\_switched\_obj() $\rightarrow$ str

Gets the full name of the circuit element switch that the fuse controls. Defaults to the MonitoredObj.

#### $\textbf{fuses\_read\_tcc\_curve} \, (\,) \, \to str$

Gets the name of the TCCcurve object that determines fuse blowing.

#### **fuses\_write\_monitored\_obj** (argument: str) → str

Sets the name of the Monitored Object by the active fuse.

#### **fuses\_write\_name** (argument: str) $\rightarrow$ str

Sets the name of the active fuse.

#### $fuses\_write\_switched\_obj(argument) \rightarrow str$

Sets the full name of the circuit element switch that the fuse controls. Defaults to the MonitoredObj.

```
fuses write tcc curve (argument: str) \rightarrow str
```

Sets the name of the TCCcurve object that determines fuse blowing.

```
class py_dss_interface.models.Fuses.Fuses.Fuses(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void Fuses V(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
fuses_all_names() \rightarrow List[str]
```

Gets the variant array of string containing names of all fuses in the circuit.

#### fuses\_read\_normal()

Gets a variant array of strings[0..Nphases-1] indicating the normal state for all phases of the active fuse. If value is -1 an error has occurred.

#### fuses\_read\_state()

Gets a variant array of strings[0..Nphases-1] indicating the present state for all phases of the active fuse. If value is -1 an error has occurred.

```
fuses_write_normal(argument)
```

Sets a variant array of strings [0..Nphases-1] indicating the state for all phases of the active fuse. If value is -1 an error has occurred.

```
fuses_write_state(argument)
```

Sets a variant array of strings [0..Nphases-1] indicating the state for all phases of the active fuse. If value is -1 an error has occurred.

#### 4.1.16 Generators

Created by eniocc at 11/10/2020

This interface implements the Generators (IGenerators) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: GeneratorsI, GeneratorsF, GeneratorsV.

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double GeneratorsF(int32\_t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
generators\_read\_kv() \rightarrow float
```

Gets the voltage base for the active generator, kV.

#### $generators\_read\_kva\_rated() \rightarrow float$

Gets the KVA rating of the generator.

#### generators read kvar() $\rightarrow$ float

Gets the kvar output for the active generator, kW is updated for current power factor.

## $generators\_read\_kw() \rightarrow float$

Gets the kW output for the active generator, kvar is updated for current power factor.

## $generators\_read\_pf() \rightarrow float$

Gets the power factor (pos. = producing vars). Updates kvar based on present kW value.

#### $generators\_read\_vmax\_pu() \rightarrow float$

Gets the Vmaxpu for Generator Model.

## $generators\_read\_vmin\_pu() \rightarrow float$

Gets the Vminpu for Generator Model.

### generators\_write\_kv (argument: float) $\rightarrow$ float

Sets the voltage base for the active generator, kV.

## $generators\_write\_kva\_rated(argument: float) \rightarrow float$

Sets the KVA rating of the generator.

#### generators write kvar (argument: float) $\rightarrow$ float

Sets the kvar output for the active generator, kW is updated for current power factor.

## $generators\_write\_kw(argument: float) \rightarrow float$

Sets the kW output for the active generator, kvar is updated for current power factor.

#### generators\_write\_pf (argument: float) $\rightarrow$ float

Sets the power factor (pos. = producing vars). Updates kvar based on present kW value.

# $\verb"generators_write_vmax_pu" (\textit{argument: float}) \rightarrow \verb"float"$

Sets the Vmaxpu for Generator Model.

#### $generators\_write\_vmin\_pu$ (argument: float) $\rightarrow$ float

Sets the Vminpu for Generator Model.

```
class py_dss_interface.models.Generators.Generators.GeneratorsI(obj_dss)
```

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t GeneratorsI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

## $generators\_count() \rightarrow int$

Returns the number of generators Objects in Active Circuit.

#### generators\_first() $\rightarrow$ int

Sets first generator to be active. Returns 0 if None.

## $generators\_next() \rightarrow int$

Sets next generator to be active. Returns 0 if None.

## generators\_read\_forced\_on() $\rightarrow$ int

Returns 1 if the generator is forced ON regardless of other dispatch criteria; otherwise, returns 0.

## $generators\_read\_idx() \rightarrow int$

Gets the active generator by Index into generators list. 1..Count.

#### $generators\_read\_model() \rightarrow int$

Gets the active generator Model (see Manual for details). 1:Generator injects a constant kW at specified power factor. 2:Generator is modeled as a constant admittance. 3:Const kW, constant kV. Somewhat like a

conventional transmission power flow P-V generator. 4:Const kW, Fixed Q (Q never varies) 5:Const kW, Fixed Q(as a constant reactance) 6:Compute load injection from User-written Model.(see usage of Xd, Xdp) 7:Constant kW, kvar, but current-limited below Vminpu. Approximates a simple inverter. See also Balanced.

```
generators\_read\_phases() \rightarrow int
```

Returns the number of phases of the active generator.

```
generators\_write\_forced\_on(argument: int) \rightarrow int
```

Allows to force ON regardless of other dispatch criteria. To force ON put 1 in the argument, otherwise put 0.

```
generators_write_idx (argument: int) \rightarrow int
```

Sets the active generator (argument) by Index into generators list. 1..Count.

```
generators\_write\_model(argument: int) \rightarrow int
```

Sets the active generator Model (see Manual for details). 1:Generator injects a constant kW at specified power factor. 2:Generator is modeled as a constant admittance. 3:Const kW, constant kV. Somewhat like a conventional transmission power flow P-V generator. 4:Const kW, Fixed Q (Q never varies) 5:Const kW, Fixed Q(as a constant reactance) 6:Compute load injection from User-written Model.(see usage of Xd, Xdp) 7:Constant kW, kvar, but current-limited below Vminpu. Approximates a simple inverter. See also Balanced.

```
generators_write_phases (argument: int) \rightarrow int
```

Sets the number of phases (argument) of the active generator.

```
class py_dss_interface.models.Generators.Generators.GeneratorsS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr GeneratorsS(int32\_t Parameter, CStr Argument);

This interface returns a string as a result of the query according to the value of the variable Parameter, which can be one of the following.

```
{\tt generators\_read\_name}\:(\:)\:\to str
```

Gets the name of the active Generator.

```
generators_write_name (argument: str) \rightarrow str
```

Sets the name of the active Generator.

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void Generators V(int32 t Parameter, VARIANT \*Argument);

This interface returns a Variant as a result of the query according to the value of the variable Parameter, which can be one of the following.

```
generators_all_names() \rightarrow List[str]
```

Gets the array of names of all Generator objects.

```
generators\_register\_names() \rightarrow List[str]
```

Gets the array of names of all generator Energy Meter registers.

```
generators\_register\_values() \rightarrow List[float]
```

Gets the array of values in generator Energy Meter registers.

# 4.1.17 ISources

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.ISources.ISources.ISources(obj_dss)
    Bases:
                          py_dss_interface.models.ISources.ISourcesI.ISourcesI,
    py dss interface.models.ISources.ISourcesF.ISourcesF,
                                                               py_dss_interface.
    models.ISources.ISourcesS.ISourcesS,
                                              py_dss_interface.models.ISources.
    ISourcesV. ISourcesV
```

This interface implements the ISources (IIsources) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: ISourcesI, ISourcesF, ISourcesV.

```
class py dss interface.models.ISources.ISources.ISourcesF(obj dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double IsourcesF(int32\_t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
isources\_read\_amps() \rightarrow float
     Gets the magnitude of the Isource in Amps.
```

```
isources\_read\_angle\_deg() \rightarrow float
```

Gets the phase angle of the Isource in degrees.

```
isources_read_frequency() → float
    Gets the frequency of the Isource in Hz.
```

```
isources\_write\_amps(argument: float) \rightarrow float
     Sets the magnitude of the Isource in Amps.
```

```
\verb|isources_write_angle_deg| (argument: float)| \rightarrow \verb|float||
      Sets the phase angle of the Isource in degrees.
```

```
isources\_write\_frequency(argument: float) \rightarrow float
```

Sets the frequency of the Isource in Hz.

```
class py_dss_interface.models.ISources.ISources.ISourcesI(obj_dss)
    Bases: py dss interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t IsourcesI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
isources\_count() \rightarrow int
```

Returns the number of Isource objects currently defined in the active circuit.

```
isources first () \rightarrow int
```

Sets the first ISource to be active; returns 0 if none.

```
isources next() \rightarrow int
```

Sets the next ISource to be active; returns 0 if none.

```
class py_dss_interface.models.ISources.ISources.ISourcesS(obj_dss)
    Bases: py dss interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr IsourcesS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
isources read name() \rightarrow str
```

Gets the name of the active Isource object.

```
\verb"isources_write_name" (argument: str) \rightarrow \operatorname{str}
```

Sets the name of the active Isource object.

```
class py_dss_interface.models.ISources.ISources.ISourcesV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void Isources V(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
isources\_all\_names() \rightarrow List[str]
```

Gets the variant array of string containing names of all ISources in the circuit.

## 4.1.18 LineCodes

Created by eniocc at 11/10/2020

This interface implements the Lines (ILineCodes) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface:

```
class py_dss_interface.models.LineCodes.LineCodes.LineCodesF(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the LineCode Class where the values are doubles.

The structure of the interface is as follows: double LineCodesF(int32 t Parameter, double argument)

This interface returns a floating point number, the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
linecodes read c0() \rightarrow float
```

Gets the Zero-sequence capacitance in ohms per unit length for the active LineCode.

```
{\tt linecodes\_read\_c1} \ () \ \to float
```

Gets the Positive-sequence capacitance in nF per unit length for the active LineCode.

```
{\tt linecodes\_read\_emerg\_amps} \; (\;) \; \to float
```

Gets the Emergency ampere rating for the active LineCode.

```
linecodes\_read\_norm\_amps() \rightarrow float
```

Gets the normal ampere rating for the active LineCode.

#### linecodes read $r0() \rightarrow float$

Gets the Zero-sequence resistance in ohms per unit length for the active LineCode.

#### linecodes read r1() $\rightarrow$ float

Gets the Positive-sequence resistance in ohms per unit length for the active LineCode.

#### linecodes\_read\_x0() $\rightarrow$ float

Gets the Zero-sequence reactance in ohms per unit length for the active LineCode.

#### linecodes read $x1() \rightarrow float$

Gets the Positive-sequence reactance in ohms per unit length for the active LineCode.

# ${\tt linecodes\_write\_c0} \ (\textit{argument: float}) \ \to {\tt float}$

Sets the Zero-sequence capacitance in ohms per unit length for the active LineCode. This value must be specified in the argument as a double.

#### linecodes\_write\_c1 (argument: float) $\rightarrow$ float

Sets the Positive-sequence capacitance in nF per unit length for the active LineCode. This value must be specified in the argument as a double.

## $linecodes\_write\_emerg\_amps(argument: float) \rightarrow float$

Sets the Emergency ampere rating for the active LineCode. This value must be specified in the argument as a double.

## $linecodes\_write\_norm\_amps(argument: float) \rightarrow float$

Sets the normal ampere rating for the active LineCode. This value must be specified in the argument as a double.

## $linecodes\_write\_r0 (argument: float) \rightarrow float$

Sets the Zero-sequence resistance in ohms per unit length for the active LineCode. This value must be specified in the argument as a double.

# $linecodes\_write\_r1 (argument: float) \rightarrow float$

Sets the Positive-sequence resistance in ohms per unit length for the active LineCode. This value must be specified in the argument as a double.

# $linecodes\_write\_x0$ (argument: float) $\rightarrow$ float

Sets the Zero-sequence reactance in ohms per unit length for the active LineCode. This value must be specified in the argument as a double.

# $linecodes\_write\_x1 (argument: float) \rightarrow float$

Sets the Positive-sequence reactance in ohms per unit length for the active LineCode. This value must be specified in the argument as a double.

```
\verb|class| py_dss_interface.models.LineCodes.LineCodes.LineCodes| (|obj_dss|)
```

Bases: py\_dss\_interface.models.Base.Base

This interface implements the Lines (ILineCodes) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface:

# $linecodes\_count() \rightarrow int$

Gets the number of Line Objects in Active Circuit.

## $linecodes\_first() \rightarrow int$

Sets the first element active. Returns 0 if no lines. Otherwise, index of the line element.

#### linecodes\_is\_z1z0() $\rightarrow$ int

Gets the flag (Boolean 1/0) denoting whether the impedance data were entered in symmetrical components.

## ${\tt linecodes\_next}\,(\,)\,\to int$

Sets the next element active. Returns 0 if no lines. Otherwise, index of the line element.

#### linecodes read phases () $\rightarrow$ int

Delivers the number of phases of the active LineCode as an integer.

#### linecodes read units() $\rightarrow$ int

Delivers the units of the active LineCode as an integer.

```
linecodes\_write\_phases(argument) \rightarrow int
```

Sets the number of phases of the active LineCode. The units must be specified as an integer in the argument.

```
linecodes write units (argument: int) \rightarrow int
```

Sets the units of the active LineCode. The units must be specified as an integer in the argument. Please refer to the OpenDSS User manual for more information. UNITS\_MAXNUM =9; UNITS\_NONE =0; UNITS\_MILES =1; UNITS\_KFT =2; UNITS\_KM =3; UNITS\_M =4; UNITS\_FT =5; UNITS\_IN =6; UNITS\_CM =7; UNITS\_MM =8;

```
class py_dss_interface.models.LineCodes.LineCodes.LineCodesS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the LineCode Class where the values are Strings.

The structure of the interface is as follows: CStr LineCodesS(int32 t Parameter, CStr argument)

This interface returns a string, the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.;

```
linecodes read name() \rightarrow str
```

Gets the name of the active LineCode element.

```
linecodes_write_name (argument: str) \rightarrow str
```

Sets the name of the active LineCode element. The new value must be specified in the argument as a string.

```
class py_dss_interface.models.LineCodes.LineCodes.LineCodesV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the LineCode Class where the values are Variants.

The structure of the interface is as follows: void LineCodes V(int32\_t Parameter, , VARIANT \*Argument);

This interface returns a Variant, the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
{\tt linecodes\_all\_names}\,(\,)\,\to str
```

Gets the capacitance matrix in ohms per unit length of the active LineCode.

```
linecodes\_read\_cmatrix() \rightarrow str
```

Gets the capacitance matrix in ohms per unit length of the active LineCode.

```
\textbf{linecodes\_read\_rmatrix}\,(\,)\,\to str
```

Gets the resistance matrix in ohms per unit length of the active LineCode.

```
linecodes\_read\_xmatrix() \rightarrow str
```

Gets the reactance matrix in ohms per unit length of the active LineCode.

```
linecodes_write_cmatrix(argument: str) → None
```

Sets the capacitance matrix in ohms per unit length of the active LineCode. The new values must be entered as a vector of doubles using the argument. :param argument: must be a string like that [383.948 lo 383.948 lo 0 383.948 ]

## linecodes write rmatrix (argument: str) → None

Sets the resistance matrix in ohms per unit length of the active LineCode. The new values must be entered as a vector of doubles using the argument. :param argument: must be a string like that [0.791721 | 0.318476 0.781649 | 0.28345, 0.318476, 0.791721]

## $linecodes\_write\_xmatrix (argument: str) \rightarrow None$

Sets the reactance matrix in ohms per unit length of the active LineCode. The new values must be entered as a vector of doubles using the argument.

# 4.1.19 Lines

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.Lines.Lines.Lines(obj_dss)
```

```
Bases: py_dss_interface.models.Lines.LinesV.LinesV, py_dss_interface.models.Lines.LinesS.LinesS.LinesI, py_dss_interface.models.LinesI.LinesI, py_dss_interface.models.Lines.LinesF.LinesF
```

This interface implements the Lines (ILines) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: LinesV, LinesS, LinesF.

```
class py_dss_interface.models.Lines.Lines.LinesF(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the Lines Class where the values are doubles.

The structure of the interface is as follows: double LinesF(int32 t Parameter, double argument)

This interface returns a floating point number, the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
lines\_read\_c0() \rightarrow float
```

Gets the zero sequence capacitance, nanofarads per unit length.

```
{\tt lines\_read\_c1}\,(\,)\,\to {\rm float}
```

Gets the positive sequence capacitance, nanofarads per unit length.

```
lines read emerg amps() \rightarrow float
```

Gets the emergency (maximum) ampere rating of Line.

```
lines\_read\_length() \rightarrow float
```

Gets the length of line section in units compatible with the LineCode definition.

```
{\tt lines\_read\_norm\_amps}\:(\:)\:\to float
```

Gets the normal ampere rating of line section.

```
lines read r0() \rightarrow float
```

Gets the zero sequence resistance, ohm per unit length.

```
\textbf{lines\_read\_r1} \; (\;) \; \rightarrow float
```

Gets the positive sequence resistance, ohm per unit length.

```
lines\_read\_rg() \rightarrow float
```

Gets the earth return value used to compute line impedance's at power frequency.

```
lines read rho() \rightarrow float
```

Gets the earth resistivity, m-ohms.

```
lines read season rating() \rightarrow float
     Returns the rating for the current season (in Amps) if the SeasonalRatings option is active.
lines\_read\_x0() \rightarrow float
     Gets the zero sequence reactance, ohm per unit length.
lines read x1() \rightarrow float
     Gets the positive sequence reactance, ohm per unit length.
lines read xq() \rightarrow float
     Gets the earth return reactance value used to compute line impedances at power frequency.
lines\_write\_cO(argument: float) \rightarrow float
     Sets the zero sequence capacitance, nanofarads per unit length.
lines\_write\_c1 (argument: float) \rightarrow float
     Sets the positive sequence capacitance, nanofarads per unit length.
lines_write_emerg_amps (argument: float) \rightarrow float
     Sets the emergency (maximum) ampere rating of Line.
lines write length (argument: float) \rightarrow float
     Sets the length of line section in units compatible with the LineCode definition.
lines\_write\_norm\_amps(argument: float) \rightarrow float
     Sets the normal ampere rating of Line.
lines write r0 (argument: float) \rightarrow float
     Sets the zero sequence resistance, ohm per unit length.
lines\_write\_r1 (argument: float) \rightarrow float
     Sets the positive sequence resistance, ohm per unit length.
lines\_write\_rg(argument: float) \rightarrow float
     Sets the earth return value used to compute line impedances at power frequency.
lines_write_rho (argument: float) \rightarrow float
     Sets the earth resistivity, m-ohms.
\textbf{lines\_write\_x0} \ (\textit{argument: float}) \ \rightarrow \textbf{float}
     Sets the zero sequence reactance, ohm per unit length.
lines\_write\_x1 (argument: float) \rightarrow float
     Sets the positive sequence reactance, ohm per unit length.
```

lines\_write\_xg (argument: float)  $\rightarrow$  float

Sets the earth return reactance value used to compute line impedances at power frequency.

```
class py_dss_interface.models.Lines.Lines.LinesI(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the Lines Class where the values are integers.

The structure of the interface is as follows: int32\_t LinesI(int32\_t Parameter, int32\_t argument)

This interface returns an integer (signed 32 bits), the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
lines_count () → int

Gets the number of Line Objects in Active Circuit.
```

#### lines first () $\rightarrow$ int

Sets the first element active. Returns 0 if no lines. Otherwise, index of the line element.

#### lines next() $\rightarrow$ int

Sets the next element active. Returns 0 if no lines. Otherwise, index of the line element.

#### lines num cust() $\rightarrow$ int

Gets the number of customers on this line section.

#### lines parent () $\rightarrow$ int

Gets the parents of the active Line to be the active Line. Return 0 if no parent or action fails.

## $lines\_read\_phases() \rightarrow int$

Gets the number of phases of the active line object.

#### lines\_read\_units() $\rightarrow$ int

Gets the units of the line (distance, check manual for details).

```
lines_write_phases (argument: int) \rightarrow int
```

Sets the number of phases of the active line object.

```
lines write units (argument: int) \rightarrow int
```

Sets the units of the line (distance, check manual for details). units: {none | milkftlkmlmlFtlinlcm } UNITS\_MAXNUM =9; UNITS\_NONE =0; UNITS\_MILES =1; UNITS\_KFT =2; UNITS\_KM =3; UNITS\_M =4; UNITS\_FT =5; UNITS\_IN =6; UNITS\_CM =7; UNITS\_MM =8;

```
class py_dss_interface.models.Lines.Lines.LinesS(obj_dss)
    Bases: py dss interface.models.Base.Base
```

This interface can be used to read/modify the properties of the Lines Class where the values are Strings.

. . .

The structure of the interface is as follows: CStr LinesS(int32\_t Parameter, CStr argument)

This interface returns a string, the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

#### $lines\_read\_bus1() \rightarrow str$

Gets the name of bus for terminal 1.

# ${\tt lines\_read\_bus2}\,(\,)\,\to str$

Gets the name of bus for terminal 2.

# $\textbf{lines\_read\_geometry}\,(\,)\,\to str$

Gets the name of the Line geometry code.

#### lines read linecode() $\rightarrow$ str

Gets the name of LineCode object that defines the impedances.

## ${\tt lines\_read\_name}\,(\,)\,\to str$

Gets the name of the active Line element.

## $lines\_read\_spacing() \rightarrow str$

Gets the name of the Line spacing code.

#### lines\_write\_bus1 (argument: str) → str

Sets the name of bus for terminal 1.

## $lines\_write\_bus2 (argument: str) \rightarrow str$

Sets the name of bus for terminal 2.

```
lines_write_geometry (argument: str) \rightarrow str
           Sets the name of the Line geometry code.
      lines\_write\_linecode(argument: str) \rightarrow str
           Sets the name of LineCode object that defines the impedances.
      lines write name (argument: str) \rightarrow str
           Sets the name of the Line element to set it active.
      lines write spacing (argument: str) \rightarrow str
           Sets the name of the Line spacing code.
class py_dss_interface.models.Lines.Lines.LinesV(obj_dss)
      Bases: py_dss_interface.models.Base.Base
      This interface can be used to read/modify the properties of the Lines Class where the values are Variants.
      The structure of the interface is as follows: void Lines V(int32_t Parameter, , VARIANT *Argument);
      This interface returns a Variant, the variable "parameter" is used to specify the property of the class to be used
      and the variable "argument" can be used to modify the value of the property when necessary. Reading and
      writing properties are separated and require a different parameter number to be executed.
      The properties (parameter) are integer numbers and are described as follows.
      lines\_all\_names() \rightarrow str
           Gets the name of all Line Objects.
      lines read cmatrix() \rightarrow str
           Gets the capacitance matrix (full), nanofarads per unit length. Variant array of doubles.
      lines_read_rmatrix()
           Gets the resistance matrix (full), ohms per unit length. Variant array of doubles.
      lines_read_xmatrix() → List[float]
           Gets the reactance matrix (full), ohms per unit length. Variant array of doubles.
      lines\_read\_yprim() \rightarrow str
           Gets the YPrimitive of the active Line.
      lines_write_cmatrix(argument) → int
           Sets the capacitance matrix (full), nanofarads per unit length. Variant array of doubles.
      lines_write_rmatrix(argument) → str
           Sets the resistance matrix (full), ohms per unit length. Variant array of doubles.
      \textbf{lines\_write\_xmatrix} (\textit{argument}) \rightarrow \textit{str}
           Sets the reactance matrix (full), ohms per unit length. Variant array of doubles.
      lines_write_yprim(argument) → str
           According the oficial documentation this parameter does nothing at present.
4.1.20 Loads
```

Created by eniocc at 11/10/2020

This interface implements the Loads (ILoads) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: LoadsF, LoadsI, LoadsS, LoadsV.

# class py\_dss\_interface.models.Loads.Loads.LoadsF(obj\_dss) Bases: py dss interface.models.Base.Base

This interface can be used to read/modify the properties of the Loads Class where the values are floating point numbers (double).

The structure of the interface is as follows: double DSSLoadsF(int32\_t Parameter,double Argument);

This interface returns a Double (IEEE 754 64 bits), the variable "parameter" (Integer) is used to specify the property of the class to be used and the variable "argument" (double) can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

## ${\tt loads\_read\_allocation\_factor}\,(\,)\,\rightarrow {\rm float}$

Allows to read the AllocationFactor property of the active load. The parameter argument can be filled with a 0.

## $loads\_read\_c\_factor() \rightarrow float$

Allows to read the CFactor property of the active load. The parameter argument can be filled with a 0.

#### $loads\_read\_cvr\_vars() \rightarrow float$

Allows to read the CVRvars property of the active load. The parameter argument can be filled with a 0.

## $loads\_read\_cvr\_watts() \rightarrow float$

Allows to read the CVRWatts property of the active load. The parameter argument can be filled with a 0.

#### loads read $kv() \rightarrow float$

Allows to read the kV property of the active load. The parameter argument can be filled with a 0.

## $loads\_read\_kva() \rightarrow float$

Allows to read the kva property of the active load. The parameter argument can be filled with a 0.

# $loads\_read\_kvar() \rightarrow float$

Allows to read the kvar property of the active load. The parameter argument can be filled with a 0.

## $loads\_read\_kw() \rightarrow float$

Allows to read the kW property of the active load. The parameter argument can be filled with a 0.

#### $loads\_read\_kwh() \rightarrow float$

Allows to read the kWh property of the active load. The parameter argument can be filled with a 0.

## loads read kwh days() $\rightarrow$ float

Allows to read the kWhdays property of the active load. The parameter argument can be filled with a 0.

## ${\tt loads\_read\_pct\_mean}\,(\,)\,\rightarrow float$

Allows to read the PctMean property of the active load. The parameter argument can be filled with a 0.

#### loads\_read\_pct\_series\_rl() → float

allows to read the PctSeriesRL (Percent of Load that is modeled as series R-L for harmonic studies) property of the active load. The parameter argument can be filled with a 0.

# $\textbf{loads\_read\_pct\_std\_dev}\,(\,)\,\rightarrow float$

Allows to read the PctStdDev property of the active load. The parameter argument can be filled with a 0.

#### $loads\_read\_pf() \rightarrow float$

Allows to read the pf property of the active load. The parameter argument can be filled with a 0.

## $loads\_read\_r\_neut() \rightarrow float$

Allows to read the RNeut (neutral resistance for wye connected loads) property of the active load. The parameter argument can be filled with a 0.

#### $loads\_read\_rel\_weight() \rightarrow float$

Allows to read the RelWeight (relative weighting factor) property of the active load. The parameter argument can be filled with a 0.

## $loads\_read\_vmax\_pu() \rightarrow float$

Allows to read the VMaxpu property of the active load. The parameter argument can be filled with a 0.

#### loads read vmin emerg() $\rightarrow$ float

Allows to read the VMinemerg property of the active load. The parameter argument can be filled with a 0.

## ${\tt loads\_read\_vmin\_norm}\,(\,)\,\to float$

Allows to read the VMinnorm property of the active load. The parameter argument can be filled with a 0.

#### loads\_read\_vmin\_pu() → float

Allows to read the VMinpu property of the active load. The parameter argument can be filled with a 0.

# $\textbf{loads\_read\_x\_neut} \ () \ \rightarrow float$

Allows to read the Xneut property of the active load. The parameter argument can be filled with a 0.

## $loads\_read\_xfkva() \rightarrow float$

Allows to read the xfKVA (Rated service transformer KVA for load allocation, using Allocationfactor. Affects kW, kvar and pf.) property of the active load. The parameter argument can be filled with a 0.

## $loads\_write\_allocation\_factor(argument) \rightarrow float$

Allows to write the AllocationFactor property of the active load. The parameter argument must contain the new value in AllocationFactor for the desired active load. The return value will be equal to 0.

### loads write c factor (argument: float) $\rightarrow$ float

Allows to write the CFactor property of the active load. The parameter argument must contain the new value in CFactor for the desired active load. The return value will be equal to 0.

## loads\_write\_cvr\_vars (argument) → float

Allows to write the CVRvars property of the active load. The parameter argument must contain the new value in CVRWatts for the desired active load. The return value will be equal to 0.

# $loads\_write\_cvr\_watts(argument) \rightarrow float$

Allows to write the CVRWatts property of the active load. The parameter argument must contain the new value in CVRWatts for the desired active load. The return value will be equal to 0.

#### **loads\_write\_kv** (*argument*) → float

Allows to write the kV property of the active load. The parameter argument must contain the new value in kV for the desired active load. The return value will be equal to 0.

# $loads\_write\_kva(argument) \rightarrow float$

Allows to write the kva property of the active load. The parameter argument must contain the new value in kva for the desired active load. The return value will be equal to 0.

### loads write $kvar(argument) \rightarrow float$

Allows to write the kvar property of the active load. The parameter argument must contain the new value in kvar for the desired active load. The return value will be equal to 0.

## loads\_write\_kw(argument) → float

Allows to write the kW property of the active load. The parameter argument must contain the new value in kW for the desired active load. The return value will be equal to 0.

#### loads write $kwh(argument) \rightarrow float$

Allows to write the kWh property of the active load. The parameter argument must contain the new value in kWh for the desired active load. The return value will be equal to 0.

# $\textbf{loads\_write\_kwh\_days} \ (\textit{argument}) \ \rightarrow \text{float}$

Allows to write the kWhdays property of the active load. The parameter argument must contain the new value in kWhdays for the desired active load. The return value will be equal to 0.

#### loads\_write\_pct\_mean (argument) → float

Allows to write the PctMean property of the active load. The parameter argument must contain the new value in PctMean for the desired active load. The return value will be equal to 0.

# $loads\_write\_pct\_series\_rl(argument) \rightarrow float$

allows to write the PctSeriesRL (Percent of Load that is modeled as series R-L for harmonic studies) property of the active load. The parameter argument must contain the new value in PctSeriesRL for the desired active load. The return value will be equal to 0.

## $loads\_write\_pct\_std\_dev(argument) \rightarrow float$

Allows to write the PctStdDev property of the active load. The parameter argument must contain the new value in PctStdDev for the desired active load. The return value will be equal to 0.

### **loads\_write\_pf** (*argument*) → float

Allows to write the pf property of the active load. The parameter argument must contain the new value in pf for the desired active load. The return value will be equal to 0.

## loads\_write\_r\_neut(argument) → float

Allows to write the RNeut (neutral resistance for wye connected loads) property of the active load. The parameter argument must contain the new value in RNeut for the desired active load. The return value will be equal to 0.

# $loads\_write\_rel\_weight(argument) \rightarrow float$

Allows to write the RelWeight (relative weighting factor) property of the active load. The parameter argument must contain the new value in RelWeight for the desired active load. The return value will be equal to 0.

## $loads\_write\_vmax\_pu(argument) \rightarrow float$

Allows to write the VMaxpu property of the active load. The parameter argument must contain the new value in VMaxpu for the desired active load. The return value will be equal to 0.

# $loads\_write\_vmin\_emerg(argument) \rightarrow float$

Allows to write the VMinemerg property of the active load. The parameter argument must contain the new value in VMinemerg for the desired active load. The return value will be equal to 0.

# $loads\_write\_vmin\_norm(argument) \rightarrow float$

Allows to write the VMinnorm property of the active load. The parameter argument must contain the new value in VMinnorm for the desired active load. The return value will be equal to 0.

# $loads\_write\_vmin\_pu(argument) \rightarrow float$

Allows to write the VMinpu property of the active load. The parameter argument must contain the new value in VMinpu for the desired active load. The return value will be equal to 0.

# $\textbf{loads\_write\_x\_neut} \; (\textit{argument}) \; \rightarrow \text{float}$

Allows to write the Xneut property of the active load. The parameter argument must contain the new value in Xneut for the desired active load. The return value will be equal to 0.

#### loads\_write\_xfkva(argument) → float

Allows to write the xfKVA (Rated service transformer KVA for load allocation, using Allocationfactor. Affects kW, kvar and pf.) property of the active load. The parameter argument must contain the new value in xfKVA for the desired active load. The return value will be equal to 0.

# ${\bf class} \ {\tt py\_dss\_interface.models.Loads.Loads.Loads.Loads} \ ({\it obj\_dss})$

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/modify the properties of the Loads Class where the values are integers.

# The structure of the interface is as follows: int32\_t DSSLoads(int32\_t Parameter, int32\_t argument)

This interface returns an integer (signed 32 bits), the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary.

Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

## $loads\_count() \rightarrow int$

Returns the number of load elements within the active circuit. The parameter argument can be filled with a 0.

#### loads first() $\rightarrow$ int

Allows to set the active load into the first load registered in the active circuit. As a result, this property will return the number 1. The parameter argument can be filled with a 0.

#### loads next() $\rightarrow$ int

Sets the active load into the next load registered in the active circuit. As a result, this property will deliver the index of the active load. The parameter argument can be filled with a 0.

#### loads read class() $\rightarrow$ int

Allows to read the code number used to separate loads by class or group. The parameter argument can be filled with a 0.

#### loads read $idx() \rightarrow int$

Allows to read the index of the active load. The parameter argument can be filled with a 0.

# $\textbf{loads\_read\_is\_delta}\,(\,)\,\rightarrow int$

Allows to read if the active load is connected in delta, if the answer is positive, this function will deliver a 1; otherwise, the answer will be 0. The parameter argument can be filled with a 0.

### loads read model() $\rightarrow$ int

Allows to read the model of the active load. The parameter argument can be filled with a 0.

# ${\tt loads\_read\_num\_cust}\:(\:)\:\to int$

Allows to read the number of customer of the active load. The parameter argument can be filled with a 0.

### $loads\_read\_status() \rightarrow int$

Allows to read Response to load multipliers: Fixed (growth only - 1), Exempt (no LD curve - 2), Variable (all - 0), of the active load. The parameter argument can be filled with a 0.

#### loads\_write\_class(argument) → int

Allows to read the code number used to separate loads by class or group. The parameter argument can be filled with a 0.

## $loads\_write\_idx (argument) \rightarrow int$

Allows to write the index of the active load. The parameter argument must contain the index of the desired active load. The return value will be equal to 0.

## $loads\_write\_is\_delta(argument) \rightarrow int$

Allows to read if the active load is connected in delta, if the answer is positive, this function will deliver a 1; otherwise, the answer will be 0. This parameter will return a 0.

## $loads\_write\_model(argument) \rightarrow int$

Allows to write the model of the active load using the parameter argument. This parameter will return a 0.

# $\textbf{loads\_write\_num\_cust} \ (\textit{argument}) \ \rightarrow int$

Allows to write the number of customers of the active load using the parameter argument. This parameter will return a 0.

## $loads\_write\_status(argument) \rightarrow int$

Allows to read Response to load multipliers: Fixed (growth only - 1), Exempt (no LD curve - 2), Variable (all - 0), of the active load. This parameter will return a 0.

```
class py_dss_interface.models.Loads.Loads.Loads(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the Loads Class where the values are strings.

## The structure of the interface is as follows: CStr DSSLoadsS(int32\_t Parameter, CStr Argument);

This interface returns a string, the variable "parameter" (Integer) is used to specify the property of the class to be used and the variable "argument" (string) can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

## $\textbf{loads\_read\_cvr\_curve} \ () \ \rightarrow str$

Allows to read the CVRCurve property of the active load. The parameter argument can be filled with an empty string.

## $loads\_read\_daily() \rightarrow str$

Allows to read the daily property of the active load. The parameter argument can be filled with an empty string.

## $loads\_read\_duty() \rightarrow str$

Allows to read the duty property of the active load. The parameter argument can be filled with an empty string.

# $\textbf{loads\_read\_growth}\:(\:)\:\to str$

Allows to read the Growth property of the active load. The parameter argument can be filled with an empty string.

## ${\tt loads\_read\_name}\,(\,)\,\to str$

Allows to read the Name property of the active load. The parameter argument can be filled with an empty string.

## ${\tt loads\_read\_spectrum}\,(\,)\,\to str$

Allows to read the Spectrum property of the active load. The parameter argument can be filled with an empty string.

# ${\tt loads\_read\_yearly}\,(\,)\,\to str$

Allows to read the Yearly property of the active load. The parameter argument can be filled with an empty string.

# $loads\_write\_cvr\_curve(argument) \rightarrow str$

Allows to set the CVRCurve property for the active load. The parameter argument must contain the Name of the new CVRCurve to be linked to the active load. The return value will be equal to empty.

# $\textbf{loads\_write\_daily} (\textit{argument}) \rightarrow \textit{str}$

Allows to set the daily property for the active load. The parameter argument must contain the Name of the new daily to be linked to the active load. The return value will be equal to empty.

## $loads\_write\_duty(argument) \rightarrow str$

Allows to set the dduty property for the active load. The parameter argument must contain the Name of the new duty to be linked to the active load. The return value will be equal to empty.

## $loads\_write\_growth(argument) \rightarrow str$

Allows to set the Growth property for the active load. The parameter argument must contain the Name of the new Growth to be linked to the active load. The return value will be equal to empty.

## $loads\_write\_name(argument) \rightarrow str$

allows to set the active load by specifying the Name load. The parameter argument must contain the Name of the load to activate. The return value will be equal to empty.

# $loads\_write\_spectrum(argument) \rightarrow str$

Allows to set the Spectrum property for the active load. The parameter argument must contain the Name of the new Spectrum to be linked to the active load. The return value will be equal to empty.

```
loads write yearly (argument) \rightarrow str
```

Allows to set the Yearly property for the active load. The parameter argument must contain the Name of the new Yearly to be linked to the active load. The return value will be equal to empty.

```
class py_dss_interface.models.Loads.Loads.LoadsV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the Loads Class where the values are variants (the value can have different formats).

The structure of the interface is as follows: void DSSLoadsV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a string, the variable "parameter" (Integer) is used to specify the property of the class to be used and the variable "argument" (Variant) is used to return the variant structure.

```
loads_all_names()
```

Allows to read the names of all the loads present in the active circuit. The result is delivered as variant, however, the content of this variant is an array of strings.

```
loads_read_zipv()
```

Allows to read the array of 7 elements (doubles) for ZIP property of the active Load object.

```
loads_write_zipv (argument)
```

Allows to write the array of 7 elements (doubles) for ZIP property of the active Load object. :param argument: Array of 7 coefficients:

First 3 are ZIP weighting factors for real power (should sum to 1) Next 3 are ZIP weighting factors for reactive power (should sum to 1) Last 1 is cut-off voltage in p.u. of base kV; load is 0 below this cut-off No defaults; all coefficients must be specified if using model=8.

# 4.1.21 LoadShapes

Created by eniocc at 11/10/2020

This interface implements the LoadShape (ILoadShape) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: .

```
class py_dss_interface.models.LoadShapes.LoadShapes.LoadShapesF(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double LoadShapeF(int32\_t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
loadshapes_read_hr_interval() \rightarrow float
```

Gets the fixed interval time value, hours.

```
{\tt loadshapes\_read\_min\_interval}\:(\:)\:\to float
```

Gets the fixed interval time value, in minutes.

```
loadshapes\_read\_p\_base() \rightarrow float
```

Gets the base for normalizing P curve. If left at zero, the peak value is used.

#### loadshapes read q base() $\rightarrow$ float

Gets the base for normalizing Q curve. If left at zero, the peak value is used.

## $loadshapes\_read\_s\_interval() \rightarrow float$

Gets the fixed interval data time interval, seconds.

#### loadshapes\_write\_hr\_interval(argument: float) → float

Sets the fixed interval time value, hours.

#### loadshapes\_write\_min\_interval (argument: float) → float

Sets the fixed interval time value, in minutes.

## $loadshapes\_write\_p\_base(argument: float) \rightarrow float$

Sets the base for normalizing P curve. If left at zero, the peak value is used.

#### loadshapes\_write\_q\_base (argument: float) $\rightarrow$ float

Sets the base for normalizing Q curve. If left at zero, the peak value is used.

#### loadshapes\_write\_s\_interval(argument: float) → float

Sets the fixed interval data time interval, seconds.

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t LoadShapeI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer number with the result of the query according to the value of the variable Parameter, which can be one of the following.

## loadshapes\_count() $\rightarrow$ int

Returns the number of LoadShape objects currently defined in LoadShape collection.

# $\textbf{loadshapes\_first} \; () \; \rightarrow int$

sets the first loadshape active and return integer index of the loadshape. Returns 0 if no more.

# $\textbf{loadshapes\_next} \; (\;) \; \rightarrow int$

Sets the next loadshape active and return integer index of the loadshape. Returns 0 if no more.

## ${\tt loadshapes\_normalize()} \rightarrow int$

Normalizes the P and Q curves based on either Pbase, Qbase or simply the peak value of the curve.

# ${\tt loadshapes\_read\_npts}\,(\,)\,\rightarrow int$

Gets the number of points in active LoadShape.

## ${\tt loadshapes\_read\_use\_actual}\:(\:)\:\to int$

Gets a TRUE/FALSE (1/0) to let Loads know to use the actual value in the curve rather than use the value as a multiplier.

#### loadshapes\_write\_npts(argument) → int

Sets the number of points in active LoadShape.

# ${\tt loadshapes\_write\_use\_actual}~(\textit{argument})~\rightarrow int$

Sets a TRUE/FALSE (1/0 - Argument) to let Loads know to use the actual value in the curve rather than use the value as a multiplier.

```
class py_dss_interface.models.LoadShapes.LoadShapesS.(obj_dss)
```

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr LoadShapeS(int32 t Parameter, CStr Argument);

This interface returns string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
loadshapes_read_name()
```

Gets the name of the active LoadShape object.

```
loadshapes_write_name (argument)
```

Sets the name of the active LoadShape object.

```
class py_dss_interface.models.LoadShapes.LoadShapes.LoadShapesV(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void LoadShapeV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
loadshapes_all_names()
```

Gets a variant array of strings containing names of all LoadShape objects currently defined.

```
loadshapes_read_p_mult()
```

Gets a variant array of doubles for the P multiplier in the LoadShape.

```
loadshapes_read_q_mult()
```

Gets a variant array of doubles for the Q multiplier in the LoadShape.

```
loadshapes_read_time_array()
```

Gets a time array in hours corresponding to P and Q multipliers when the Interval = 0.

```
loadshapes_write_p_mult (argument)
```

Sets a variant array of doubles for the P multiplier in the LoadShape.

```
loadshapes_write_q_mult (argument)
```

Sets a variant array of doubles for the Q multiplier in the LoadShape.

```
loadshapes_write_time_array(argument)
```

Sets a time array in hours corresponding to P and Q multipliers when the Interval = 0.

# 4.1.22 Meters

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.Meters.Meters.Meters(obj_dss)
    Bases: py_dss_interface.models.Meters.MetersV.MetersV, py_dss_interface.
    models.MetersS.MetersS, py_dss_interface.models.MetersF.
    MetersF, py_dss_interface.models.MetersI.MetersI
```

This interface implements the Meters (IMeters) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: MetersV, MetersF, MetersF, MetersI.

```
class py_dss_interface.models.Meters.Meters.Metersf(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double MetersF(int32\_t Parameter, double Argument);

This interface returns a floating point number (64 bits) according to the number sent in the variable "parameter". The parameter can be one of the following.

```
meters avg repair time () \rightarrow float
```

Returns the average Repair Time in this Section of the meter zone.

```
meters\_cust\_interrupts() \rightarrow float
```

Returns the total customer interruptions for this meter zone based on reliability calcs.

```
meters\_fault\_rate\_x\_repair\_hrs() \rightarrow float
```

Returns the sum of Fault Rate Time Repair Hours in this section of the meter zone.

```
meters saidi() \rightarrow float
```

Returns the SAIDI for this meter zone. Execute DoreliabilityCalc first.

```
\textbf{meters\_saifi} () \rightarrow float
```

Returns SAIFI for this meter's zone. Execute reliability calc method first.

```
meters_saifi_kw() \rightarrow float
```

Returns the SAIFI based on kW rather than number of customers. Get after reliability calcs.

```
meters\_sum\_branch\_flt\_rates() \rightarrow float
```

Returns the sum of the branch fault rates in this section of the meter's zone.

```
class py_dss_interface.models.Meters.Meters.MetersI(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t MetersI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer according to the number sent in the variable "parameter". The parameter can be one of the following.

```
{\tt meters\_close\_all\_di\_files\,()} \, \to int
```

Closes all Demand Interval (DI) files. Necessary at the end of a run.

```
meters\_count() \rightarrow int
```

Returns the number of Energy Meters in the Active Circuit.

```
\textbf{meters\_count\_branches}\;(\;)\;\to int
```

Returns the number of branches in active Energy Meter zone (same as sequencelist size).

```
{\tt meters\_count\_end\_elements}\:(\:)\:\to int
```

Returns the number of zone end elements in the active meter zone.

```
meters\_di\_files\_are\_open() \rightarrow int
```

Returns a global flag (1=true, 0=false) to indicate if Demand Interval (DI) files have been properly opened.

```
meters\_do\_reliability\_calc() \rightarrow int
```

Calculates SAIFI, etc. if the Argument is equal to 1 this parameter will assume restoration, otherwise it will not.

```
meters\_first() \rightarrow int
```

Sets the first Energy Meter active. Returns 0 if no monitors.

```
meters\_next() \rightarrow int
```

Sets the next energy Meter Active. Returns 0 if no more.

```
meters\_num\_section\_branches() \rightarrow int
```

Returns the number of branches (lines) in the active section.

```
meters_num_section_customers() \rightarrow int
```

Returns the number of customers in the active section.

```
meters num sections() \rightarrow int
```

Returns the number of feeder sections in this meter's zone.

```
meters_ocp_device_type() → int
```

Returns the type of OCP device: {1=fuse | 2+ recloser | 3= relay}.

## $meters\_open\_all\_di\_files() \rightarrow int$

Opens Demand Interval (DI) files. Returns 0.

#### $meters\_read\_metered\_terminal() \rightarrow int$

Returns the number of metered terminal by the active Energy Meter.

```
meters read sequence index() \rightarrow int
```

Returns the index into meter's SequenceList that contains branch pointers in lexical order. Earlier index guaranteed to be up line from later index. Sets PDElement active.

```
meters\_reset() \rightarrow int
```

Resets the active Meter object.

```
meters\_reset\_all() \rightarrow int
```

Resets all Meter object.

```
meters\_sample() \rightarrow int
```

Causes active meter to take a sample.

```
{\tt meters\_sample\_all}\,(\,)\,\to int
```

Causes all Energy Meters to take a sample of the present state. Returns 0.

```
meters save() \rightarrow int
```

Causes active meter to save its current sample buffer to its meter stream. Then you can access the Bytestream or channel data. Most standard solution modes do this automatically.

```
meters save all() \rightarrow int
```

Save all Energy Meter buffers to their respective file streams. Returns 0.

```
meters\_sect\_seq\_idx() \rightarrow int
```

Returns the Sequence Index of the branch at the head of this section.

```
\textbf{meters\_sect\_total\_cust} \; (\;) \; \to int
```

Returns the total customers down line from this section.

```
meters\_seq\_list\_size() \rightarrow int
```

Returns the size of Sequence List.

```
meters_set_active_section (argument) \rightarrow int
```

Sets the designated section (argument) if the index is valid.

```
{\tt meters\_total\_customers}\:(\:)\:\to int
```

Returns the total number of customers in this zone (down line from the Energy Meter).

```
meters write metered terminal (argument) \rightarrow int
```

Sets the number of metered terminal by the active Energy Meter.

```
meters_write_sequence_index (argument) → int
```

Sets the index into meter's SequenceList that contains branch pointers in lexical order. Earlier index guaranteed to be up line from later index. Sets PDElement active.

```
{\tt class} \ {\tt py\_dss\_interface.models.Meters.Meters.Meters} \ (obj\_dss)
```

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr MetersS(int32\_t Parameter, CStr Argument);

This interface returns a string according to the number sent in the variable "parameter". The parameter can be one of the following.

#### meters read metered element()

Returns the name of the metered element (considering the active Energy Meter).

## meters\_read\_name()

Returns the active Energy Meter's name.

#### meters write metered element (argument)

Sets the name of the metered element (considering the active Energy Meter).

#### meters write name(argument)

Sets the active Energy Meter's name.

## class py\_dss\_interface.models.Meters.Meters.MetersV(obj\_dss)

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void MetersV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a variant according to the number sent in the variable "parameter". The parameter can be one of the following.

#### meters\_all\_branches\_in\_zone()

Returns a wide string list of all branches in zone of the active Energy Meter object.

#### meters\_all\_end\_elements()

Returns a variant array of names of all zone end elements.

#### meters all names()

Returns an array of all Energy Meter names.

#### meters\_all\_pce\_in\_zone()

This parameter returns a wide string list of all the PCE in zone of the active Energy Meter object.

## meters\_read\_alloc\_factors()

Returns an array of doubles: allocation factors for the active Meter.

## meters\_read\_calc\_current()

Returns the magnitude of the real part of the Calculated Current (normally determined by solution) for the meter to force some behavior on Load Allocation.

#### meters\_read\_peak\_current()

Returns an array of doubles with the Peak Current Property.

# meters\_register\_names()

Returns an array of strings containing the names of the registers.

## meters\_register\_values()

Returns an array of values contained in the Meter registers for the active Meter.

#### meters\_totals()

Returns the totals for all registers of all Meters.

# meters\_write\_alloc\_factors(argument)

Receives an array of doubles to set the phase allocation factors for the active Meter.

#### meters\_write\_calc\_current (argument: str)

Sets the magnitude of the real part of the Calculated Current (normally determined by solution) for the meter to force some behavior on Load Allocation.

## meters\_write\_peak\_current (argument)

Receives an array of doubles to set values of Peak Current Property.

# 4.1.23 Monitors

```
Created by eniocc at 11/10/2020
```

This interface implements the Monitors (IMonitors) interface of OpenDSS by declaring 3 procedures for accessing the different properties included in this interface: MonitorsI, MonitorsV, MonitorsS.

```
class py_dss_interface.models.Monitors.Monitors.MonitorsI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t MonitorsI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer according to the number sent in the variable "parameter". The parameter can be one of the following.

```
{\tt monitors\_count}\;(\;)\;\to int
```

Returns the number of monitors.

```
\textbf{monitors\_file\_version}\,(\,)\,\rightarrow int
```

Returns the Monitor File version (integer).

```
monitors_first() \rightarrow int
```

Sets the first monitor active. Returns 0 if no monitors.

```
monitors\_next() \rightarrow int
```

Set the next monitor active. Returns 0 if no more.

```
{\tt monitors\_num\_channels}\,(\,)\,\to int
```

Returns the number of Channels on the active Monitor.

```
monitors\_process() \rightarrow int
```

Post-process monitor examples taken so far, e.g., Pst for mode = 4.

```
monitors process all() \rightarrow int
```

Makes that all monitors post-process the data taken so far.

```
{\tt monitors\_read\_mode} () \to int
```

Returns the monitor mode (bitmask integer - see DSS Help).

```
\textbf{monitors\_read\_terminal}\,(\,)\,\rightarrow int
```

Returns the terminal number of element being monitored.

```
monitors\_record\_size() \rightarrow int
```

Returns the size of each record in ByteStream.

```
monitors\_reset() \rightarrow int
```

Resets the active Monitor object.

```
monitors\_reset\_all() \rightarrow int
```

Resets all Monitor object.

```
monitors\_sample() \rightarrow int
```

Causes active monitor to take a sample.

```
monitors\_sample\_all() \rightarrow int
```

Causes all Monitors to take a sample of the present state. Returns 0.

#### monitors sample count() $\rightarrow$ int

Returns number of examples in Monitor at present.

```
monitors save() \rightarrow int
```

Causes active monitor to save its current sample buffer to its monitor stream. Then you can access the Bytestream or channel data. Most standard solution modes do this automatically.

```
monitors save all() \rightarrow int
```

Save all Monitor buffers to their respective file streams. Returns 0.

```
\textbf{monitors\_show}\,(\,)\,\rightarrow int
```

Converts monitor file into text and displays with text editor.

```
monitors\_write\_mode(argument) \rightarrow int
```

Sets the monitor mode (bitmask integer - see DSS Help). Bitmask integer designating the values the monitor is to capture: 0 = Voltages and currents at designated terminal 1 = Powers at designated terminal 2 = Tap Position (Transformer Device only) 3 = State Variables (PCElements only) 4 = Flicker level and severity index (Pst) for voltages. No adders apply.

Flicker level at simulation time step, Pst at 10-minute time step.

5 = Solution variables (Iterations, etc). Normally, these would be actual phasor quantities from solution. 6 = Capacitor Switching (Capacitors only) 7 = Storage state vars (Storage device only) 8 = All winding currents (Transformer device only) 9 = Losses, watts and var (of monitored device) 10 = All Winding voltages (Transformer device only) Normally, these would be actual phasor quantities from solution. 11 = All terminal node voltages and line currents of monitored device Combine mode with adders below to achieve other results for terminal quantities: +16 = Sequence quantities +32 = Magnitude only +64 = Positive sequence only or avg of all phases

Mix adder to obtain desired results. For example: Mode=112 will save positive sequence voltage and current magnitudes only Mode=48 will save all sequence voltages and currents, but magnitude only.

```
monitors_write_terminal(argument) → int
```

Sets sets the terminal number of element being monitored.

```
class py_dss_interface.models.Monitors.Monitors.MonitorsS(obj_dss)
     Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr MonitorsS(int32\_t Parameter, CStr Argument);

This interface returns a string according to the number sent in the variable "parameter". The parameter can be one of the following.

```
monitors\_file\_name() \rightarrow str
```

Returns the name of the CSV file associated with active monitor.

```
monitors_read_element() \rightarrow str
```

Returns the full name of element being monitored by the active Monitor.

```
\textbf{monitors\_read\_name} \ (\ ) \ \to str
```

Returns the active Monitor object by name.

```
monitors\_write\_element(argument) \rightarrow str
```

Sets the full name of element being monitored by the active Monitor.

```
monitors\_write\_name (argument) \rightarrow str
```

Sets the active Monitor object by name.

```
class py_dss_interface.models.Monitors.Monitors.MonitorsV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object. The structure of the interface is as follows:

```
void MonitorsV(int32 t Parameter, VARIANT *Argument);
```

This interface returns a variant according to the number sent in the variable "parameter". The parameter can be one of the following.

#### monitors\_all\_names()

Returns an array of all Monitor names (array of strings).

```
monitors_byte_stream()
```

Returns a byte array containing monitor stream values. Make sure a "save" is done first (standard solution modes do this automatically).

```
monitors_channel(argument) → str
```

Returns a variant array of doubles for the specified channel (usage: MyArray = DSSmonitor. Channel(i)) A save or SaveAll should be executed first. Done automatically by most standard solution modes.

```
monitors_dbl_freq()
```

Returns a variant array of doubles containing time values for harmonics mode solutions; empty for time mode solutions (use dblHour).

```
monitors_dbl_hour()
```

Returns returns a variant array of doubles containing time value in hours for the time-sampled monitor values; empty if frequency-sampled values for harmonics solution (see dblFreq).

#### monitors header()

Returns the header string; Variant array of strings containing Channel Names.

# 4.1.24 Parallel

Created by eniocc at 11/10/2020

These interfaces allows users to use the parallel processing features included in OpenDSS-PM. With this interface it is possible to create multiple actors, specify the CPU where the actor will be executed, control the execution of the actors, check the actors status and progress among many other functionalities.

```
class py_dss_interface.models.Parallel.Parallel.ParallelI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface allows to control parameters of the parallel computing suite of OpenDSS-PM where its value can be specified as an integer number.

The structure of the interface is as follows: int32\_t ParalleI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
parallel\_create\_actor() \rightarrow int
```

Creates a new actor and sets the active actor ID as the ID for the recently created actor. If there are no more CPUs available, the system will not allow the creation of the new actor.

```
parallel\_num\_actors() \rightarrow int
```

Gets the number of actors created in the actual session.

#### parallel num cores () $\rightarrow$ int

Returns the number of physical cores available in the local computer. If your computer has less than 64 Cores, this number should be the number of CPUs/2. For more information, please check: https://www.howtogeek.com/194756/cpu-basics-multiple-cpus-cores-and-hyper-threading-explained/.

#### $parallel\_num\_cpus() \rightarrow int$

Returns the number of CPUs available in the local computer.

#### $parallel\_read\_active\_actor() \rightarrow int$

Returns the ID of the active actor.

### $parallel\_read\_active\_parallel() \rightarrow int$

Gets if the parallel features of OpenDSS-PM are active. If active, this parameter will return 1, otherwise, will return 0 and OpenDSS-PM will behave sequentially.

#### parallel\_read\_actor\_cpu() → int

Gets the ID of the CPU assigned for the execution of the active actor.

## ${\tt parallel\_read\_concatenate\_reportsl}\:(\:)\:\to int$

Gets the state of the ConcatenateReports property of OpenDSS-PM. If 1, means that every time the user executes a Show/Export monitor operation, the data stored on the monitors with the same name for each actor will be concatenated one after the other. Otherwise (0), to get access of each monitor the user will have to activate the actor of interest and then perform the Show/Export command on the desired monitor.

#### parallel wait() $\rightarrow$ int

Waits until all the actors are free and ready to receive a new command.

## parallel\_write\_active\_actor (argument) → int

Sets the ID of the active actor; this number cannot be higher than the number of existing actors.

```
parallel\_write\_active\_parallel(argument) \rightarrow int
```

Sets enables/disables the parallel features of OpenDSS-PM. To enable set the argument in 1, otherwise, the argument should be 0 and OpenDSS-PM will behave sequentially.

## $parallel\_write\_actor\_cpu(argument) \rightarrow int$

Sets the CPU for the execution of the active actor.

```
parallel_write_concatenate_reportsl(argument) → int
```

Sets the state of the ConcatenateReports property of OpenDSS-PM. If 1, means that every time the user executes a Show/Export monitor operation, the data stored on the monitors with the same name for each actor will be concatenated one after the other. Otherwise (0), to get access of each monitor the user will have to activate the actor of interest and then perform the Show/Export command on the desired monitor.

```
class py_dss_interface.models.Parallel.Parallel.ParallelV(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void ParallelV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

## parallel\_actor\_progress()

Returns an array of integers containing the progress in percentage for each active actor.

## parallel\_actor\_status()

Returns an array of integers containing the status of each active actor. If 1, the actor is ready to receive new commands, if 0, the actor is busy performing a simulation and cannot take new ?solve? commands at this time. However, the actor is capable to deliver values while the simulation is being performed.

# 4.1.25 Parser

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.Parser.Parser.Parser(obj_dss)
    Bases: py_dss_interface.models.Parser.ParserI.ParserI, py_dss_interface.
    models.Parser.ParserS.ParserS, py_dss_interface.models.Parser.ParserF.
    ParserF, py_dss_interface.models.Parser.ParserV.ParserV
```

This interface implements the CmathLib (ICmathLib) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: ParserI, ParserF, ParserV.

```
parser_matrix()
```

Use this property to parse a Matrix token in OpenDSS format. Returns square matrix of order specified. Order same as default fortran order: column by column.

```
parser_symmatrix()
```

Use this property to parse a Matrix token in lower triangular form. Symmetry is forced.

```
parser_vector()
```

Returns token as variant array of doubles. For parsing quoted array syntax.

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double ParserF(int32\_t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
parser_dbl_value() → float
```

Returns next parameter as a double.

```
class py_dss_interface.models.Parser.Parser.ParserI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32 t ParserI(int32 t Parameter, int32 t Argument);

This interface returns an integer number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
parser_int_value() \rightarrow int
```

Returns next parameter as a long integer.

```
parser_read_auto_increment() → int
```

In this parameter the default is false (0). If true (1) parser automatically advances to next token after DblValue, IntValue, or StrValue. Simpler when you don't need to check for parameter names.

```
parser\_reset\_delimeters() \rightarrow int
```

Reset delimiters to their default values.

```
{\tt parser\_write\_auto\_increment} \; (\mathit{argument}) \; \rightarrow int
```

In this parameter the default is false (0). If true (1) parser automatically advances to next token after DblValue, IntValue, or StrValue. Simpler when you don't need to check for parameter names.

```
class py_dss_interface.models.Parser.Parser.ParserS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

## The structure of the interface is as follows: CStr ParserS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

#### parser\_next\_param()

Gets next token and return tag name (before = sign) if any. See Autoincrement.

#### parser\_read\_begin\_quote()

Gets the string containing the characters for quoting in OpenDSS scripts. Matching pairs defined in EndQuote. Default is "([{.

## parser\_read\_cmd\_string()

Gets a string to be parsed. Loading this string resets the parser to the beginning of the line. Then parse off the tokens in sequence.

#### parser\_read\_delimiters()

Gets the string defining hard delimiters used to separate token on the command string. Default is , and =. The = separates token name from token value. These override whitespaces to separate tokens.

# parser\_read\_end\_quote()

Gets the string containing the characters, in order, that match the beginning quote characters in Begin-Quote. Default is ")]}.

## parser\_read\_white\_space()

Gets the characters used for White space in the command string. Default in blank and Tab.

#### parser\_str\_value()

Returns next parameter as a string.

## parser\_write\_begin\_quote(argument)

Sets the string containing the characters for quoting in OpenDSS scripts. Matching pairs defined in EndQuote. Default is "([{.

# parser\_write\_cmd\_string(argument)

Sets a string to be parsed. Loading this string resets the parser to the beginning of the line. Then parse off the tokens in sequence.

## parser\_write\_delimiters (argument)

Sets the string defining hard delimiters used to separate token on the command string. Default is, and =. The = separates token name from token value. These override whitespace to separate tokens.

## parser write end quote(argument)

Sets the string containing the characters, in order, that match the beginning quote characters in BeginQuote. Default is ")]}.

#### parser write white space(argument)

Sets the characters used for White space in the command string. Default in blank and Tab.

This interface can be used to read/write certain properties of the active DSS object.

## The structure of the interface is as follows: void ParserV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

## parser\_matrix()

Use this property to parse a Matrix token in OpenDSS format. Returns square matrix of order specified. Order same as default fortran order: column by column.

```
parser sym matrix()
```

Use this property to parse a Matrix token in lower triangular form. Symmetry is forced.

```
parser_vector()
```

Returns token as variant array of doubles. For parsing quoted array syntax.

## 4.1.26 PDElements

Created by eniocc at 11/10/2020

This interface implements the PDElements (IPDElements) interface of OpenDSS by declaring 3 procedures for accessing the different properties included in this interface: PDElementsF, PDElementsI, PDElementsS.

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double PDElementsF(int32\_t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
pdelements_accumulated_1() \rightarrow float
```

Gets the accumulated failure rate for this branch on down line.

```
pdelements\_lambda() \rightarrow float
```

Gets the failure rate for this branch. Faults per year including length of line.

```
pdelements_read_fault_rate() → float
```

Gets the number of failures per year. For LINE elements: Number of failures per unit length per year.

```
{\tt pdelements\_read\_pct\_permanent}\;(\;)\;\to float
```

Gets the percent of faults that are permanent (require repair). Otherwise, fault is assumed to be transient/temporary.

```
pdelements\_repair\_time() \rightarrow float
```

Gets the average time to repair a permanent fault on this branch, hours.

```
pdelements\_total\_miles() \rightarrow float
```

Gets the total miles of line from this element to the end of the zone. For recloser siting algorithm.

```
pdelements_write_fault_rate(argument) → float
```

Sets the number of failures per year. For LINE elements: Number of failures per unit length per year.

```
pdelements\_write\_pct\_permanent(argument) \rightarrow float
```

Sets the percent of faults that are permanent (require repair). Otherwise, fault is assumed to be transient/temporary.

```
class py_dss_interface.models.PDElements.PDElements.PDElementsI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t PDElementsI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

#### $pdelements\_count() \rightarrow int$

Gets number of PDElements in active circuit.

## $pdelements\_first() \rightarrow int$

Sets the first enabled PD element to be the active element. Returns 0 if none found.

#### pdelements from terminal() $\rightarrow$ int

Gets the number of the terminal of active PD element that is on the "from" side. This is set after the meter zone is determined.

```
pdelements_is_shunt() \rightarrow int
```

Sets returns 1 if the PD element should be treated as a shunt element rather than a series element. Applies to capacitor and reactor elements in particular.

```
pdelements\_next() \rightarrow int
```

Sets the next enabled PD element to be the active element. Returns 0 if none found.

```
pdelements_num_customers() → int
```

Gets the number of customers in this branch.

```
{\tt pdelements\_parent\_pd\_element}\;(\;)\;\to int
```

Gets the parent PD element to be the active circuit element. Returns 0 if no more elements upline.

```
pdelements\_section\_id() \rightarrow int
```

Gets the integer ID of the feeder section that this PDElement branch is part of.

```
pdelements_total_customers() \rightarrow int
```

Gets the total number of customers from this branch to the end of the zone.

```
class py_dss_interface.models.PDElements.PDElements.PDElementsS(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr PDElementsF(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
pdelements_read_name()
```

Gets the name of the active PDElement, returns null string if active element id not PDElement.

```
pdelements_write_name (argument)
```

Sets the name of the active PDElement, returns null string if active element id not PDElement.

# 4.1.27 PVSystems

Created by eniocc at 11/10/2020

This interface implements the PVSystems (IPVSystems) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: PVSystemsV, PVSystemsS, PVSystemsI, PVSystemsF.

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double PVSystemsF(int32\_t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
pvsystems kw() \rightarrow float
          Gets the kW output.
     pvsystems\_read\_irradiance() \rightarrow float
          Gets the present value of the Irradiance property in W/sq-m.
     pvsystems\_read\_kva\_rated() \rightarrow float
          Gets the rated kVA of the PVSystem.
     pvsystems\_read\_kvar() \rightarrow float
          Gets the kvar value.
     pvsystems\_read\_pf() \rightarrow float
          Gets the power factor value.
     pvsystems\_read\_pmpp() \rightarrow float
          Gets the Pmpp.
     pvsystems_write_irradiance (argument) → float
          Sets the present value of the Irradiance property in W/sq-m.
     pvsystems_write_kva_rated(argument) → float
          Sets the rated kVA of the PVSystem.
     pvsystems_write_kvar(argument) → float
          Sets the kvar value.
     pvsystems_write_pf (argument) → float
          Sets the power factor value.
     pvsystems\_write\_pmpp(argument) \rightarrow float
          Sets the Pmpp.
class py_dss_interface.models.PVSystems.PVSystems.PVSystemsI(obj_dss)
     Bases: py_dss_interface.models.Base.Base
     This interface can be used to read/write certain properties of the active DSS object.
     The structure of the interface is as follows: int32_t PVSystemsI(int32_t Parameter, int32_t Argument);
```

This interface returns an integer number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
Returns the number of PVSystem objects currently defined in the active circuit.
pvsystems_first() \rightarrow int
     Sets the first PVSystem to be active; returns 0 if none.
pvsystems_next() \rightarrow int
     Sets the next PVSystem to be active; returns 0 if none.
pvsystems\_read\_idx() \rightarrow int
     Gets the active PVSystem by index; 1..Count.
pvsystems write idx(argument) \rightarrow int
```

Sets the active PVSystem by index; 1..Count..

 $pvsystems\_count() \rightarrow int$ 

```
class py_dss_interface.models.PVSystems.PVSystems.PVSystemsS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr PVSystemsS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
pvsystems_read_name()
```

Gets the name of the active PVSystem.

```
pvsystems_write_name (argument)
```

Sets the name of the active PVSystem.

```
class py_dss_interface.models.PVSystems.PVSystems.PVSystemsV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void PVSystemsV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
pvsystems all names()
```

Gets the variant array of string containing names of all PVSystems in the circuit.

# 4.1.28 Reclosers

This interface implements the Reclosers (IReclosers) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: ReclosersI, ReclosersV, ReclosersS, ReclosersF.

```
class py_dss_interface.models.Reclosers.Reclosers.ReclosersF(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double ReclosersF(int32\_t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
reclosers_read_ground_inst() → float
```

Gets the ground (3I0) instantaneous trip setting - curve multiplier or actual amps.

```
{\tt reclosers\_read\_ground\_trip()} \to {\tt float}
```

Gets the ground (3I0) trip multiplier or actual amps.

```
{\tt reclosers\_read\_phase\_inst} \; (\,) \; \to float
```

Gets the phase instantaneous curve multiplier or actual amps.

```
reclosers\_read\_phase\_trip() \rightarrow float
```

Gets the phase trip curve multiplier or actual amps.

```
reclosers\_write\_ground\_inst(argument) \rightarrow float
```

```
reclosers_write_ground_trip(argument) → float
           Sets the ground (3I0) trip multiplier or actual amps.
      reclosers\_write\_phase\_inst(argument) \rightarrow float
           Sets the phase instantaneous curve multiplier or actual amps.
      reclosers\_write\_phase\_trip(argument) \rightarrow float
           Sets the phase trip curve multiplier or actual amps.
class py dss interface.models.Reclosers.Reclosers.ReclosersI(obj dss)
      Bases: py_dss_interface.models.Base.Base
      This interface can be used to read/write certain properties of the active DSS object.
      The structure of the interface is as follows: int32_t ReclosersI(int32_t Parameter, int32_t Argument);
      This interface returns an integer with the result of the query according to the value of the variable Parameter,
      which can be one of the following.
      reclosers\_close() \rightarrow int
           Close the switched object controlled by the recloser. Resets recloser to first operation.
      \textbf{reclosers\_count}\;(\,)\;\rightarrow int
           Gets number of Reclosers in active circuit.
      reclosers first () \rightarrow int
           Sets first recloser to be active Circuit Element. Returns 0 if none.
      reclosers next() \rightarrow int
           Sets next recloser to be active Circuit Element. Returns 0 if none.
      reclosers_open() \rightarrow int
           Open recloser's controlled element and lock out the recloser.
      reclosers\_read\_idx() \rightarrow int
           Gets the active recloser by index into the recloser list. 1..Count.
      reclosers_read_monitored_term() \rightarrow int
           Gets the terminal number of Monitored Object for the Recloser.
      {\tt reclosers\_read\_num\_fast}\;(\,)\;\to int
           Gets the number of fast shots.
      reclosers read shots () \rightarrow int
           Gets the number of shots to lockout (fast + delayed).
      {\tt reclosers\_read\_switched\_term}\,(\,)\,\rightarrow int
           Gets the terminal of the controlled device being switched by the Recloser.
      reclosers_write_idx(argument) → int
           Sets the active recloser by index into the recloser list. 1..Count.
      reclosers\_write\_monitored\_term(argument) \rightarrow int
           Sets the terminal number of Monitored Object for the Recloser.
      reclosers_write_num_fast(argument) → int
           Sets the number of fast shots.
      reclosers\_write\_shots(argument) \rightarrow int
           Sets the number of shots to lockout (fast + delayed).
      reclosers_write_switched_term (argument) \rightarrow int
           Sets the terminal of the controlled device being switched by the Recloser.
```

```
class py_dss_interface.models.Reclosers.Reclosers.ReclosersS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr ReclosersS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
reclosers_read_monitored_obj()
```

Gets the full name of object this Recloser is monitoring.

```
reclosers_read_name()
```

Gets the name of the active Recloser Object.

```
reclosers_read_switched_obj()
```

Gets the full name of the circuit element that is being switched by this Recloser.

```
reclosers_write_monitored_obj(argument)
```

Sets the full name of object this Recloser is monitoring.

```
reclosers_write_name (argument)
```

Sets the name of the active Recloser Object.

```
reclosers_write_switched_obj(argument)
```

Sets the full name of the circuit element that is being switched by this Recloser.

```
class py_dss_interface.models.Reclosers.Reclosers.Reclosers(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void Reclosers V(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
reclosers_all_names()
```

Gets a variant array of strings with names of all reclosers in active circuit.

```
reclosers_reclose_intervals()
```

Gets a variant array of doubles: reclose intervals (s) between shots.

# 4.1.29 RegControls

Created by eniocc at 11/10/2020

This interface implements the RegControls (IRegControls) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: RegControlsI, RegControlsF, RegControlsV, RegControlsS.

```
class py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.Bases:
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double RegControlsF(int32\_t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

# $\verb"regcontrols_read_ct_primary"() \to float$

Gets the CT primary ampere rating (secondary is 0.2 amperes).

## $regcontrols\_read\_delay() \rightarrow float$

Gets the time delay [s] after arming before the first tap change. Control may reset before actually changing taps.

## $regcontrols\_read\_forward\_band() \rightarrow float$

Gets the regulation bandwidth in forward direction, centered on Vreg.

### regcontrols\_read\_forward\_r() $\rightarrow$ float

Gets the LDC R settings in Volts.

## $regcontrols\_read\_forward\_vreg() \rightarrow float$

Gets the target voltage in the forward direction, on PT secondary base.

## $regcontrols\_read\_forward\_x() \rightarrow float$

Gets the LDC X settings in Volts.

## $regcontrols\_read\_pt\_ratio() \rightarrow float$

Gets the PT ratio for voltage control settings.

#### $regcontrols\_read\_reverse\_band() \rightarrow float$

Gets the bandwidth in reverse direction, centered on reverse Vreg.

# $regcontrols\_read\_reverse\_r() \rightarrow float$

Gets the reverse LDC R settings in Volts.

#### regcontrols\_read\_reverse\_vreg() $\rightarrow$ float

Gets the target voltage in the reverse direction, on PT secondary base.

## $regcontrols\_read\_reverse\_x() \rightarrow float$

Gets the reverse LDC X settings in Volts.

## $regcontrols\_read\_tap\_delay() \rightarrow float$

Gets the time delay [s] for subsequent tap changes in a set. Control may reset before actually changing taps.

## ${\tt regcontrols\_read\_voltage\_limit} \; (\;) \; \rightarrow float$

Gets the first house voltage limit on PT secondary base. Setting to 0 disables this function.

# regcontrols\_write\_ct\_primary(argument) → float

Sets the CT primary ampere rating (secondary is 0.2 amperes).

#### regcontrols\_write\_delay(argument) → float

Sets the time delay [s] after arming before the first tap change. Control may reset before actually changing taps.

# $\verb"regcontrols_write_forward_band" (\textit{argument}) \rightarrow float$

Sets the regulation bandwidth in forward direction, centered on Vreg.

#### regcontrols\_write\_forward\_r (argument) → float

Sets the LDC R settings in Volts.

## $regcontrols\_write\_forward\_vreg(argument) \rightarrow float$

Sets the target voltage in the forward direction, on PT secondary base.

# $\verb|regcontrols_write_forward_x| (argument) \rightarrow \verb|float|$

Sets sets the LDC X settings in Volts.

#### regcontrols\_write\_pt\_ratio (argument) → float

Sets the PT ratio for voltage control settings.

## $regcontrols\_write\_reverse\_band(argument) \rightarrow float$

Sets the bandwidth in reverse direction, centered on reverse Vreg.

#### regcontrols\_write\_reverse\_r (argument) → float

Sets the reverse LDC R settings in Volts.

#### regcontrols\_write\_reverse\_vreg(argument) → float

Sets the target voltage in the reverse direction, on PT secondary base.

## regcontrols\_write\_reverse\_x (argument) → float

Sets the reverse LDC X settings in Volts.

#### regcontrols\_write\_tap\_delay(argument) → float

Sets the time delay [s] for subsequent tap changes in a set. Control may reset before actually changing taps.

## regcontrols\_write\_voltage\_limit(argument) → float

Sets the first house voltage limit on PT secondary base. Setting to 0 disables this function.

```
class py_dss_interface.models.RegControls.RegControls.RegControlsI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t RegControlsI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

## $regcontrols\_count() \rightarrow int$

Gets the number of RegControl objects in Active Circuit.

# $\textbf{regcontrols\_first} \; () \; \rightarrow int$

Sets the first RegControl active. Returns 0 if no more.

#### $regcontrols_next() \rightarrow int$

Sets the next RegControl active. Returns 0 if no more

#### regcontrols\_read\_is\_inverse\_time() $\rightarrow$ int

Gets the inverse time feature. Time delay is inversely adjusted, proportional to the amount of voltage outside the regulator band.

# $\textbf{regcontrols\_read\_is\_reversible}\,(\,)\,\,\rightarrow int$

Gets the setting in the reverse direction, usually not applicable to substation transformers.

#### $regcontrols\_read\_max\_tap\_change() \rightarrow int$

Gets the maximum tap change per iteration in STATIC solution mode. 1 is more realistic, 16 is the default for faster solution.

# $\textbf{regcontrols\_read\_tap\_number} \; (\;) \; \rightarrow int$

Gets the actual tap number of the active RegControl.

#### regcontrols\_read\_tap\_winding() $\rightarrow$ int

Gets the tapped winding number.

## $regcontrols\_read\_winding() \rightarrow int$

Gets the winding number for PT and CT connections.

# $regcontrols\_write\_is\_inverse\_time(argument) \rightarrow int$

Sets the inverse time feature. Time delay is inversely adjusted, proportional to the amount of voltage outside the regulator band.

```
regcontrols_write_is_reversible (argument) → int
```

Sets the different settings for the reverse direction (see Manual for details), usually not applicable to substation transformers.

```
regcontrols_write_max_tap_change (argument) \rightarrow int
```

Sets the maximum tap change per iteration in STATIC solution mode. 1 is more realistic, 16 is the default for faster solution.

```
regcontrols_write_tap_number(argument) → int
```

Sets the actual tap number of the active RegControl.

```
regcontrols\_write\_tap\_winding(argument) \rightarrow int
```

Sets the tapped winding number.

```
regcontrols\_write\_winding(argument) \rightarrow int
```

Sets the winding number for PT and CT connections.

```
class py_dss_interface.models.RegControls.RegControls.RegControlsS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr RegControlsS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
regcontrols_read_monitored_bus()
```

Gets the name of the remote regulated bus, in lieu of LDC settings.

```
regcontrols_read_name()
```

Gets the active RegControl name.

```
regcontrols_read_transformer()
```

Gets the name of the transformer this regulator controls.

```
regcontrols_write_monitored_bus (argument)
```

Sets the name of the remote regulated bus, in lieu of LDC settings.

```
regcontrols_write_name (argument)
```

Sets the active RegControl name.

```
regcontrols_write_transformer(argument)
```

Sets the name of the transformer this regulator controls.

```
class py_dss_interface.models.RegControls.RegControls.RegControls(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void RegControlsV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
regcontrols_all_names()
```

Gets a variant array of strings containing all RegControl names.

# 4.1.30 Relays

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.Relays.Relays.Relays(obj_dss)
```

Bases: py\_dss\_interface.models.RelaysS.RelaysS, py\_dss\_interface.models.RelaysV.RelaysV, py\_dss\_interface.models.RelaysI.RelaysI

This interface implements the Relays (IRelays) interface of OpenDSS by declaring 3 procedures for accessing the different properties included in this interface: RelaysV, RelaysI.

```
class py_dss_interface.models.Relays.Relays.RelaysI(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t RelaysI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
relays\_count() \rightarrow int
```

Gets number of Relays in active circuit.

```
relays first() \rightarrow int
```

Sets first relay active. If none, returns 0.

```
relays\_next() \rightarrow int
```

Sets next relay active. If none, returns 0.

```
relays\_read\_idx() \rightarrow int
```

Gets the active relay by index into the Relay list. 1..Count.

```
relays\_read\_monitored\_term() \rightarrow int
```

Gets the number of terminal of monitored element that this relay is monitoring.

```
{\tt relays\_read\_switched\_term}\,(\,)\,\to int
```

Gets the number of terminal of the switched object that will be opened when the relay trips.

```
relays\_write\_idx(argument) \rightarrow int
```

Sets the active relay by index into the Relay list. 1.. Count.

```
relays\_write\_monitored\_term(argument) \rightarrow int
```

Sets the number of terminal of monitored element that this relay is monitoring.

```
\verb"relays_write_switched_term" (\textit{argument}) \rightarrow int
```

Sets the number of terminal of the switched object that will be opened when the relay trips.

```
class py_dss_interface.models.Relays.Relays.RelaysS(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr RelaysS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
relays_read_monitored_obj()
```

Gets the full name of the object this relay is monitoring.

```
relays_read_name()
```

Gets the name of the active Relay.

```
relays_read_switched_obj()
```

Gets the full name of element that will switched when relay trips.

```
relays_write_monitored_obj(argument)
```

Sets the full name of the object this relay is monitoring.

```
relays_write_name (argument)
```

Sets the name of the active Relay.

```
relays_write_switched_obj(argument)
```

Sets the full name of element that will switched when relay trips.

```
class py_dss_interface.models.Relays.Relays.RelaysV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void Relays V(int32 t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
relays_all_names()
```

Getsa variant array of strings containing names of all relay elements.

#### 4.1.31 Sensors

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.Sensors.Sensors.Sensors(obj_dss)
```

```
Bases: py_dss_interface.models.Sensors.SensorsV.SensorsV, py_dss_interface.models.Sensors.SensorsS.SensorsS, py_dss_interface.models.SensorsI.SensorsI, py_dss_interface.models.SensorsF.SensorsF
```

This interface implements the Sensors (ISensors) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: SensorsV, SensorsS, SensorsI, SensorsF.

```
class py_dss_interface.models.Sensors.Sensors.SensorsF(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double SensorsF(int32\_t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
sensors\_read\_kv\_base() \rightarrow float
```

Gets the voltage base for the sensor measurements. LL for 2 and 3 - phase sensors, LN for 1-phase sensors.

```
sensors\_read\_pct\_error() \rightarrow float
```

Gets the assumed percent error in the Sensor measurement. Default is 1.

```
sensors\_read\_weight() \rightarrow float
```

Gets the weighting factor for this sensor measurement with respect to the other sensors. Default is 1.

```
sensors\_write\_kv\_base(argument) \rightarrow float
```

Sets the voltage base for the sensor measurements. LL for 2 and 3 - phase sensors, LN for 1-phase sensors.

```
sensors_write_pct_error(argument) → float
```

Sets the assumed percent error in the Sensor measurement. Default is 1.

```
sensors\_write\_weight(argument) \rightarrow float
```

Sets the weighting factor for this sensor measurement with respect to the other sensors. Default is 1.

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t SensorsI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
sensors\_count() \rightarrow int
```

Gets number of sensors in active circuit.

```
sensors\_first() \rightarrow int
```

Sets the first sensor active. Returns 0 if none.

```
sensors_next() \rightarrow int
```

Sets the next sensor active. Returns 0 if none.

```
sensors\_read\_is\_delta() \rightarrow int
```

Returns 1 if the sensor is connected in delta; otherwise, returns 0.

```
\verb|sensors_read_metered_terminal|()| \rightarrow int
```

Gets the number of the measured terminal in the measured element.

```
sensors\_read\_reverse\_delta() \rightarrow int
```

Returns 1 if voltage measurements are 1-3, 3-2, 2-1; otherwise 0.

```
sensors\_reset() \rightarrow int
```

Clears the active sensor.

```
sensors\_reset\_all() \rightarrow int
```

Clears all sensors in the active circuit.

```
sensors_write_is_delta(argument) → int
```

Allows to set 1 if the sensor is connected in delta; otherwise, set 0 (argument).

```
sensors\_write\_metered\_terminal(argument) \rightarrow int
```

Sets the number of the measured terminal in the measured element.

```
sensors\_write\_reverse\_delta(argument) \rightarrow int
```

Allows to set 1 if voltage measurements are 1-3, 3-2, 2-1; otherwise 0.

```
class py_dss_interface.models.Sensors.Sensors.SensorsS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr SensorsS(int32\_t Parameter, CStr Argument); This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
sensors_read_metered_element()
```

Gets the full name of the measured element.

```
\verb"sensors_read_name" (\ )
```

Gets the name of the active sensor object.

```
sensors_write_metered_element (argument)
```

Sets the full name of the measured element.

```
sensors_write_name (argument)
```

Sets the name of the active sensor object.

```
class py_dss_interface.models.Sensors.Sensors.SensorsV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void SensorsV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
sensors_all_names()
```

Returns a variant array of sensor names.

```
sensors_read_currents()
```

Gets an array of doubles for the line current measurements; don't use with KWS and KVARS.

```
sensors_read_kvars()
```

Gets an array of doubles for Q measurements; overwrites currents with a new estimate using KWS.

```
sensors_read_kws()
```

Gets an array of doubles for P measurements; overwrites currents with a new estimate using KVARS.

```
sensors_write_currents (argument)
```

Sets an array of doubles for the line current measurements; don't use with KWS and KVARS.

```
sensors_write_kvars(argument)
```

Sets an array of doubles for Q measurements; overwrites currents with a new estimate using KWS.

```
sensors_write_kws (argument)
```

Sets an array of doubles for P measurements; overwrites currents with a new estimate using KVARS.

#### 4.1.32 Settings

Created by eniocc at 11/10/2020

This interface implements the Settings (ISettings) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: SettingsS, SettingsF, SettingsI, SettingsV.

```
class py_dss_interface.models.Settings.Settings.SettingsF(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double SettingsF(int32\_t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
\verb|settings_allocation_factors|() \rightarrow \verb|float||
```

Sets all load allocation factors for all loads defined by XFKVA property to this value.

```
\verb|settings_read_emerg_vmax_pu|() \rightarrow float
```

Gets the per unit maximum voltage for Emergency conditions.

```
settings_read_emerg_vmin_pu() \rightarrow float
```

Gets the per unit minimum voltage for Emergency conditions.

```
settings read loss weight () \rightarrow float
           Gets the weighting factor applied to Loss register values.
      \verb|settings_read_norm_vmax_pu|() \rightarrow \verb|float|
           Gets the per unit maximum voltage for Normal conditions.
      settings\_read\_norm\_vmin\_pu() \rightarrow float
           Gets the per unit minimum voltage for Normal conditions.
      settings\_read\_price\_signal() \rightarrow float
           Gets the price signal for the circuit.
      settings\_read\_ue\_weight() \rightarrow float
           Gets the weighting factor applied to UE register values.
      settings_write_emerg_vmax_pu (argument) → float
           Sets the per unit maximum voltage for Emergency conditions.
      settings_write_emerg_vmin_pu (argument) → float
           Sets the per unit minimum voltage for Emergency conditions.
      settings write loss weight (argument) \rightarrow float
           Sets the weighting factor applied to Loss register values.
      settings\_write\_norm\_vmax\_pu (argument) \rightarrow float
           Sets the per unit maximum voltage for Normal conditions.
      settings write norm vmin pu (argument) \rightarrow float
           Sets the per unit minimum voltage for Normal conditions.
      settings\_write\_price\_signal(argument) \rightarrow float
           Sets the price signal for the circuit.
      settings\_write\_ue\_weight(argument) \rightarrow float
           Sets the weighting factor applied to UE register values.
class py_dss_interface.models.Settings.Settings.SettingsI(obj_dss)
      Bases: py_dss_interface.models.Base.Base
      This interface can be used to read/write certain properties of the active DSS object.
      The structure of the interface is as follows: int32_t SettingsI(int32_t Parameter, int32_t Argument);
      This interface returns an integer with the result of the query according to the value of the variable Parameter,
      which can be one of the following.
      \verb|settings_read_allow_duplicates|()| \rightarrow int
           Gets if OpenDSS allows duplicate names of objects: {1 allow, 0 not allow}.
      settings\_read\_ckt\_model() \rightarrow int
           Gets {dssMultiphase* | dssPositiveSeq} Indicate if the circuit model is positive sequence.
      \textbf{settings\_read\_trapezoidal} \; () \; \rightarrow int
           Gets {True (1) | False (0)} value of trapezoidal integration flag in Energy Meters.
      settings\_read\_zone\_lock() \rightarrow int
           Gets the status of Lock zones on energy meters to prevent rebuilding if a circuit change occurs: {1= true,
           0 = \text{False}.
      settings_write_allow_duplicates (argument) → int
           Sets if OpenDSS allows duplicate names of objects: {1 allow, 0 not allow}.
      settings write ckt model (argument) \rightarrow int
           Sets {dssMultiphase* | dssPositiveSeq} Indicate if the circuit model is positive sequence.
```

```
settings write trapezoidal (argument) \rightarrow int
```

Sets {True (1) | False (0)} value of trapezoidal integration flag in Energy Meters.

```
settings\_write\_zone\_lock(argument) \rightarrow int
```

Sets the status of Lock zones on energy meters to prevent rebuilding if a circuit change occurs: {1= true, 0= False}.

```
{\tt class} \ {\tt py\_dss\_interface.models.Settings.Settings.Settings} \ (obj\_dss)
```

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr SettingsS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
settings_read_auto_bus_list()
```

Gets the list of Buses or (File=xxxxx) syntax for the AutoAdd solution mode.

```
settings_read_price_curve()
```

Gets the name of LoadShape object that serves as the source of price signal data for yearly simulations, etc.

```
settings_write_auto_bus_list(argument)
```

Sets the list of Buses or (File=xxxxx) syntax for the AutoAdd solution mode.

```
settings_write_price_curve (argument)
```

Sets the name of LoadShape object that serves as the source of price signal data for yearly simulations, etc.

```
class py_dss_interface.models.Settings.Settings.SettingsV(obj_dss)
Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void Settings V(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
settings_read_loss_regs()
```

Gets the array of Integers defining Energy Meter registers to use for computing Losses.

```
settings_read_ue_regs()
```

Gets the array of Integers defining Energy Meter registers to use for computing UE.

```
settings_read_voltage_bases()
```

Gets the array of doubles defining the legal voltage bases in kV L-L.

```
settings_write_loss_regs(argument)
```

Sets the array of Integers defining Energy Meter registers to use for computing Losses.

```
settings_write_ue_regs (argument)
```

Sets the array of Integers defining Energy Meter registers to use for computing UE.

```
settings_write_voltage_bases(argument)
```

Sets the array of doubles defining the legal voltage bases in kV L-L.

#### 4.1.33 Solution

Created by eniocc at 11/10/2020

#### **class** py\_dss\_interface.models.Solution.Solution.**Solution**(obj\_dss)

Bases: py\_dss\_interface.models.Solution.SolutionI.SolutionI, py\_dss\_interface.models.Solution.SolutionF.SolutionF, py\_dss\_interface.models.SolutionSolutionS.SolutionS

This interface implements the Solution (ISolution) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: SolutionI, SolutionF, SolutionS, SolutionV.

```
class py_dss_interface.models.Solution.Solution.SolutionF(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double SolutionF(int32\_t Parameter, double Argument);

This interface returns a floating point number according to the number sent in the variable "parameter". The parameter can be one of the following.

```
solution\_process\_time() \rightarrow float
```

Retrieves the time required (microseconds) to perform the latest solution time step, this time does not include the time required for sampling meters/monitors.

```
solution\_process\_time\_step() \rightarrow float
```

Retrieves the time required (microseconds) to perform the latest solution time step including the time required for sampling meters/monitors.

```
solution_read_cap_kvar() → float
```

Returns the capacitor kvar for adding in AutoAdd mode.

```
solution\_read\_dbl\_hour() \rightarrow float
```

Returns the hour as a double, including fractional part.

```
solution\_read\_frequency() \rightarrow float
```

Returns the frequency for the next solution.

```
solution\_read\_gen\_kw() \rightarrow float
```

Returns the generator kW for AutoAdd mode.

```
solution\_read\_gen\_mult() \rightarrow float
```

Returns the default multiplier applied to generators (like LoadMult).

```
solution\_read\_gen\_pf() \rightarrow float
```

Returns the pf for generators in AutoAdd mode.

```
solution\_read\_load\_mult() \rightarrow float
```

Returns the default load multiplier applied to all non-fixed loads.

```
solution\_read\_pct\_growth() \rightarrow float
```

Returns the percent default annual load growth rate.

```
solution\_read\_seconds() \rightarrow float
```

Returns the seconds from top of the hour.

```
solution\_read\_step\_size() \rightarrow float
```

Returns the step size for the next solution.

```
solution\_read\_tolerance() \rightarrow float
```

Returns the solution convergence tolerance.

```
solution\_read\_total\_time() \rightarrow float
```

Retrieves the accumulated time required (microseconds) to perform the simulation.

```
solution\_step\_size\_hr() \rightarrow float
     Sets the step size in Hours.
solution\_step\_size\_min() \rightarrow float
     Sets the step size in minutes.
solution_write_cap_kvar(argument) → float
     Sets the capacitor kvar for adding in AutoAdd mode.
solution write dbl hour (argument) \rightarrow float
     Sets the hour as a double, including fractional part.
solution\_write\_frequency(argument) \rightarrow float
     Sets the frequency for the next solution.
solution_write_gen_kw (argument) → float
     Sets the generator kW for AutoAdd mode.
solution\_write\_gen\_mult(argument) \rightarrow float
     Sets the default multiplier applied to generators (like LoadMult).
\verb|solution_write_gen_pf| (argument) \rightarrow \verb|float|
     Sets the pf for generators in AutoAdd mode.
solution\_write\_load\_mult(argument) \rightarrow float
     Sets the default load multiplier applied to all non-fixed loads.
solution write pct growth (argument) \rightarrow float
     Sets the percent default annual load growth rate.
solution\_write\_seconds(argument) \rightarrow float
     Sets the seconds from top of the hour.
solution_write_step_size(argument) → float
     Sets the step size for the next solution.
solution_write_tolerance(argument) → float
     Sets the solution convergence tolerance.
solution\_write\_total\_time(argument) \rightarrow float
     Sets the accumulated time (microseconds) register. The new value for this register must be specified in the
     argument.
```

class py\_dss\_interface.models.Solution.Solution.SolutionI(obj\_dss)
 Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32 t SolutionI(int32 t Parameter, int32 t Argument);

This interface returns an integer according to the number sent in the variable "parameter". The parameter can be one of the following.

```
solution\_build\_y\_matrix() \rightarrow int
```

Forces building of the System Y matrix according to the argument:  $\{1 = \text{ series elements only } | 2 = \text{ Whole Y matrix}\}$ .

#### solution\_calc\_inc\_matrix() → int

Starts the calculation of the incidence matrix for the active actor. Please be sure that the circuits of each actor have been compiled and ready to be solved before using this command.

#### $\verb"solution_calc_inc_matrix_0"\ (\ ) \ \to int$

Starts the calculation of the Branch to Node incidence matrix for the active actor. Please be sure that the circuits of each actor have been compiled and ready to be solved before using this command. The

difference between this command and the CalcIncMatrix is that the calculated matrix will be ordered hierarchically from the substation to the feeder end, which can be helpful for many operations. Additionally, the Bus Levels vector is calculated and the rows (PDElements) and columns (Buses) are permuted so it is easy to identify their position in the circuit.

#### $\verb|solution_check_controls|()| \rightarrow int$

Performs the normal process for sampling and executing Control Actions and Fault Status and rebuilds Y if necessary.

#### $\verb|solution_check_fault_status|()| \rightarrow int$

Executes status check on all fault objects defined in the circuit. Returns 0.

#### $solution\_clean\_up() \rightarrow int$

Update storage, invcontrol, etc., at end of time step.

#### $solution_do_control_actions() \rightarrow int$

Pops control actions off the control queue and dispatches to the proper control element.

#### $\verb|solution_finish_time_step|()| \rightarrow int$

Calls cleanup, sample monitors, and increment time at end of time step.

#### $solution\_init\_snap() \rightarrow int$

Initializes some variables for snap shot power flow. SolveSnap does this automatically.

#### $solution\_iterations() \rightarrow int$

Returns the number of iterations taken for the last solution.

#### $solution_most_iterations_done() \rightarrow int$

Returns the max number of iterations required to converge at any control iteration of the most recent solution..

#### $solution\_read\_add\_type() \rightarrow int$

Returns the type of device to add in AutoAdd Mode: {dssGen (default)|dssCap}.

#### $\textbf{solution\_read\_algorithm}\,(\,)\,\to int$

Returns the base solution algorithm: {dssNormalSolve | dssNewtonSolve}.

#### $solution\_read\_control\_actions\_done() \rightarrow int$

Indicates that the control actions are done: {1 done, 0 not done}.

#### $solution\_read\_control\_iterations() \rightarrow int$

Returns the current value of the control iteration counter.

#### $\textbf{solution\_read\_control\_mode} \; (\;) \; \rightarrow int$

Returns the mode for control devices: {dssStatic (default) | dssEvent | dssTime}.

#### $solution\_read\_converged() \rightarrow int$

Indicates whether the circuit solution converged (1 converged | 0 not converged).

#### $solution\_read\_hour() \rightarrow int$

Returns the present hour (See DSS help).

#### $solution\_read\_load\_model() \rightarrow int$

Returns the Load Model: {dssPowerFlow (default)|dssAdmittance}.

#### $solution\_read\_max\_control\_iterations() \rightarrow int$

Returns the maximum allowable control iterations.

#### $\verb|solution_read_max_iterations|()| \rightarrow int$

Returns the Maximum number of iterations used to solve the circuit.

#### solution read mode() $\rightarrow$ int

Returns the present solution mode (See DSS help).

#### solution read number () $\rightarrow$ int

Returns the number of solutions to perform for MonteCarlo and time series simulations.

#### $solution\_read\_random() \rightarrow int$

Returns the randomization mode for random variables "Gaussian" o "Uniform".

#### $solution\_read\_year() \rightarrow int$

Returns the present Year (See DSS help).

#### solution sample control devices () $\rightarrow$ int

Executes a sampling of all intrinsic control devices, which push control actions into the control queue.

#### $\verb|solution_sample_do_control_actions|()| \rightarrow int$

Sample controls and then process the control queue for present control mode and dispatch control actions. Returns 0.

#### solution solve() $\rightarrow$ int

Solution for the present solution mode. Returns 0.

#### $solution\_solve\_all() \rightarrow int$

Starts the solution process for all the actors created in memory. Please be sure that the circuits of each actor have been compiled and ready to be solved before using this command.

#### $solution\_solve\_direct() \rightarrow int$

Executes a direct solution from the system Y matrix, ignoring compensation currents of loads, generators (includes Yprim only).

#### solution solve no control() $\rightarrow$ int

Is similar to SolveSnap except no control actions are checked or executed.

#### $\verb|solution_solve_plus_control|()| \rightarrow int$

Executes a power flow solution (SolveNoControl) plus executes a CheckControlActions that executes any pending control actions.

#### $\verb"solution_solve_power_flow"() \rightarrow int$

Solves using present power flow method. Iterative solution rather than direct solution.

#### $solution\_system\_y\_changed() \rightarrow int$

Indicates if elements of the System Y have been changed by recent activity. If changed returns 1; otherwise 0.

#### $solution\_total\_iterations() \rightarrow int$

Returns the total iterations including control iterations for most recent solution.

#### $solution\_write\_add\_type (argument) \rightarrow int$

Modifies the type of device to add in AutoAdd Mode: {dssGen (default)|dssCap}.

#### solution write algorithm (argument) $\rightarrow$ int

Modifies the base solution algorithm: {dssNormalSolve | dssNewtonSolve}.

#### $solution\_write\_control\_actions\_done(argument) \rightarrow int$

Modifies the flag to indicate that the control actions are done: {1 done, 0 not done}.

#### $solution\_write\_control\_iterations(argument) \rightarrow int$

Modifies the current value of the control iteration counter.

#### $solution\_write\_control\_mode(argument) \rightarrow int$

Modifies the mode for control devices: {dssStatic (default) | dssEvent | dssTime}.

#### $solution\_write\_converged(argument) \rightarrow int$

Modifies the converged flag (1 converged | 0 not converged).

```
solution write hour (argument) \rightarrow int
          Modifies the present hour (See DSS help).
     solution\_write\_load\_model(argument) \rightarrow int
          Modifies the Load Model: {dssPowerFlow (default)|dssAdmittance}.
     solution write max control iterations (argument) → int
          Modifies the maximum allowable control iterations.
     solution write max iterations (argument) \rightarrow int
          Modifies the Maximum number of iterations used to solve the circuit.
     solution\_write\_mode(argument) \rightarrow int
          Modifies the present solution mode (See DSS help).
     solution_write_number (argument) → int
          Modifies the number of solutions to perform for MonteCarlo and time series simulations.
     solution\_write\_random(argument) \rightarrow int
          Modifies the randomization mode for random variables "Gaussian" o "Uniform".
     solution write year (argument) \rightarrow int
          Modifies the present Year (See DSS help).
class py_dss_interface.models.Solution.Solution.SolutionS(obj_dss)
     Bases: py_dss_interface.models.Base.Base
     This interface can be used to read/write certain properties of the active DSS object.
     The structure of the interface is as follows: CStr SolutionS(int32_t Parameter, CStr Argument);
     This interface returns a string according to the number sent in the variable "parameter". The parameter can be
     one of the following.
     solution_mode_id()
          Returns the ID (text) of the present solution mode.
     solution_read_default_daily()
          Returns the default daily load shape (defaults to "Default").
     solution_read_default_yearly()
          Returns the default yearly load shape (defaults to "Default").
     solution read ld curve()
          Returns the Load-Duration Curve name for LD modes.
     solution_write_default_daily (argument)
          Sets the default daily load shape (defaults to "Default").
     solution_write_default_yearly (argument)
          Sets the default yearly load shape (defaults to "Default").
     solution_write_ld_curve (argument)
          Sets the Load-Duration Curve name for LD modes.
class py_dss_interface.models.Solution.Solution.SolutionV(obj_dss)
     Bases: py_dss_interface.models.Base.Base
     This interface can be used to read/write certain properties of the active DSS object.
```

The structure of the interface is as follows: void SolutionV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a variant according to the number sent in the variable "parameter". The parameter can be one of the following.

#### solution bus levels()

Returns an array of integers containing BusLevels array. This array gives a numeric value to each bus to specify how far it is from the circuit?s backbone (a continuous path from the feeder head to the feeder end). It is very handy to understand the circuit?s topology.

#### solution\_event\_log()

Returns an array of strings containing the Event Log.

#### solution\_inc\_matrix\_cols()

Returns an array of strings specifying the way the cols of the incidence matrix (buses) are organized, depending on the way the Branch to node incidence matrix was calculated (CalcIncMatrix/CalcIncMatrix\_O) the result could be very different.

#### solution\_inc\_matrix\_rows()

Returns an array of strings specifying the way the rows of the incidence matrix (PDElements) are organized, depending on the way the Branch to node incidence matrix was calculated (CalcIncMatrix/CalcIncMatrix\_O) the result could be very different..

#### solution\_laplacian()

Returns an array of integers containing the Laplacian matrix using the incidence matrix previously calculated, this means that before calling this command the incidence matrix needs to be calculated using calcincmatrix/calcincmatrix\_o. This command will return only the non-zero values in compressed coordinate format (row, col, value)..

#### solution\_nc\_matrix()

Returns an array of integers containing the incidence matrix (1-D). Each cell of the incidence matrix is delivered using 3 elements of the array delivered, the first is the row, the second is the column and the third is the value (1/-1). This procedure will only deliver the non-zero elements..

#### 4.1.34 SwtControls

Created by eniocc at 11/10/2020

This interface implements the SwtControls (ISwtControls) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: SwtControlsS, SwtControlsV, SwtControlsI, SwtControlsF.

```
class py_dss_interface.models.SwtControls.SwtControls.SwtControls(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double SwtControlsF(int32\_t Parameter, double Argument);

This interface returns a floating point number (64 bits) with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
swtcontrols\_read\_delay() \rightarrow float
```

Gets the time delay [s] between arming and opening or closing the switch. Control may reset before actually operating the switch.

```
swtcontrols\_write\_delay(argument) \rightarrow float
```

Sets sets the time delay [s] between arming and opening or closing the switch. Control may reset before actually operating the switch.

```
class py_dss_interface.models.SwtControls.SwtControls.SwtControlsI(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t SwtControlsI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
swtcontrols\_count() \rightarrow int
```

Gets the total number of SwtControls in the active circuit.

```
swtcontrols\_first() \rightarrow int
```

Sets the first SwtControl active. Returns 0 if no more.

```
swtcontrols_next() → int
```

Sets the next SwtControl active. Returns 0 if no more.

```
swtcontrols\_read\_action() \rightarrow int
```

Gets the open (1) or close (2) action of the switch. No effect if switch is locked. However, reset removes any lock and then closes the switch (shelf state). 0 = none action.

```
{\tt swtcontrols\_read\_is\_locked}\,(\,)\,\rightarrow int
```

Gets the lock state: {1 locked | 0 not locked}.

```
swtcontrols\_read\_switched\_term() \rightarrow int
```

Gets the terminal number where the switch is located on the SwitchedObj.

```
swtcontrols\_write\_action(argument) \rightarrow int
```

Sets open (1) or close (2) the switch. No effect if switch is locked. However, reset removes any lock and then closes the switch (shelf state). 0 = none action (see manual for details).

```
swtcontrols\_write\_is\_locked(argument) \rightarrow int
```

Sets the lock to prevent both manual and automatic switch operation.

```
swtcontrols\_write\_switched\_term(argument) \rightarrow int
```

Sets the terminal number where the switch is located on the SwitchedObj.

```
class py_dss_interface.models.SwtControls.SwtControls.SwtControls(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr SwtControlsS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
swtcontrols read name()
```

Gets the active swtcontrol name.

```
swtcontrols_read_switched_obj()
```

Gets the name of the switched object by the active SwtControl

```
swtcontrols_write_name (argument)
```

Sets the active swtcontrol by name.

```
swtcontrols_write_switched_obj(argument)
```

Sets the switched object by name.

```
 \textbf{class} \  \, \texttt{py\_dss\_interface.models.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void SwtControlsV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
swtcontrols_all_names()
```

Gets a variant array of strings with all SwtControl names in the active circuit.

#### 4.1.35 Text

Created by eniocc at 11/10/2020

```
class py_dss_interface.models.Text.Text.Text(obj_dss)
    Bases: py_dss_interface.models.Base.Base
    text(argument)
```

Can be used to send commands to the text interface of OpenDSS (DSS.Text).

#### 4.1.36 Topology

Created by eniocc at 11/10/2020

This interface implements the Topology (ITopology) interface of OpenDSS by declaring 3 procedures for accessing the different properties included in this interface: TopologyI, TopologyV, TopologyS.

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32 t TopologyI(int32 t Parameter, int32 t Argument);

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
{\tt topology\_active\_branch}\,(\,)\,\to int
```

Returns the index of the active Branch.

```
topology_active_level() \rightarrow int
```

Gets the topological depth of the active branch.

```
\verb"topology_backward_branch"() \rightarrow int
```

Moves back toward the source, return index of new active branch or 0 if no more.

```
\textbf{topology\_first}\,(\,)\,\rightarrow int
```

Sets the first branch active, returns 0 if none.

```
topology_first_load() \rightarrow int
```

Sets as active load the first load at the active branch, return index or 0 if none.

```
topology_forward_branch() \rightarrow int
```

Moves forward in the tree, return index of new active branch or 0 if no more.

```
{\tt topology\_looped\_branch}\,(\,)\,\to int
```

Moves to looped branch, return index or 0 if none.

```
topology_next() → int
Sets the next branch active, returns 0 if none.

topology_next_load() → int
Sets as active load the next load at the active branch, return index or 0 if none.

topology_num_isolated_branches() → int
Gets the number of isolated branches (PD elements and capacitors).

topology_num_isolated_loadss() → int
Gets the number of isolated loads.

topology_num_loops() → int
Gets the number of loops.

topology_parallel_branch() → int
Mode to directly parallel branch, return index or 0 if none.
```

 ${\tt class} \ {\tt py\_dss\_interface.models.Topology.Topology.TopologyS} \ (obj\_dss)$ 

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr TopologyS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
topology_read_branch_name()
```

Gets the name of the active branch.

```
topology_read_bus_name()
```

Gets the name of the active Bus.

```
topology_write_branch_name (argument)
```

Sets the name of the active branch.

```
topology_write_bus_name (argument)
```

Sets the Bus active by name.

```
class py_dss_interface.models.Topology.Topology(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void TopologyV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
topology_all_isolated_branches()
```

Gets a variant array of all isolated branch names.

```
topology_all_isolated_loads()
```

Gets a variant array of all isolated load names.

```
topology_all_looped_pairs()
```

Gets a variant array of all looped element names, by pairs.

#### 4.1.37 Transformers

Created by eniocc at 11/10/2020

This interface implements the Transformers (ITransformer) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: TransformersV, TransformersF, TransformersI, TransformersS.

```
class py_dss_interface.models.Transformers.Transformers.TransformersF(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the Transformers Class where the values are doubles.

The structure of the interface is as follows: double TransformersF(int32\_t Parameter, double argument);

This interface returns a floating point number (64 bits), the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

```
\texttt{transformers\_read\_kv}\,(\,)\,\to float
```

Gets the active winding kV rating. Phase-phase for 2 or 3 phases, actual winding kV 1 phase transformer.

```
transformers\_read\_kva() \rightarrow float
```

Gets the active winding kVA rating. On winding 1, this also determines normal and emergency current ratings for all windings.

```
\texttt{transformers\_read\_max\_tap}\,(\,)\,\rightarrow float
```

Gets the active winding maximum tap in per-unit.

#### $\texttt{transformers\_read\_min\_tap}\,(\,)\,\rightarrow float$

Gets the active winding minimum tap in per-unit.

#### $transformers\_read\_r() \rightarrow float$

Gets the active winding resistance in %.

```
\texttt{transformers\_read\_r\_neut}\,(\,)\,\rightarrow float
```

Gets the active winding neutral resistance [ohms] for wye connections. Set less than zero ungrounded wye.

#### transformers read tap() $\rightarrow$ float

Gets the active winding tap in per-unit.

```
{\tt transformers\_read\_x\_neut}\,(\,)\,\rightarrow {\rm float}
```

Gets the active winding neutral reactance [ohms] for wye connections.

```
transformers\_read\_xhl() \rightarrow float
```

Gets the percent reactance between windings 1 and 2, on winding 1 kVA base. Use for 2 winding or 3 winding transformers.

```
\texttt{transformers\_read\_xht}\,(\,)\,\to float
```

Gets the percent reactance between windings 1 and 3, on winding 1 kVA base. Use for 3 winding transformers only.

#### $\textbf{transformers\_read\_xlt} \; (\;) \; \rightarrow float$

Gets he percent reactance between windings 2 and 3, on winding 1 kVA base. Use for 3 winding transformers only.

#### $transformers\_write\_kv(argument) \rightarrow float$

Sets the active winding kV rating. Phase-phase for 2 or 3 phases, actual winding kV 1 phase transformer.

#### transformers write $kva(argument) \rightarrow float$

Sets the active winding kVA rating. On winding 1, this also determines normal and emergency current ratings for all windings.

#### $transformers\_write\_max\_tap(argument) \rightarrow float$

Sets the active winding maximum tap in per-unit.

#### transformers\_write\_min\_tap(argument) → float

Sets the active winding minimum tap in per-unit.

#### $transformers\_write\_r(argument) \rightarrow float$

Sets the active winding resistance in %.

#### transformers\_write\_r\_neut (argument) → float

Sets the active winding neutral resistance [ohms] for wye connections. Set less than zero ungrounded wye.

#### $transformers\_write\_tap(argument) \rightarrow float$

Sets the active winding tap in per-unit.

#### $transformers\_write\_x\_neut(argument) \rightarrow float$

Sets the active winding neutral reactance [ohms] for wye connections.

#### transformers\_write\_xhl(argument) → float

Sets the percent reactance between windings 1 and 2, on winding 1 kVA base. Use for 2 winding or 3 winding transformers.

#### transformers\_write\_xht(argument) → float

Sets the percent reactance between windings 1 and 3, on winding 1 kVA base. Use for 3 winding transformers only.

#### $transformers\_write\_xlt(argument) \rightarrow float$

Sets the percent reactance between windings 2 and 3, on winding 1 kVA base. Use for 3 winding transformers only.

This interface can be used to read/modify the properties of the Transformers Class where the values are integers.

```
The structure of the interface is as follows: int32_t TransformersI(int32_t Parameter, int32_t argument);
```

This interface returns an integer (signed 32 bits), the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

#### transformers count () $\rightarrow$ int

Gets the number of transformers within the active circuit.

#### $\textbf{transformers\_first} \; () \; \rightarrow int$

Sets the first Transformer active. Return 0 if no more.

#### $\textbf{transformers\_next} \; (\,) \; \rightarrow int$

Sets the next Transformer active. Return 0 if no more.

#### $transformers\_read\_is\_delta() \rightarrow int$

Gets the information about if the active winding is delta (1) or wye (0) connection.

#### ${\tt transformers\_read\_num\_taps}\,(\,)\,\rightarrow int$

Gets the active winding number of tap steps between MinTap and MaxTap.

#### ${\tt transformers\_read\_num\_windings}\,(\,)\,\rightarrow int$

Gets the number of windings on this transformer. Allocates memory; set or change this property first.

#### transformers read wdg() $\rightarrow$ int

Gets the active winding number from 1..NumWindings. Update this before reading or setting a sequence of winding properties (R, Tap, kV, kVA, etc.).

#### transformers\_write\_is\_delta(argument) → int

Sets the information about if the active winding is delta (1) or wye (0) connection.

#### transformers\_write\_num\_taps(argument) → int

Sets the active winding number of tap steps between MinTap and MaxTap.

#### $transformers\_write\_num\_windings(argument) \rightarrow int$

Sets the number of windings on this transformer. Allocates memory; set or change this property first.

#### transformers\_write\_wdg(argument) → int

Sets the active winding number from 1..NumWindings. Update this before reading or setting a sequence of winding properties (R, Tap, kV, kVA, etc.).

```
class py_dss_interface.models.Transformers.Transformers.TransformersS(obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/modify the properties of the Transformers Class where the values are Strings.

The structure of the interface is as follows: CStr TransformersS(int32\_t Parameter, CStr argument);

This interface returns a string, the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

#### transformers read name()

Gets the active transformer name.

#### transformers\_read\_xfmr\_code()

Gets the name of an XfrmCode that supplies electrical parameters for this transformer.

#### transformers\_str\_wdg\_voltages()

Gets the voltages at the active winding of the active transformer in string format.

#### transformers\_write\_name (argument)

Sets the active transformer by name.

#### transformers\_write\_xfmr\_code (argument)

Sets the name of an XfrmCode that supplies electrical parameters for this transformer.

This interface can be used to read/modify the properties of the Transformers Class where the values are Variants.

```
The structure of the interface is as follows: void TransformersV(int32_t Parameter, VARIANT *Argument);
```

This interface returns a Variant, the variable "parameter" is used to specify the property of the class to be used and the variable "argument" can be used to modify the value of the property when necessary. Reading and writing properties are separated and require a different parameter number to be executed.

The properties (parameter) are integer numbers and are described as follows.

#### transformers\_all\_Names()

Gets a variant array of strings with all Transformer names in the active circuit.

#### transformers wdg currents()

Gets a a variant array of doubles containing the currents at the active winding on the active transformer. These currents come as complex pairs.

#### transformers\_wdg\_voltages()

Gets a variant array of doubles containing the voltages at the active winding on the active transformer. These voltages come as complex pairs.

#### 4.1.38 VSources

Created by eniocc at 11/10/2020

This interface implements the Vsources (IVSources) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: VSourcesS, VSourcesV, VSourcesI, VSourcesF.

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double VSourcesF(int32\_t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
vsources_read_angle_deg() → float
```

```
Gets the source phase angle of first phase in degrees.
```

```
\textbf{vsources\_read\_base\_kv} \ () \rightarrow \text{float}
```

Gets the source voltage in kV.

```
\textbf{vsources\_read\_frequency} \, (\,) \, \to float
```

Gets the source frequency in Hz.

```
vsources read pu() \rightarrow float
```

Gets the source voltage in pu.

```
vsources_write_angle_deg(argument) \rightarrow float
```

Sets the source phase angle of first phase in degrees.

```
\textbf{vsources\_write\_base\_kv} (\textit{argument}) \ \rightarrow float
```

Sets the source voltage in kV.

```
vsources_write_frequency (argument) → float
```

Sets the source frequency in Hz.

```
vsources_write_pu (argument) → float
```

Sets the source voltage in pu.

```
class py_dss_interface.models.VSources.VSources.VSourcesI (obj_dss)

Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32\_t VSourcesI(int32\_t Parameter, int32\_t Argument);

This interface returns an integer number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
\textbf{vsources\_count} \; (\,) \; \rightarrow int
```

Returns the number of VSource objects currently defined in the active circuit.

```
vsources\_first() \rightarrow int
```

Sets the first VSource to be active; returns 0 if none.

```
vsources next() \rightarrow int
```

Sets the next VSource to be active; returns 0 if none.

```
vsources\_read\_phases() \rightarrow int
```

Gets the number of phases of the active VSource.

```
vsources_write_phases (argument) \rightarrow int
```

Sets the number of phases of the active VSource.

```
class py_dss_interface.models.VSources.VSources.VSourcesS(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: CStr VSourcesS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
vsources read name()
```

Gets the name of the active VSource.

```
vsources_write_name (argument)
```

Sets the name of the active VSource.

```
class py_dss_interface.models.VSources.VSources.VSourcesV(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: void VSourcesV(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
vsources all names()
```

Gets the name of the active VSource.

#### 4.1.39 XYCurves

Created by eniocc at 11/10/2020

This interface implements the XYCurves (IXYCurves) interface of OpenDSS by declaring 4 procedures for accessing the different properties included in this interface: XYCurvesS, XYCurvesI, XYCurvesF, XYCurvesV.

```
class py_dss_interface.models.XYCurves.XYCurves.XYCurvesF(obj_dss)
    Bases: py_dss_interface.models.Base.Base
```

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: double XYCurvesF(int32 t Parameter, double Argument);

This interface returns a floating point number with the result of the query according to the value of the variable Parameter, which can be one of the following.

```
xycurves\_read\_x() \rightarrow float
```

Gets the interpolated value after setting Y.

#### $xycurves\_read\_x\_scale() \rightarrow float$

Gets the factor to scale X values from original curve.

#### xycurves\_read\_x\_shift() → float

Gets the amount to shift X value from original curve.

#### $xycurves\_read\_y() \rightarrow float$

Gets the interpolated value after setting X.

#### $xycurves\_read\_y\_scale() \rightarrow float$

Gets the factor to scale Y values from original curve.

#### $xycurves\_read\_y\_shift() \rightarrow float$

Gets the amount to shift Y value from original curve.

#### xycurves\_write\_x (argument) → float

Sets the X value.

#### $xycurves\_write\_x\_scale(argument) \rightarrow float$

Sets the factor to scale X values from original curve.

#### $xycurves\_write\_x\_shift(argument) \rightarrow float$

Sets the amount to shift X value from original curve.

#### $xycurves\_write\_y(argument) \rightarrow float$

Sets the Y value.

#### $xycurves\_write\_y\_scale(argument) \rightarrow float$

Sets the factor to scale Y values from original curve.

#### $xycurves\_write\_y\_shift(argument) \rightarrow float$

Sets the amount to shift Y value from original curve.

```
{\tt class} \ {\tt py\_dss\_interface.models.XYCurves.XYCurves.XYCurves.} \ (obj\_dss)
```

Bases:  $py_dss_interface.models.Base.Base$ 

This interface can be used to read/write certain properties of the active DSS object.

The structure of the interface is as follows: int32 t XYCurvesI(int32 t Parameter, int32 t Argument);

This interface returns an integer with the result of the query according to the value of the variable Parameter, which can be one of the following.

#### ${\tt xycurves\_count}$ () $\to int$

Gets number of XYCurves in active circuit.

#### $xycurves\_first() \rightarrow int$

Sets first XYCurves object active; returns 0 if none.

#### $xycurves_next() \rightarrow int$

Sets next XYCurves object active; returns 0 if none.

#### $xycurves\_read\_npts() \rightarrow int$

Gets the number of points in X-Y curve.

#### xycurves\_write\_npts(argument) → int

Sets the number of points in X-Y curve.

#### class py\_dss\_interface.models.XYCurves.XYCurves.XYCurvesS(obj\_dss)

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

#### The structure of the interface is as follows: CStr XYCurvesS(int32\_t Parameter, CStr Argument);

This interface returns a string with the result of the query according to the value of the variable Parameter, which can be one of the following.

#### xycurves\_read\_name()

Gets the name of the active XYCurve Object.

#### xycurves\_write\_name(argument)

Sets the name of the active XYCurve Object.

```
class py_dss_interface.models.XYCurves.XYCurves.XYCurvesV(obj_dss)
```

Bases: py\_dss\_interface.models.Base.Base

This interface can be used to read/write certain properties of the active DSS object.

#### The structure of the interface is as follows: void XYCurvesS(int32\_t Parameter, VARIANT \*Argument);

This interface returns a Variant with the result of the query according to the value of the variable Parameter, which can be one of the following.

#### xycurves\_read\_x\_array()

Gets the X values as a variant array of doubles. Set Npts to max number expected if setting.

#### xycurves\_read\_y\_array()

Gets the Y values as a variant array of doubles. Set Npts to max number expected if setting..

#### xycurves\_write\_x\_array(argument)

Sets the X values as a variant array of doubles specified in Argument. Set Npts to max number expected if setting.

#### xycurves\_write\_y\_array (argument)

Sets the Y values as a variant array of doubles specified in Argument. Set Npts to max number expected if setting.

# Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

### 5.1 Bug reports

When reporting a bug please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

## 5.2 Documentation improvements

py\_dss\_interface could always use more documentation, whether as part of the official py\_dss\_interface docs, in docstrings, or even on the web in blog posts, articles, and such.

# 5.3 Feature requests and feedback

The best way to send feedback is to file an issue at https://github.com/PauloRadatz/py\_dss\_interface/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome:)

# CHAPTER 6

# **Authors**

- Paulo Radatz https://www.linkedin.com/in/pauloradatz/ and paulo.radatz@gmail.com
- Ênio Viana https://www.linkedin.com/in/enioviana/ and eniocc@gmail.com
- Rodolfo Pilar Londero https://www.linkedin.com/in/rodolfolondero/ and rodolfopl@gmail.com

94 Chapter 6. Authors

# CHAPTER 7

# Changelog

# 7.1 1.1.0 (2021-10-10)

- · Code refactored
- PEP 8 in some methods

# 7.2 1.0.0 (2021-01-21)

- · Code refactored
- Works on Linux version of OpenDSS provided in the package. This version is from 2020.
- · Tests included
- Methods renamed to satisfy PEP 8 Style Guide for Python Code

## 7.3 0.1.0 (2021-01-21)

• Update OpenDSS version to: OpenDSS Version 9.2.0.1; License Status: Open

# 7.4 0.0.9 (2021-01-21)

- text method returns string values (Jouni request)
- OpenDSS exe 64bits included

## 7.5 0.0.8 (2020-12-10)

- Allowing run multiple instances of DDLL without problems with memory Gustavo asked it
- Allowing run multiple OpenDSS' DLLs
- Update OpenDSS version to: OpenDSS Version 9.1.3.3 (64-bit build); License Status: Open

# 7.6 0.0.7 (2020-10-22)

- PV systems updated.
- Update OpenDSS version to: OpenDSS Version 9.1.0.1 (64-bit build); License Status: Open.
- Allowing to write values into Variant methods.

## 7.7 0.0.4 (2020-08-17)

• DSSDLL class can receive OpenDSS folder in order to use the user own OpenDSS.

## 7.8 0.0.1 (2020-06-12)

• Integrating CI.

## 7.9 0.0.0 (2020-06-12)

• First release on PyPI.

# CHAPTER 8

# Indices and tables

- genindex
- modindex
- search

# Python Module Index

```
py dss interface.models.Meters.Meters,
py_dss_interface.models.ActiveClass.ActiveClass,
                                                                                                                                           py_dss_interface.models.Monitors.Monitors,
py_dss_interface.models.Bus.Bus, 8
	t py_dss_interface.models.Capacitors.CapacPYods,s_interface.models.Parallel.Parallel,
{\tt py\_dss\_interface.models.CapControls.CapCBM+dss\_interface.models.Parser.Parser,}
py_dss_interface.models.Circuit.Circuit, py_dss_interface.models.PDElements.PDElements,
py_dss_interface.models.CktElement.CktElement.cktElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElement.chtElemen
py_dss_interface.models.CMathLib.CMathLiBy_dss_interface.models.Reclosers.Reclosers,
py_dss_interface.models.CtrlQueue.CtrlQueWedss_interface.models.RegControls.RegControls,
py_dss_interface.models.DSSElement.DSSElemenf.py_dss_interface.models.Relays.Relays,
py_dss_interface.models.DSSExecutive.DSSExedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexedutive.pssexed
py_dss_interface.models.DSSInterface.DSSPMtdsface.models.Settings.Settings,
py_dss_interface.models.DSSProgress.DSSPP&gdess_interface.models.Solution.Solution, 28
py_dss_interface.models.DSSProperties.DSBFrdperinterface.models.SwtControls.SwtControls,
py_dss_interface.models.ErrorInterface.EPY-dfnterface.models.Text.Text,82
                                                                                                                                          py_dss_interface.models.Topology.Topology,
py_dss_interface.models.Fuses.Fuses, 30
py_dss_interface.models.Generators.Generators.j.interface.models.Transformers.Transformers,
py_dss_interface.models.ISources.ISourcegy_dss_interface.models.VSources.VSources,
py_dss_interface.models.LineCodes.LineCodes.jnterface.models.XYCurves.XYCurves,
py_dss_interface.models.Lines.Lines, 39
py_dss_interface.models.Loads.Loads, 42
py_dss_interface.models.LoadShapes.LoadShapes,
```

100 Python Module Index

```
Α
                                                                                                                                                                                                                                  8
active_class_all_names()
                                _____(py_dss_interface.models.ActiveClass.ActiveClass.ActiveClassV
                                                                                                                                                                                                 Bus (class in py_dss_interface.models.Bus.Bus), 8
                               method), 8
                                                                                                                                                                                                 bus all pce active bus()
active_class_count()
                                method), 10
                               method), 7
                                                                                                                                                                                                 bus_all_pde_active_bus()
active class first()
                                method), 10
                               method), 7
                                                                                                                                                                                                 bus_axc_012_matrix()
active_class_get_class_name()
                                (py\_dss\_interface.models. Active Class. Ac
                                                                                                                                                                                                                                  method), 10
                               method), 8
                                                                                                                                                                                                  bus_coord_defined()
active_class_get_name()
                                (py\_dss\_interface.models. Active Class. Ac
                                                                                                                                                                                                                                  method), 10
                                method), 8
                                                                                                                                                                                                 bus_cplx_sequence_voltages()
active_class_next()
                                method), 10
                               method), 7
                                                                                                                                                                                                 bus_distance()(py_dss_interface.models.Bus.Bus.BusF
active class num elements()
                                (py_dss_interface.models.ActiveClass.ActiveClass.ActiveClassIhod), 9
                                                                                                                                                                                                 bus_get_unique_node_number()
                               method), 8
                                                                                                                                                                                                                                  (py_dss_interface.models.Bus.Bus.BusI
active_class_parent_class_name()
                                (py\_dss\_interface.models.ActiveClass.ActiveClass.ActiveClassSpod), 10
                                                                                                                                                                                                 bus_interruptions_avg_duration()
                               method), 8
                                                                                                                                                                                                                                  (py_dss_interface.models.Bus.Bus.BusF
active_class_write_name()
                                (py\_dss\_interface.models. Active Class. Active Class. Active Class food), 9
                                                                                                                                                                                                 bus_interruptions_num()
                                method), 8
                                                                                                                                                                                                                                   (py_dss_interface.models.Bus.Bus.BusF
ActiveClass
                                                                                                           (class
                                                                                                                                                                                   in
                                                                                                                                                                                                                                  method), 9
                               py\_dss\_interface.models. Active Class. Active Class),
                                                                                                                                                                                                  bus_interruptions_total_customers()
                                                                                                                                                                                                                                  (py_dss_interface.models.Bus.Bus.BusF
ActiveClassI
                                                                                                             (class
                                                                                                                                                                                   in
                                                                                                                                                                                                                                  method), 9
                               py_dss_interface.models.ActiveClass.ActiveClass),
                                                                                                                                                                                                                                                         (py_dss_interface.models.Bus.Bus.BusV
                                                                                                                                                                                                 bus isc()
                                                                                                                                                                                                                                  method), 10
ActiveClassS
                                                                                                             (class
                                                                                                                                                                                    in
                               py\_dss\_interface.models. Active Class. Active Class. \\ \verb|pus\_kv\_base()| (py\_dss\_interface.models. Bus. Bus. Bus. Bus. Bus. \\ \\ | Pus\_dss\_interface. \\ | Pus\_ds
                                                                                                                                                                                                                                  method), 9
                                                                                                                                                                                   in bus_lambda()(py_dss_interface.models.Bus.Bus.BusF
ActiveClassV
                                                                                                             (class
                                                                                                                                                                                                                                  method), 9
                               py dss interface.models.ActiveClass.ActiveClass),
```

bus_line_list() (py_dss_interface.models.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bu		y_dss_interface.models.Bus.Bus.BusF ethod), 9
<pre>bus_line_total_miles()</pre>	bus_writ	e_x() (py_dss_interface.models.Bus.Bus.BusF
(py_dss_interface.models.Bus.Bus.BusF		ethod), 9
method), 9		e_y () (py_dss_interface.models.Bus.Bus.BusF
bus_load_list() (py_dss_interface.models.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bu		ethod), 9 matrix() (py_dss_interface.models.Bus.Bus.BusV
bus_name() (py_dss_interface.models.Bus.Bus.BusS		ethod), 11
method), 10		() (py_dss_interface.models.Bus.Bus.BusV
bus_nodes() (py_dss_interface.models.Bus.Bus.BusV		ethod), 11
method), 11		() (py_dss_interface.models.Bus.Bus.BusV
bus_num_nodes() (py_dss_interface.models.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bu		ethod), 11
bus_outage_customer_accum_duration()		matrix() (py_dss_interface.models.Bus.Bus.BusV ethod), 11
(py_dss_interface.models.Bus.Bus.BusF		refresh()
method), 9		y_dss_interface.models.Bus.Bus.BusI
bus_pu_vll() (py_dss_interface.models.Bus.Bus.BusV	_	ethod), 10
method), 11		s in py_dss_interface.models.Bus.Bus), 8
<pre>bus_pu_vmag_angle()</pre>		s in py_dss_interface.models.Bus.Bus), 9
(py_dss_interface.models.Bus.Bus.BusV		s in py_dss_interface.models.Bus.Bus), 10
method), 11		s in py_dss_interface.models.Bus.Bus), 10
<pre>bus_pu_voltages()</pre>		
(py_dss_interface.models.Bus.Bus.BusV	С	
method), 11	Capacito	rs (class in
<pre>bus_read_latitude()</pre>	_	y_dss_interface.models.Capacitors.Capacitors),
$(py\_dss\_interface.models.Bus.Bus.BusF$	11	
method), 9	capacito	rs_add_step()
<pre>bus_read_longitude()</pre>	( <i>p</i>	$y\_dss\_interface.models. Capacitors. Capacitors. Capacitors Capacitor Capaci$
(py_dss_interface.models.Bus.Bus.BusF	m	ethod), 12
method), 9		rs_all_names()
<pre>bus_read_x() (py_dss_interface.models.Bus.Bus.BusF</pre>	_	y_dss_interface.models.Capacitors.Capacitors.Capacitors\ ethod), 13
bus_read_y() (py_dss_interface.models.Bus.Bus.BusF	capacito	rs_available_steps()
method), 9	T	$y\_dss\_interface.models. Capacitors. Capacitors. Capacitors Capacitor Capaci$
bus_section_id() (py_dss_interface.models.Bus.Bus.		ethod), 12
method), 10		rs_close()
<pre>bus_seq_voltages()</pre>		y_dss_interface.models.Capacitors.Capacitors.Capacitors
method), 11		ethod), 12
bus_total_customers()		rs_count()
(py_dss_interface.models.Bus.Bus.BusI	_	y_dss_interface.models.Capacitors.Capacitors.Capacitorsl ethod), 12
method), 10		rs_first()
bus_vll() (py_dss_interface.models.Bus.BusV		ry_dss_interface.models.Capacitors.Capacitors.Capacitors
method), 11	-	ethod), 12
$\verb"bus_vmag_angle" () \textit{ (py\_dss\_interface.models.Bus.Bus.} \\$	BusVacito	rs next()
method), 11		y_dss_interface.models.Capacitors.Capacitors.Capacitors1
bus_voc() (py_dss_interface.models.Bus.Bus.BusV	-	ethod), 12
method), 11	capacito	rs_open()
$\verb"bus_voltages"()" (py\_dss\_interface.models.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bus.Bu$		y_dss_interface.models.Capacitors.Capacitors.Capacitorsl
method), 11	_	ethod), 12
<pre>bus_write_latitude()</pre>	capacito	rs_read_is_delta()
(py_dss_interface.models.Bus.Bus.BusF	_	$y\_dss\_interface.models. Capacitors. Capacitors. Capacitors and the contract of the contract $
method), 9		ethod), 12
<pre>bus_write_longitude()</pre>	capacito	rs_read_kv()

102 Index

```
(py_dss_interface.models.Capacitors.Capacitors.Capacitors.Tpy_dss_interface.models.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapCont
                                                                       method), 12
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             method), 15
 capacitors_read_kvar()
                                                                                                                                                                                                                                                                                                                                                                                                                                                    capcontrols_first()
                                                                         (py\_dss\_interface.models. Capacitors. Ca
                                                                       method), 12
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            method), 15
 capacitors_read_name()
                                                                                                                                                                                                                                                                                                                                                                                                                                                    capcontrols_next()
                                                                        (py_dss_interface.models.Capacitors.Capacitors.Capacitors.Sy_dss_interface.models.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapContr
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            method), 15
                                                                       method), 13
 capacitors_read_num_steps()
                                                                                                                                                                                                                                                                                                                                                                                                                                                    capcontrols_read_capacitor()
                                                                         (py_dss_interface.models.Capacitors.Capacitors.Capacitors.py_dss_interface.models.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapContr
                                                                       method), 12
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             method), 15
 capacitors_read_states()
                                                                                                                                                                                                                                                                                                                                                                                                                                                    capcontrols_read_ct_ratio()
                                                                        (py_dss_interface.models.Capacitors.Capacitors.Capacitors.Vyy_dss_interface.models.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapCont
                                                                       method), 13
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            method), 14
capacitors_subtract_step()
                                                                                                                                                                                                                                                                                                                                                                                                                                                   capcontrols_read_dead_time()
                                                                         (py\_dss\_interface.models. Capacitors. Ca
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            method), 14
                                                                       method), 13
capacitors_write_is_delta()
                                                                                                                                                                                                                                                                                                                                                                                                                                                    capcontrols_read_delay()
                                                                         (py\_dss\_interface.models. Capacitors. Ca
                                                                       method), 13
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             method), 14
capacitors_write_kv()
                                                                                                                                                                                                                                                                                                                                                                                                                                                   capcontrols_read_delay_off()
                                                                        (py_dss_interface.models.Capacitors.Capacitors.Capacitors.Tapacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacitors.Capacito
                                                                       method), 12
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             method), 14
 capacitors_write_kvar()
                                                                                                                                                                                                                                                                                                                                                                                                                                                   capcontrols_read_mode()
                                                                         (py_dss_interface.models.Capacitors.Capacitors.Capacitors.Fy_dss_interface.models.CapControls.CapControls.CapControls
                                                                       method), 12
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            method), 15
 capacitors_write_name()
                                                                                                                                                                                                                                                                                                                                                                                                                                                    capcontrols_read_monitored_obj()
                                                                        (py_dss_interface.models.Capacitors.Capacitors.Capacitors.Sy_dss_interface.models.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapContr
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            method), 16
capacitors_write_num_steps()
                                                                                                                                                                                                                                                                                                                                                                                                                                                   capcontrols_read_monitored_term()
                                                                         (py_dss_interface.models.Capacitors.Capacitors.Capacitors.fpy_dss_interface.models.CapControls.CapControls.CapControls
                                                                       method), 13
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             method), 15
                                                                                                                                                                                                                                                                                                                                                                                                                                                   capcontrols_read_name()
capacitors_write_states()
                                                                         (py_dss_interface.models.Capacitors.Capacitors.Capacitors.Yby_dss_interface.models.CapControls.CapControls.CapControls
                                                                       method), 13
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            method), 16
CapacitorsF
                                                                                                                                                                                                                                                (class
                                                                                                                                                                                                                                                                                                                                                                                                                   in capcontrols_read_off_setting()
                                                                       py_dss_interface.models.Capacitors.Capacitors),
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             (py_dss_interface.models.CapControls.CapControls.CapControls
                                                                        11
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            method), 14
CapacitorsI
                                                                                                                                                                                                                                                (class
                                                                                                                                                                                                                                                                                                                                                                                                                     in capcontrols_read_on_setting()
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             (py_dss_interface.models.CapControls.CapControls.CapControls
                                                                       py_dss_interface.models.Capacitors.Capacitors),
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            method), 14
                                                                                                                                                                                                                                                (class
                                                                                                                                                                                                                                                                                                                                                                                                                   in capcontrols_read_pt_ratio()
CapacitorsS
                                                                       py_dss_interface.models.Capacitors.Capacitors),
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             (py\_dss\_interface.models. Cap Controls. Cap Controls. Cap Controls and Cap Control and Cap C
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             method), 14
                                                                         13
                                                                                                                                                                                                                                                                                                                                                                                                                   in capcontrols_read_use_volt_override()
CapacitorsV
                                                                                                                                                                                                                                               (class
                                                                       py_dss_interface.models.Capacitors.Capacitors),
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             (py\_dss\_interface.models. CapControls. Cap
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            method), 15
                                                                         13
 CapControls
                                                                                                                                                                                                                                                (class
                                                                                                                                                                                                                                                                                                                                                                                                                     in capcontrols_read_vmax()
                                                                       py\_dss\_interface.models.CapControls.CapControls),
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             (py\_dss\_interface.models. CapControls. Cap
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             method), 14
                                                                                                                                                                                                                                                                                                                                                                                                                                                    capcontrols_read_vmin()
capcontrols_all_names()
                                                                        (py_dss_interface.models.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapControls.CapCo
                                                                       method), 16
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             method), 14
 capcontrols_count()
                                                                                                                                                                                                                                                                                                                                                                                                                                                    capcontrols_write_capacitor()
```

Index 103

```
(py_dss_interface.models.CapControls.CapControls.CapControls.CapControls),
        method), 16
capcontrols_write_ct_ratio()
                                                  Circuit (class in py_dss_interface.models.Circuit.Circuit),
        (py_dss_interface.models.CapControls.CapControls.CapControlsF
        method), 14
                                                  circuit_all_bus_distances()
capcontrols write dead time()
                                                           (py dss interface.models.Circuit.Circuit.CircuitV
        (py dss interface.models.CapControls.CapControls.CapContethsH), 18
        method), 14
                                                  circuit_all_bus_names()
capcontrols_write_delay()
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitV
        (py_dss_interface.models.CapControls.CapContethsH), 18
        method), 14
                                                  circuit_all_bus_vmag()
capcontrols_write_delay_off()
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitV
        (py_dss_interface.models.CapControls.CapControls.CapContetlsH), 18
        method), 14
                                                  circuit_all_bus_vmag_pu()
capcontrols_write_mode()
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitV
        (py_dss_interface.models.CapControls.CapControls.CapContetlsId), 18
        method), 15
                                                  circuit_all_bus_volts()
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitV
capcontrols_write_monitored_obj()
        (py_dss_interface.models.CapControls.CapControls.CapConteths81), 18
                                                  circuit_all_element_losses()
        method), 16
capcontrols_write_monitored_term()
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitV
        (py_dss_interface.models.CapControls.CapControls.CapContethsld), 18
        method), 15
                                                  circuit_all_element_names()
capcontrols write name()
                                                           (py dss interface.models.Circuit.Circuit.CircuitV
        (py_dss_interface.models.CapControls.CapControls.CapConteths81), 19
        method), 16
                                                  circuit_all_node_distances()
capcontrols_write_off_setting()
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitV
        (py_dss_interface.models.CapControls.CapControls.CapContetlsEI), 19
                                                  circuit_all_node_distances_by_phase()
        method), 14
capcontrols_write_on_setting()
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitV
        (py_dss_interface.models.CapControls.CapControls.CapContetlsH), 19
        method), 14
                                                  circuit_all_node_names()
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitV
capcontrols_write_pt_ratio()
        (py_dss_interface.models.CapControls.CapContethsH), 19
        method), 14
                                                  circuit_all_node_names_by_phase()
                                                           (py\_dss\_interface.models.Circuit.Circuit.CircuitV
capcontrols_write_use_volt_override()
        (py_dss_interface.models.CapControls.CapControls.CapContethsld), 19
        method), 15
                                                  circuit_all_node_vmag_by_phase()
capcontrols_write_vmax()
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitV
        (py_dss_interface.models.CapControls.CapControls.CapContethsH), 19
        method), 15
                                                  circuit_all_node_vmag_pu_by_phase()
capcontrols_write_vmin()
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitV
        (py_dss_interface.models.CapControls.CapControls.CapContethsH), 19
        method), 15
                                                  circuit_capacity()
CapControlsF
                            (class
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitF
                                               in
        py_dss_interface.models.CapControls.CapControls),
                                                           method), 16
        14
                                                  circuit_disable()
                                                           (py_dss_interface.models.Circuit.Circuit.CircuitS
CapControlsI
                            (class
                                              in
        py\_dss\_interface.models.CapControls.CapControls),
                                                           method), 18
                                                  circuit_enable() (py_dss_interface.models.Circuit.Circuit.CircuitS
                            (class
CapControlsS
                                                           method), 18
                                               in
        py_dss_interface.models.CapControls.CapControls(),rcuit_end_of_time_step_update()
                                                           (py_dss_interface.models.Circuit.CircuitI
CapControlsV
                                              in
                            (class
                                                           method), 17
```

104 Index

```
circuit first element()
                                                                                                (py_dss_interface.models.Circuit.Circuit.CircuitS
             (py_dss_interface.models.Circuit.CircuitI
                                                                                                method), 18
             method), 17
                                                                                  circuit set active element()
                                                                                                (py_dss_interface.models.Circuit.Circuit.CircuitS
circuit_first_pc_element()
             (py_dss_interface.models.Circuit.CircuitI
                                                                                                method), 18
             method), 17
                                                                                  circuit substation losses()
circuit_first_pd_element()
                                                                                                (py dss interface.models.Circuit.CircuitV
             (py_dss_interface.models.Circuit.CircuitI
                                                                                                method), 19
             method), 17
                                                                                  circuit_system_y()
circuit_float()(py_dss_interface.models.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circuit.Circui
             method), 16
                                                                                                method), 19
circuit_integer()
                                                                                  circuit_total_power()
             (py_dss_interface.models.Circuit.CircuitI
                                                                                                (py_dss_interface.models.Circuit.CircuitV
             method), 17
                                                                                                method), 19
circuit_line_losses()
                                                                                  circuit_update_storage_t()
             (py_dss_interface.models.Circuit.Circuit.CircuitV
                                                                                                (py_dss_interface.models.Circuit.Circuit.CircuitI
                                                                                                method), 17
             method), 19
circuit_losses() (py_dss_interface.models.Circuit.Circuit.CircuitV_currents()
                                                                                                (py_dss_interface.models.Circuit.Circuit.CircuitV
             method), 19
circuit_name() (py_dss_interface.models.Circuit.Circuit.Circuit&nethod), 19
             method), 18
                                                                                  circuit_y_node_order()
circuit_next_element()
                                                                                                (py_dss_interface.models.Circuit.CircuitV
             (py_dss_interface.models.Circuit.CircuitI
                                                                                                method), 19
             method), 17
                                                                                  circuit_y_node_varray()
                                                                                                (py_dss_interface.models.Circuit.Circuit.CircuitV
circuit_next_pc_element()
             (py_dss_interface.models.Circuit.CircuitI
                                                                                                method), 19
             method), 17
                                                                                  CircuitF (class in py_dss_interface.models.Circuit.Circuit),
circuit_next_pd_element()
             (py_dss_interface.models.Circuit.Circuit.CircuitI CircuitI (class in py_dss_interface.models.Circuit.Circuit),
             method), 17
circuit_num_buses()
                                                                                  CircuitS (class in py_dss_interface.models.Circuit.Circuit),
             (py_dss_interface.models.Circuit.CircuitI
                                                                                                17
             method), 17
                                                                                  Circuit V (class in py_dss_interface.models.Circuit.Circuit),
circuit_num_ckt_elements()
                                                                                                18
             (pv dss interface.models.Circuit.Circuit.CircuitI CktElement
                                                                                                                              (class
                                                                                                py_dss_interface.models.CktElement.CktElement),
             method), 17
circuit num nodes()
                                                                                                20
             (py_dss_interface.models.Circuit.Circuit.CircuitI cktelement_all_property_names()
             method), 17
                                                                                                (py_dss_interface.models.CktElement.CktElement.CktElementV
circuit_parent_pd_element()
                                                                                                method), 21
             (py_dss_interface.models.Circuit.CircuitI cktelement_all_variables_names()
                                                                                                (py_dss_interface.models.CktElement.CktElement.CktElementV
             method), 17
circuit_sample() (py_dss_interface.models.Circuit.Circuit.Circuit.Circuit.
             method), 17
                                                                                  cktelement_all_variables_values()
                                                                                                (py\_dss\_interface.models.CktElement.CktElement.CktElementV
circuit_save_sample()
             (py_dss_interface.models.Circuit.CircuitI
                                                                                                method), 22
             method), 17
                                                                                  cktelement_close()
                                                                                                (py_dss_interface.models.CktElement.CktElementI
circuit_set_active_bus()
             (py_dss_interface.models.Circuit.Circuit.CircuitS
                                                                                                method), 20
             method), 18
                                                                                  cktelement_controller()
                                                                                                (py_dss_interface.models.CktElement.CktElement.CktElementS
circuit_set_active_bus_i()
             (py_dss_interface.models.Circuit.CircuitI
                                                                                                method), 21
             method), 17
                                                                                  cktelement_cplx_seq_currents()
circuit_set_active_class()
                                                                                                (py dss interface.models.CktElement.CktElement.CktElementV
```

```
method), 22
                                                                                                                                                                                                                                                                                                                                               method), 21
cktelement_cplx_seq_voltages()
                                                                                                                                                                                                                                                                                               cktelement_open()
                                               (py dss interface.models.CktElement.CktElement.CktElement!) dss interface.models.CktElement.CktElement.CktElement!
                                              method), 22
                                                                                                                                                                                                                                                                                                                                                method), 21
cktelement currents()
                                                                                                                                                                                                                                                                                               cktelement_phase_losses()
                                               (py dss interface.models.CktElement.CktElement.CktElement.VtElement.CktElement.CktElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElement.VtElem
                                              method), 22
                                                                                                                                                                                                                                                                                                                                               method), 22
cktelement_currents_mag_ang()
                                                                                                                                                                                                                                                                                               cktelement_powers()
                                                (py_dss_interface.models.CktElement.CktElement.CktElementV_dss_interface.models.CktElement.CktElement.CktElementV
                                                                                                                                                                                                                                                                                                                                               method), 22
                                              method), 22
cktelement_energymeter()
                                                                                                                                                                                                                                                                                                cktelement_read_bus_names()
                                                (py_dss_interface.models.CktElement.CktElement.CktElementyS_dss_interface.models.CktElement.CktElement.CktElementV
                                              method), 21
                                                                                                                                                                                                                                                                                                                                               method), 22
                                                                                                                                                                                                                                                                                               cktelement_read_display()
cktelement_guid()
                                               (py_dss_interface.models.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktEleme
                                              method), 21
                                                                                                                                                                                                                                                                                                                                                method), 21
cktelement_has_switch_control()
                                                                                                                                                                                                                                                                                               cktelement_read_emerg_amps()
                                               (py_dss_interface.models.CktElement.CktElement.CktElement.TktElement.CktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktElement.TktEleme
                                              method), 20
                                                                                                                                                                                                                                                                                                                                               method), 20
cktelement has volt control()
                                                                                                                                                                                                                                                                                               cktelement read enabled()
                                               (py_dss_interface.models.CktElement.CktElement.CktElement.TktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktEleme
                                              method), 20
                                                                                                                                                                                                                                                                                                                                               method), 21
cktelement_is_open()
                                                                                                                                                                                                                                                                                                cktelement_read_norm_amps()
                                               (py dss interface.models.CktElement.CktElement.CktElement.TktElement.CktElement.CktElement.CktElement.TktElement.CktElement.CktElement.CktElement.TktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktEleme
                                              method), 20
                                                                                                                                                                                                                                                                                                                                               method), 20
cktelement_losses()
                                                                                                                                                                                                                                                                                               cktelement_residuals()
                                                (py_dss_interface.models.CktElement.CktElement.CktElementV_dss_interface.models.CktElement.CktElement.CktElementV
                                              method), 22
                                                                                                                                                                                                                                                                                                                                                method), 22
cktelement_name()
                                                                                                                                                                                                                                                                                               cktelement_seq_currents()
                                               (py_dss_interface.models.CktElement.CktElement.CktElement.VtElement.CktElement.CktElement.CktElement.VtElement.CktElement.CktElement.CktElement.VtElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.
                                              method), 21
                                                                                                                                                                                                                                                                                                                                                method), 22
cktelement_node_order()
                                                                                                                                                                                                                                                                                                cktelement_seq_powers()
                                               (py_dss_interface.models.CktElement.CktElement.CktElementVdss_interface.models.CktElement.CktElementV
                                              method), 22
                                                                                                                                                                                                                                                                                                                                                method), 22
cktelement num conductors()
                                                                                                                                                                                                                                                                                               cktelement_seq_voltages()
                                               (py_dss_interface.models.CktElement.CktElement.CktElement.VtElement.CktElement.CktElement.CktElement.VtElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement
                                              method), 20
                                                                                                                                                                                                                                                                                                                                               method), 22
cktelement_num_controls()
                                                                                                                                                                                                                                                                                               cktelement_variable_i()
                                                (py_dss_interface.models.CktElement.CktElement.CktElementpt_dss_interface.models.CktElement.CktElement.CktElementF
                                              method), 20
                                                                                                                                                                                                                                                                                                                                               method), 20
cktelement num phases()
                                                                                                                                                                                                                                                                                                cktelement voltages()
                                                (py_dss_interface.models.CktElement.CktElement.CktElementy_dss_interface.models.CktElement.CktElement.CktElementV
                                              method), 20
                                                                                                                                                                                                                                                                                                                                               method), 22
cktelement_num_properties()
                                                                                                                                                                                                                                                                                               cktelement_voltages_mag_ang()
                                               (py_dss_interface.models.CktElement.CktElement.CktElement.VtElement.CktElement.CktElement.CktElement.VtElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement.CktElement
                                              method), 20
                                                                                                                                                                                                                                                                                                                                                method), 22
cktelement_num_terminals()
                                                                                                                                                                                                                                                                                               cktelement_write_bus_names()
                                               (py_dss_interface.models.CktElement.CktElement.CktElementV_dss_interface.models.CktElement.CktElement.CktElementV
                                              method), 21
                                                                                                                                                                                                                                                                                                                                                method), 22
cktelement_ocp_dev_index()
                                                                                                                                                                                                                                                                                               cktelement_write_display()
                                                (py_dss_interface.models.CktElement.CktElement.CktElementy_dss_interface.models.CktElement.CktElement.CktElementS
                                              method), 21
                                                                                                                                                                                                                                                                                                                                               method), 21
cktelement_ocp_dev_type()
                                                                                                                                                                                                                                                                                               cktelement_write_emerg_amps()
```

(py\_dss\_interface.models.CktElement.CktElement.CktElementpt\_dss\_interface.models.CktElement.CktElement.CktElementF

(py\_dss\_interface.models.DSSInterface.DSSInterface.DSSInterface

```
method), 20
                                                                                                                                                           ctrlqueue_clear_queue()
                                                                                                                                                                                     (py_dss_interface.models.CtrlQueue.CtrlQueue.CtrlQueueI
cktelement_write_enabled()
                         (py_dss_interface.models.CktElement.CktElement.CktElementtIthod), 24
                         method), 21
                                                                                                                                                           ctrlqueue_ctrlqueue()
cktelement_write_norm_amps()
                                                                                                                                                                                     (py_dss_interface.models.CtrlQueue.CtrlQueue.CtrlQueueV
                         (py_dss_interface.models.CktElement.CktElement.CktElementHhod), 25
                                                                                                                                                           ctrlqueue_delete()
                         method), 20
                                                                                                                                                                                     (py_dss_interface.models.CtrlQueue.CtrlQueue.CtrlQueueI
cktelement_y_prim()
                          (py_dss_interface.models.CktElement.CktElement.CktElement.Vhod), 24
                         method), 22
                                                                                                                                                           ctrlqueue_device_handle()
CktElementF
                                                                                     (class
                                                                                                                                               in
                                                                                                                                                                                     (py_dss_interface.models.CtrlQueue.CtrlQueue.CtrlQueueI
                         py_dss_interface.models.CktElement.CktElement),
                                                                                                                                                                                     method), 24
                                                                                                                                                           ctrlqueue_do_all_queue()
                                                                                                                                                                                     (py_dss_interface.models.CtrlQueue.CtrlQueue.CtrlQueueI
CktElementI
                                                                                     (class
                                                                                                                                                in
                         py_dss_interface.models.CktElement.CktElement),
                                                                                                                                                                                     method), 24
                         20
                                                                                                                                                           ctrlqueue_num_actions()
                                                                                                                                                                                     (py_dss_interface.models.CtrlQueue.CtrlQueue.CtrlQueueI
                                                                                     (class
CktElementS
                                                                                                                                                in
                         py_dss_interface.models.CktElement.CktElement),
                                                                                                                                                                                     method), 24
                         21
                                                                                                                                                            ctrlqueue_pop_action()
CktElementV
                                                                                     (class
                                                                                                                                                                                     (py_dss_interface.models.CtrlQueue.CtrlQueue.CtrlQueueI
                         py_dss_interface.models.CktElement.CktElement),
                                                                                                                                                                                     method), 24
                                                                                                                                                            ctrlqueue_push() (py_dss_interface.models.CtrlQueue.CtrlQueue.Ct
CMathLib (class in py_dss_interface.models.CMathLib.CMathLib), method), 24
                                                                                                                                                            ctrlqueue_queue_size()
cmathlib_cabs() (py_dss_interface.models.CMathLib.CMathLib(6MdtkLibterface.models.CtrlQueue.CtrlQueue.CtrlQueueI
                         method), 23
                                                                                                                                                                                    method), 24
cmathlib_cdang() (py_dss_interface.models.CMathLib:CMathLib:EMathLib:F(py_dss_interface.models.CtrlQueue.CtrlQueue.Ct
                                                                                                                                                                                     method), 24
                         method), 23
cmathlib_cmplx() (py_dss_interface.models.CMathLibCMathLibCMathLibV
                                                                                                                                                                                                                                              (class
                                                                                                                                                                                     py_dss_interface.models.CtrlQueue.CtrlQueue),
                         method), 23
cmathlib_ctopolardeg()
                                                                                                                                                                                     24
                          (py_dss_interface.models.CMathLib.CMathLib.CMathLib.VeueV
                                                                                                                                                                                                                                              (class
                                                                                                                                                                                                                                                                                                           in
                         method), 23
                                                                                                                                                                                    py_dss_interface.models.CtrlQueue.CtrlQueue),
cmathlib_pdegtocomplex()
                         (py_dss_interface.models.CMathLib.CMathLib.CMathLibV
                         method), 23
CMathLibF
                                                                                  (class
                                                                                                                                               in
                                                                                                                                                           dss_classes() (py_dss_interface.models.DSSInterface.DSSInterface.D
                         py_dss_interface.models.CMathLib.CMathLib),
                                                                                                                                                                                     method), 28
                         23
                                                                                                                                                           dss_clear_all() (py_dss_interface.models.DSSInterface.DSSInterface
CMathLibV
                                                                                  (class
                                                                                                                                               in
                                                                                                                                                                                    method), 27
                         py_dss_interface.models.CMathLib.CMathLib),
                                                                                                                                                          dss_default_editor()
                         23
                                                                                                                                                                                     (py_dss_interface.models.DSSInterface.DSSInterface.DSSInterface
CtrlQueue
                                                                                 (class
                                                                                                                                               in
                                                                                                                                                                                     method), 27
                         py_dss_interface.models.CtrlQueue.CtrlQueue), dss_new_circuit()
                                                                                                                                                                                     (py_dss_interface.models.DSSInterface.DSSInterface.DSSInterface
ctrlqueue_action()
                                                                                                                                                                                     method), 27
                         (py\_dss\_interface.models.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue
                         method), 24
                                                                                                                                                                                     (py_dss_interface.models.DSSInterface.DSSInterface.DSSInterface
ctrlqueue_action_code()
                                                                                                                                                                                     method), 27
                          (py_dss_interface.models.CtrlQueue.CtrlQueue.CtrlQueuefn_classes()
                         method), 24
                                                                                                                                                                                     (py_dss_interface.models.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInterface.DSSInt
ctrlqueue_clear_actions()
                                                                                                                                                                                     method), 27
                         (py_dss_interface.models.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.CtrlQueue.C
```

Index 107

method), 24

26

```
method), 27
                                                                                                                                                                                                                                                                                                                              DSSInterface
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   (class
                                                                                                                                                                                                                                                                                                                                                                                    py\_dss\_interface.models.DSSInterface.DSSInterface),
dss_read_allow_forms()
                                                      (py_dss_interface.models.DSSInterface.DSSInterface.DSSInterfaceI
                                                    method), 27
                                                                                                                                                                                                                                                                                                                              DSSInterfaceI
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      in
dss_read_datapath()
                                                                                                                                                                                                                                                                                                                                                                                    py\_dss\_interface.models.DSSInterface.DSSInterface),
                                                     (py\_dss\_interface. DSSInterface. DSSInterface. DSSInterface. DSSInterfaceS
                                                                                                                                                                                                                                                                                                                               DSSInterfaceS
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        (class
 dss_reset () (py_dss_interface.models.DSSInterface.DSSInterfacep<u>DSSInterface.models.DSSI</u>nterface.DSSInterface),
                                                     method), 27
dss_show_panel() (py_dss_interface.models.DSSInterface.DSSInterface.DSSInterface.
                                                    method), 27
                                                                                                                                                                                                                                                                                                                                                                                    py_dss_interface.models.DSSInterface.DSSInterface),
 dss_start() (py_dss_interface.models.DSSInterface.DSSInterface2BSSInterfaceI
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               (class
                                                    method), 27
                                                                                                                                                                                                                                                                                                                               DSSProgress
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      in
dss_user_classes()
                                                                                                                                                                                                                                                                                                                                                                                    py_dss_interface.models.DSSProgress.DSSProgress),
                                                     (py\_dss\_interface. DSSInterface. DSSInterface. DSSInterface. DSSInterface V
                                                                                                                                                                                                                                                                                                                               dssprogress_caption()
dss_version() (py_dss_interface.models.DSSInterface.DSSInterface.DSSInterfaceSmodels.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSPr
                                                    method), 28
                                                                                                                                                                                                                                                                                                                                                                                    method), 28
                                                                                                                                                                                                                                                                                                                              dssprogress_close()
dss_write_allow_forms()
                                                      (py_dss_interface.models.DSSInterface.DSSInterface.DSSIntpyfadxsl_interface.models.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProg
                                                    method), 27
                                                                                                                                                                                                                                                                                                                                                                                    method), 28
dss_write_datapath()
                                                                                                                                                                                                                                                                                                                              dssprogress_pct_progress()
                                                      (py_dss_interface.models.DSSInterface.DSSInterface.DSSIntpyfacess_interface.models.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProg
                                                                                                                                                                                                                                                                                                                                                                                    method), 28
                                                    method), 28
DSSElement
                                                                                                                                                                            (class
                                                                                                                                                                                                                                                                                                        in dssprogress_show()
                                                                                                                                                                                                                                                                                                                                                                                     (py\_dss\_interface.models.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSSProgress.DSS
                                                    py_dss_interface.models.DSSElement.DSSElement),
                                                                                                                                                                                                                                                                                                                                                                                    method), 28
dsselement_all_property_names()
                                                                                                                                                                                                                                                                                                                              DSSProgressI
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       in
                                                      (py_dss_interface.models.DSSElement.DSSElement.DSSEleppend\s_interface.models.DSSProgress.DSSProgress),
                                                    method), 26
 dsselement_name()
                                                                                                                                                                                                                                                                                                                              DSSProgressS
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   (class
                                                     (py_dss_interface.models.DSSElement.DSSElement.DSSEleppendss_interface.models.DSSProgress.DSSProgress),
                                                    method), 25
                                                                                                                                                                                                                                                                                                                                                                                      28
dsselement_num_properties()
                                                                                                                                                                                                                                                                                                                              DSSProperties
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        (class
                                                      (py_dss_interface.models.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSElement.DSSEleme
                                                    method), 25
DSSElementI
                                                                                                                                                                                (class
                                                                                                                                                                                                                                                                                                       in dssproperties_description()
                                                    py\_dss\_interface.models.DSSElement.DSSElement),
                                                                                                                                                                                                                                                                                                                                                                                     (py\_dss\_interface.models.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSS
                                                     25
                                                                                                                                                                                                                                                                                                                                                                                    method), 29
                                                                                                                                                                                                                                                                                                       in dssproperties_name()
DSSElementS
                                                                                                                                                                               (class
                                                    py\_dss\_interface.models.DSSElement.DSSElement),
                                                                                                                                                                                                                                                                                                                                                                                     (py_dss_interface.models.DSSProperties.DSSProperties.DSSProp
                                                     25
                                                                                                                                                                                                                                                                                                                                                                                    method), 29
DSSElementV
                                                                                                                                                                               (class
                                                                                                                                                                                                                                                                                                      in dssproperties_read_value()
                                                    py_dss_interface.models.DSSElement.DSSElement),
                                                                                                                                                                                                                                                                                                                                                                                     (py_dss_interface.models.DSSProperties.DSSProperties.DSSProp
                                                                                                                                                                                                                                                                                                                                                                                    method), 29
                                                                                                                                                                                                                                                                                                       in dssproperties_write_value()
DSSExecutive
                                                                                                                                                                                    (class
                                                                                                                                                                                                                                                                                                                                                                                    (py\_dss\_interface.models.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSSProperties.DSS
                                                    py_dss_interface.models.DSSExecutive.DSSExecutive),
                                                                                                                                                                                                                                                                                                                                                                                     method), 29
                                                                                                                                                                                       (class
DSSExecutiveI
                                                    py_dss_interface.models.DSSExecutive.DSSExecutive),
                                                     26
                                                                                                                                                                                                                                                                                                                               error_code() (py_dss_interface.models.ErrorInterface.ErrorInterface.E
DSSExecutiveS
                                                                                                                                                                                        (class
                                                                                                                                                                                                                                                                                                       in
                                                                                                                                                                                                                                                                                                                                                                                    method), 29
                                                    py\_dss\_interface.models.DSSExecutive.DSSExecutive) \\ \texttt{or\_desc()} (py\_dss\_interface.models.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.ErrorInterface.Err
```

108 Index

method), 30

```
ErrorCode
                                                                                                                                                                          (class
                                                                                                                                                                                                                                                                                                          in fuses_read_monitored_obj()
                                                    py_dss_interface.models.ErrorInterface.ErrorInterface),
                                                                                                                                                                                                                                                                                                                                                                                       (py_dss_interface.models.Fuses.Fuses.FusesS
                                                      29
                                                                                                                                                                                                                                                                                                                                                                                        method), 31
                                                                                                                                                                         (class
                                                                                                                                                                                                                                                                                                          in fuses_read_monitored_term()
ErrorDesc
                                                    py_dss_interface.models.ErrorInterface.ErrorInterface),
                                                                                                                                                                                                                                                                                                                                                                                       (py_dss_interface.models.Fuses.Fuses.FusesI
                                                                                                                                                                                                                                                                                                                                                                                       method), 31
executive_command()
                                                                                                                                                                                                                                                                                                                                   fuses read name()
                                                      (py_dss_interface.models.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExe
                                                    method), 26
                                                                                                                                                                                                                                                                                                                                                                                        method), 31
executive_command_help()
                                                                                                                                                                                                                                                                                                                                   fuses_read_normal()
                                                      (py_dss_interface.models.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExe
                                                     method), 26
                                                                                                                                                                                                                                                                                                                                                                                        method), 32
executive_num_commands()
                                                                                                                                                                                                                                                                                                                                  fuses_read_rated_current()
                                                      (py_dss_interface.models.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExe
                                                    method), 26
                                                                                                                                                                                                                                                                                                                                                                                        method), 30
executive_num_options()
                                                                                                                                                                                                                                                                                                                                   fuses_read_state()
                                                     (py_dss_interface.models.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.dss_interface.models.Fuses.Fuses.FusesV
                                                    method), 26
                                                                                                                                                                                                                                                                                                                                                                                        method), 32
executive_option()
                                                                                                                                                                                                                                                                                                                                  fuses_read_switched_obj()
                                                      (py_dss_interface.models.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExe
                                                    method), 26
                                                                                                                                                                                                                                                                                                                                                                                       method), 31
executive_option_help()
                                                                                                                                                                                                                                                                                                                                   fuses_read_switched_term()
                                                      (py_dss_interface.models.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExe
                                                    method), 26
                                                                                                                                                                                                                                                                                                                                                                                       method), 31
executive_option_value()
                                                                                                                                                                                                                                                                                                                                   fuses_read_tcc_curve()
                                                      (py_dss_interface.models.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExecutive.DSSExe
                                                    method), 26
                                                                                                                                                                                                                                                                                                                                                                                        method), 31
                                                                                                                                                                                                                                                                                                                                   fuses_reset() (py_dss_interface.models.Fuses.Fuses.FusesI
F
                                                                                                                                                                                                                                                                                                                                                                                       method), 31
                                                                                                                                                                                                                                                                                                                                  fuses_write_delay()
Fuses (class in py dss interface.models.Fuses.Fuses),
                                                                                                                                                                                                                                                                                                                                                                                         (py_dss_interface.models.Fuses.Fuses.FusesF
                                                      30
                                                                                                                                                                                                                                                                                                                                                                                        method), 30
 fuses_all_names()
                                                                                                                                                                                                                                                                                                                                  fuses_write_idx()
                                                      (py dss interface.models.Fuses.Fuses.FusesV
                                                                                                                                                                                                                                                                                                                                                                                         (py_dss_interface.models.Fuses.Fuses.FusesI
                                                    method), 32
                                                                                                                                                                                                                                                                                                                                                                                        method), 31
 fuses_close() (py_dss_interface.models.Fuses.Fuses.FusesI
                                                                                                                                                                                                                                                                                                                                   fuses_write_monitored_obj()
                                                    method), 30
                                                                                                                                                                                                                                                                                                                                                                                         (py_dss_interface.models.Fuses.Fuses.FusesS
 fuses_count() (py_dss_interface.models.Fuses.Fuses.FusesI
                                                                                                                                                                                                                                                                                                                                                                                        method), 31
                                                     method), 30
 \verb|fuses_first(|)| \textit{(py\_dss\_interface.models.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.F
                                                                                                                                                                                                                                                                                                                                                                                        (py\_dss\_interface.models.Fuses.Fuses.FusesI
                                                    method), 30
 fuses_is_blown() (py_dss_interface.models.Fuses.Fuses.Fuses.Imethod), 31
                                                                                                                                                                                                                                                                                                                                    fuses write name()
                                                    method), 30
                                                                                                                                                                                                                                                                                                                                                                                         (py_dss_interface.models.Fuses.FusesS
 fuses_next() (py_dss_interface.models.Fuses.Fuses.FusesI
                                                                                                                                                                                                                                                                                                                                                                                        method), 31
                                                    method), 31
                                                                                                                                                                                                                                                                                                                                   fuses_write_normal()
 fuses num phases()
                                                                                                                                                                                                                                                                                                                                                                                         (py_dss_interface.models.Fuses.Fuses.FusesV
                                                     (py_dss_interface.models.Fuses.Fuses.FusesI
                                                                                                                                                                                                                                                                                                                                                                                       method), 32
                                                    method), 31
 \verb|fuses_open()| \textit{(py\_dss\_interface.models.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fuses.Fus
                                                                                                                                                                                                                                                                                                                                                                                         (py_dss_interface.models.Fuses.Fuses.FusesF
                                                    method), 31
                                                                                                                                                                                                                                                                                                                                                                                       method), 30
 fuses read delay()
                                                                                                                                                                                                                                                                                                                                  fuses_write_state()
                                                     (py dss interface.models.Fuses.Fuses.FusesF
                                                                                                                                                                                                                                                                                                                                                                                        (py_dss_interface.models.Fuses.Fuses.FusesV
                                                    method), 30
 fuses_read_idx() (py_dss_interface.models.Fuses.Fuses.FusesImethod), 32
                                                                                                                                                                                                                                                                                                                                   fuses write switched obj()
                                                    method), 31
```

```
(py_dss_interface.models.Fuses.Fuses.FusesS
                                                                                                                                                                                                                                                                                                                                            generators_read_name()
                                                      method), 31
                                                                                                                                                                                                                                                                                                                                                                                                     (py_dss_interface.models.Generators.Generators.GeneratorsS
 fuses write switched term()
                                                                                                                                                                                                                                                                                                                                                                                                    method), 34
                                                        (py_dss_interface.models.Fuses.Fuses.FusesI
                                                                                                                                                                                                                                                                                                                                             generators_read_pf()
                                                      method), 31
                                                                                                                                                                                                                                                                                                                                                                                                      (py_dss_interface.models.Generators.Generators.GeneratorsF
 fuses_write_tcc_curve()
                                                                                                                                                                                                                                                                                                                                                                                                    method), 33
                                                        (py_dss_interface.models.Fuses.Fuses.FusesS
                                                                                                                                                                                                                                                                                                                                             generators_read_phases()
                                                                                                                                                                                                                                                                                                                                                                                                     (py_dss_interface.models.Generators.Generators.GeneratorsI
                                                      method), 31
FusesF (class in py_dss_interface.models.Fuses.Fuses),
                                                                                                                                                                                                                                                                                                                                                                                                     method), 34
                                                                                                                                                                                                                                                                                                                                             generators_read_vmax_pu()
FusesI (class in py_dss_interface.models.Fuses.Fuses),
                                                                                                                                                                                                                                                                                                                                                                                                     (py\_dss\_interface.models.Generators.Generators.GeneratorsF
                                                                                                                                                                                                                                                                                                                                                                                                    method), 33
FusesS (class in py_dss_interface.models.Fuses.Fuses),
                                                                                                                                                                                                                                                                                                                                            generators_read_vmin_pu()
                                                                                                                                                                                                                                                                                                                                                                                                      (py_dss_interface.models.Generators.Generators.GeneratorsF
FusesV (class in py_dss_interface.models.Fuses.Fuses),
                                                                                                                                                                                                                                                                                                                                                                                                     method), 33
                                                                                                                                                                                                                                                                                                                                             generators_register_names()
                                                                                                                                                                                                                                                                                                                                                                                                     (py_dss_interface.models.Generators.Generators.GeneratorsV
G
                                                                                                                                                                                                                                                                                                                                                                                                     method), 34
                                                                                                                                                                                                                                                                                                                                        generators_register_values()
 Generators
                                                                                                                                                                                   (class
                                                      py_dss_interface.models.Generators.Generators),
                                                                                                                                                                                                                                                                                                                                                                                                     (py_dss_interface.models.Generators.Generators.GeneratorsV
                                                                                                                                                                                                                                                                                                                                                                                                     method), 34
                                                        32
generators_all_names()
                                                                                                                                                                                                                                                                                                                                            generators_write_forced_on()
                                                        (py dss interface.models.Generators.Generators.GeneratorsW_dss_interface.models.Generators.Generators.GeneratorsI
                                                                                                                                                                                                                                                                                                                                                                                                    method), 34
                                                      method), 34
 generators_count()
                                                                                                                                                                                                                                                                                                                                             generators_write_idx()
                                                        (py\_dss\_interface.models. Generators. Generators. Generators. Generators Large My\_dss\_interface. models. Generators Large My\_dss\_interface. My\_dss\_int
                                                                                                                                                                                                                                                                                                                                                                                                    method), 34
                                                      method), 33
                                                                                                                                                                                                                                                                                                                                            generators_write_kv()
generators first()
                                                       (py\_dss\_interface.models. Generators. Ge
                                                                                                                                                                                                                                                                                                                                                                                                     method), 33
                                                      method), 33
generators_next()
                                                                                                                                                                                                                                                                                                                                            generators_write_kva_rated()
                                                        (py\_dss\_interface.models. Generators. Generators. Generators I y\_dss\_interface.models. Generators. Generators Face. Models I generator for the property of t
                                                                                                                                                                                                                                                                                                                                                                                                    method), 33
                                                      method), 33
                                                                                                                                                                                                                                                                                                                                            generators_write_kvar()
 generators_read_forced_on()
                                                       (py_dss_interface.models.Generators.Generators.Generators!Ju_dss_interface.models.Generators.Generators.Generators.F
                                                                                                                                                                                                                                                                                                                                                                                                    method), 33
                                                      method), 33
                                                                                                                                                                                                                                                                                                                                            generators_write_kw()
generators_read_idx()
                                                        (py\_dss\_interface.models. Generators. Generators. Generators I y\_dss\_interface. models. Generators. Generators Face. models Generators Generators Face. models Generators Generators Face. models Generators Generators Generators Face. models Generators 
                                                                                                                                                                                                                                                                                                                                                                                                     method), 33
                                                      method), 33
                                                                                                                                                                                                                                                                                                                                            generators_write_model()
generators read kv()
                                                        (py\_dss\_interface.models. Generators. Ge
                                                                                                                                                                                                                                                                                                                                                                                                     method), 34
                                                      method), 32
 generators_read_kva_rated()
                                                                                                                                                                                                                                                                                                                                            generators_write_name()
                                                       (py\_dss\_interface.models. Generators. Ge
                                                                                                                                                                                                                                                                                                                                                                                                    method), 34
                                                      method), 32
                                                                                                                                                                                                                                                                                                                                             generators_write_pf()
 generators_read_kvar()
                                                       (py\_dss\_interface.models. Generators. Ge
                                                      method), 32
                                                                                                                                                                                                                                                                                                                                                                                                    method), 33
                                                                                                                                                                                                                                                                                                                                            generators_write_phases()
 generators_read_kw()
                                                       (py\_dss\_interface.models. Generators. Ge
                                                                                                                                                                                                                                                                                                                                                                                                     method), 34
                                                      method), 33
generators_read_model()
                                                                                                                                                                                                                                                                                                                                            generators_write_vmax_pu()
                                                        (py\_dss\_interface.models. Generators. Generators. Generators I y\_dss\_interface. models. Generators. Generators Face. Matter Section 1 and 1 an
                                                                                                                                                                                                                                                                                                                                                                                                     method), 33
                                                      method), 33
```

```
35
generators_write_vmin_pu()
                           (py_dss_interface.models.Generators.Generators.Generators.F1
                                                                                                                                                                                                                                                   (class
                                                                                                                                                                                                                                                                                                                  in
                                                                                                                                                                                         py_dss_interface.models.ISources.ISources),
                          method), 33
GeneratorsF
                                                                                        (class
                                                                                                                                                   in
                          py_dss_interface.models.Generators.Generators), ISourcesS
                                                                                                                                                                                                                                                   (class
                                                                                                                                                                                                                                                                                                                  in
                                                                                                                                                                                         py_dss_interface.models.ISources.ISources),
GeneratorsI
                                                                                        (class
                                                                                                                                                   in
                          py_dss_interface.models.Generators.Generators),ISourcesV
                                                                                                                                                                                                                                                   (class
                                                                                                                                                                                                                                                                                                                  in
                          33
                                                                                                                                                                                         py_dss_interface.models.ISources.ISources),
GeneratorsS
                                                                                        (class
                                                                                                                                                    in
                                                                                                                                                                                          36
                          py_dss_interface.models.Generators.Generators),
                           34
GeneratorsV
                                                                                        (class
                                                                                                                                                    in LineCodes
                                                                                                                                                                                                                                                   (class
                                                                                                                                                                                                                                                                                                                  in
                          py_dss_interface.models.Generators.Generators),
                                                                                                                                                                                         py_dss_interface.models.LineCodes.LineCodes),
                           34
                                                                                                                                                                                          36
                                                                                                                                                               linecodes_all_names()
                                                                                                                                                                                          (py_dss_interface.models.LineCodes.LineCodes.LineCodesV
ISources (class in py_dss_interface.models.ISources.ISources),
                                                                                                                                                                                          method), 38
                                                                                                                                                               linecodes_count()
isources_all_names()
                                                                                                                                                                                          (py_dss_interface.models.LineCodes.LineCodes.LineCodesI
                          (py_dss_interface.models.ISources.ISourcesV
                                                                                                                                                                                          method), 37
                                                                                                                                                                linecodes_first()
                          method), 36
isources count() (py dss interface.models.ISources.ISources.ISources.ISources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isources.Isourc
                          method), 35
                                                                                                                                                                                          method), 37
isources_first() (py_dss_interface.models.ISources.ISources_first() (py_dss_interface.models.ISources.ISources_first()
                                                                                                                                                                                          (py_dss_interface.models.LineCodes.LineCodes.LineCodesI
                          method), 35
isources_next() (py_dss_interface.models.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISource
                                                                                                                                                               linecodes next() (py dss interface.models.LineCodes.LineCodes.Li
                          method), 35
                                                                                                                                                                                          method), 37
isources read amps()
                          (py_dss_interface.models.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISources.ISour
                          method), 35
                                                                                                                                                                                          (py_dss_interface.models.LineCodes.LineCodes.LineCodesF
isources_read_angle_deg()
                                                                                                                                                                                          method), 36
                           (py_dss_interface.models.ISources.ISources.ISources.Fecodes_read_c1()
                          method), 35
                                                                                                                                                                                          (py_dss_interface.models.LineCodes.LineCodes.LineCodesF
isources_read_frequency()
                                                                                                                                                                                          method), 36
                           (py_dss_interface.models.ISources.ISources.ISourcasfecodes_read_cmatrix()
                          method), 35
                                                                                                                                                                                          (py_dss_interface.models.LineCodes.LineCodes.LineCodesV
isources_read_name()
                                                                                                                                                                                          method), 38
                          (py_dss_interface.models.ISources.ISources.ISources.Secodes_read_emerg_amps()
                                                                                                                                                                                          (py dss interface.models.LineCodes.LineCodes.LineCodesF
                          method), 36
isources_write_amps()
                                                                                                                                                                                          method), 36
                           (py dss interface.models.ISources.ISources.ISources.Fecodes read name()
                          method), 35
                                                                                                                                                                                          (py_dss_interface.models.LineCodes.LineCodes.LineCodesS
isources_write_angle_deg()
                                                                                                                                                                                          method), 38
                           (py_dss_interface.models.ISources.ISources.ISources.Fecodes_read_norm_amps()
                          method), 35
                                                                                                                                                                                          (py_dss_interface.models.LineCodes.LineCodes.LineCodesF
isources_write_frequency()
                                                                                                                                                                                          method), 36
                           (py_dss_interface.models.ISources.ISources.ISources.Fecodes_read_phases()
                                                                                                                                                                                          (py_dss_interface.models.LineCodes.LineCodes.LineCodesI
                          method), 35
isources_write_name()
                                                                                                                                                                                          method), 37
                           (py_dss_interface.models.ISources.ISources.ISources.Secodes_read_r0()
                          method), 36
                                                                                                                                                                                          (py_dss_interface.models.LineCodes.LineCodes.LineCodesF
ISourcesF
                                                                                    (class
                                                                                                                                                                                          method), 36
                                                                                                                                                               linecodes read r1()
                          py dss interface.models.ISources.ISources),
```

```
(py_dss_interface.models.LineCodes.LineCodes.LineCodes.Fipy_dss_interface.models.LineCodes.LineCodes.LineCodesF
                        method), 37
                                                                                                                                                                                 method), 37
linecodes read rmatrix()
                                                                                                                                                        linecodes write xmatrix()
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodesVpy_dss_interface.models.LineCodes.LineCodes.LineCodesV
                         method), 38
                                                                                                                                                                                 method), 39
linecodes read units()
                                                                                                                                                       LineCodesF
                                                                                                                                                                                                                                          (class
                                                                                                                                                                                                                                                                                                    in
                         (py dss interface.models.LineCodes.LineCodes.LineCodeslpy dss interface.models.LineCodes.LineCodes),
                        method), 38
                                                                                                                                                                                 36
linecodes_read_x0()
                                                                                                                                                        LineCodesI
                                                                                                                                                                                                                                          (class
                                                                                                                                                                                                                                                                                                     in
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodesFpy_dss_interface.models.LineCodes.LineCodes),
                         method), 37
linecodes_read_x1()
                                                                                                                                                       LineCodesS
                                                                                                                                                                                                                                          (class
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodes.Fy_dss_interface.models.LineCodes.LineCodes),
                        method), 37
                                                                                                                                                                                  38
linecodes_read_xmatrix()
                                                                                                                                                                                                                                          (class
                                                                                                                                                       LineCodesV
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodes)\(\psi_y\)_dss_interface.models.LineCodes.LineCodes),
                        method), 38
linecodes write c0()
                                                                                                                                                       Lines (class in py_dss_interface.models.Lines.Lines),
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodesE9
                        method), 37
                                                                                                                                                        lines all names()
linecodes_write_c1()
                                                                                                                                                                                 (py_dss_interface.models.Lines.Lines.LinesV
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodesImethod), 42
                        method), 37
                                                                                                                                                        lines_count() (py_dss_interface.models.Lines.Lines.LinesI
linecodes write cmatrix()
                                                                                                                                                                                 method), 40
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodes.Virst() (py_dss_interface.models.Lines.Lines.LinesI
                        method), 38
                                                                                                                                                                                 method), 40
linecodes_write_emerg_amps()
                                                                                                                                                       lines_next() (py_dss_interface.models.Lines.Lines.LinesI
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodesImethod), 41
                        method), 37
                                                                                                                                                       lines_num_cust()(py_dss_interface.models.Lines.Lines.LinesI
linecodes_write_name()
                                                                                                                                                                                 method), 41
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodes.LineSodesSarent() (py_dss_interface.models.Lines.Lines.LinesI
                        method), 38
                                                                                                                                                                                 method), 41
linecodes_write_norm_amps()
                                                                                                                                                        lines_read_bus1()
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodes.Kpy_dss_interface.models.Lines.Lines.LinesS
                         method), 37
                                                                                                                                                                                 method), 41
linecodes_write_phases()
                                                                                                                                                       lines_read_bus2()
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodesI(py_dss_interface.models.Lines.Lines.LinesS
                        method), 38
                                                                                                                                                                                 method), 41
linecodes_write_r0()
                                                                                                                                                        lines_read_c0() (py_dss_interface.models.Lines.Lines.LinesF
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodesFmethod), 39
                        method), 37
                                                                                                                                                       lines_read_c1() (py_dss_interface.models.Lines.Lines.LinesF
linecodes write r1()
                                                                                                                                                                                 method), 39
                         (py dss interface.models.LineCodes.LineCodes.LineCodes.LineCodes.Fead cmatrix()
                        method), 37
                                                                                                                                                                                 (py\_dss\_interface.models.Lines.Lines.LinesV
linecodes_write_rmatrix()
                                                                                                                                                                                 method), 42
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.L
                                                                                                                                                                                  (py_dss_interface.models.Lines.Lines.LinesF
                        method), 38
                                                                                                                                                                                 method), 39
linecodes_write_units()
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCo
                                                                                                                                                                                  (py_dss_interface.models.Lines.Lines.LinesS
                         method), 38
linecodes_write_x0()
                                                                                                                                                                                  method), 41
                         (py_dss_interface.models.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.LineCodes.L
                        method), 37
                                                                                                                                                                                  (py dss interface.models.Lines.LinesF
linecodes_write_x1()
                                                                                                                                                                                 method), 39
```

<pre>lines_read_linecode()</pre>	lines_	_write_cmatrix() (py_dss_interface.models.Lines.Lines.LinesV method),42
lines_read_name() (py_dss_interface.models.Lines.Lines.LinesS method), 41	lines_	_write_emerg_amps() (py_dss_interface.models.Lines.Lines.LinesF method), 40
lines_read_norm_amps()	lines_	_write_geometry()
(py_dss_interface.models.Lines.Lines.LinesF method), 39		(py_dss_interface.models.Lines.Lines.LinesS method), 41
lines_read_phases() (py_dss_interface.models.Lines.Lines.LinesI method), 41	lines_	_write_length() (py_dss_interface.models.Lines.Lines.LinesF method), 40
<pre>lines_read_r0() (py_dss_interface.models.Lines.Line method), 39</pre>	esl <i>Ei<b>ne</b>sF_</i>	_write_linecode() (py_dss_interface.models.Lines.Lines.LinesS
<pre>lines_read_r1() (py_dss_interface.models.Lines.Line method), 39</pre>		
lines_read_rg() (py_dss_interface.models.Lines.Line method), 39		
lines_read_rho() (py_dss_interface.models.Lines.Li	nėsi bines	<u>F</u> write_norm_amps()
<pre>method), 39 lines_read_rmatrix()</pre>		(py_dss_interface.models.Lines.Lines.LinesF method), 40
(py_dss_interface.models.Lines.Lines.LinesV method), 42	lines_	_write_phases() (py_dss_interface.models.Lines.Lines.LinesI
lines_read_season_rating()		method), 41
(py_dss_interface.models.Lines.Lines.LinesF method), 39	lines_	_write_r0() (py_dss_interface.models.Lines.LinesF method), 40
lines_read_spacing() (py_dss_interface.models.Lines.Lines.LinesS	lines_	_write_r1() (py_dss_interface.models.Lines.Lines.LinesF method), 40
method), 41	lines	_write_rg() (py_dss_interface.models.Lines.LinesF
lines_read_units()		method), 40
(py_dss_interface.models.Lines.Lines.LinesI	lines	_write_rho()
method), 41		(py_dss_interface.models.Lines.Lines.LinesF
<pre>lines_read_x0() (py_dss_interface.models.Lines.Line method), 40</pre>		<pre>method), 40 write_rmatrix()</pre>
lines_read_x1() (py_dss_interface.models.Lines.Line method), 40		
lines_read_xg() (py_dss_interface.models.Lines.Line	esLi <b>ne</b> sF_	
		(py_dss_interface.models.Lines.Lines.LinesS method), 42
$(py\_dss\_interface.models.Lines.Lines.V$	lines_	_write_units()
method), 42		(py_dss_interface.models.Lines.Lines.LinesI
<pre>lines_read_yprim()</pre>		method), 41
(py_dss_interface.models.Lines.LinesV	lines_	_write_x0() (py_dss_interface.models.Lines.Lines.LinesF
method), 42	1	method), 40
<pre>lines_write_bus1()</pre>	lines_	_write_x1() (py_dss_interface.models.Lines.Lines.LinesF method), 40
method), 41	linos	_write_xg() (py_dss_interface.models.Lines.LinesF
lines_write_bus2()	TIMES_	method), 40
(py_dss_interface.models.Lines.LinesS	lines_	_write_xmatrix()  (py_dss_interface.models.Lines.LinesV
<pre>method), 41 lines_write_c0() (py_dss_interface.models.Lines.L</pre>	nos Linos	
method), 40		r memoa),42 _write_yprim()
lines_write_c1() (py_dss_interface.models.Lines.Li		
method), 40	,	method), 42

LinesF (class in py_dss_interface.models.Lines.Lines),		method), 43
LinesI (class in py_dss_interface.models.Lines.Lines),	loads_	read_kvar() (py_dss_interface.models.Loads.Loads.LoadsF
40 LinesS (class in py_dss_interface.models.Lines.Lines),	loads	method), 43 _read_kw() (py_dss_interface.models.Loads.Loads.LoadsF
41	10005_	method), 43
LinesV (class in py_dss_interface.models.Lines.Lines), 42	loads_	<pre>_read_kwh() (py_dss_interface.models.Loads.Loads.Loads     method), 43</pre>
Loads (class in py_dss_interface.models.Loads.Loads),	loads_	read_kwh_days()
42 loads_all_names()		(py_dss_interface.models.Loads.Loads.LoadsF method), 43
(py_dss_interface.models.Loads.Loads.LoadsV	loads_	read_model()
method), 48		$(py\_dss\_interface.models.Loads.Loads.LoadsI$
loads_count() (py_dss_interface.models.Loads.Loads		method), 46
method), 46		_read_name()
<pre>loads_first() (py_dss_interface.models.Loads.Loads     method), 46</pre>	.LoadsI	(py_dss_interface.models.Loads.Loads.LoadsS method), 47
${\tt loads\_next()} \ (py\_dss\_interface.models.Loads.Lo$	<i>loladal</i> ds_	read_num_cust()
method), 46		(py_dss_interface.models.Loads.Loads.LoadsI
<pre>loads_read_allocation_factor()</pre>		method), 46
(py_dss_interface.models.Loads.Loads.LoadsF	loads_	read_pct_mean() (py_dss_interface.models.Loads.Loads.LoadsF
<pre>method), 43 loads_read_c_factor()</pre>		method), 43
(py_dss_interface.models.Loads.Loads.LoadsF	loads	read_pct_series_rl()
method), 43	_	(py_dss_interface.models.Loads.Loads.LoadsF
loads_read_class()		method), 43
$(py\_dss\_interface.models.Loads.Loads.LoadsI$	loads_	read_pct_std_dev()
method), 46		(py_dss_interface.models.Loads.Loads.LoadsF
loads_read_cvr_curve()		method), 43
(py_dss_interface.models.Loads.Loads.LoadsS method), 47		<pre>_read_pf() (py_dss_interface.models.Loads.Loads.LoadsF method), 43</pre>
<pre>loads_read_cvr_vars()</pre>	loads_	read_r_neut()
(py_dss_interface.models.Loads.Loads.LoadsF method), 43		(py_dss_interface.models.Loads.Loads.LoadsF method), 43
loads_read_cvr_watts()	loads_	_read_rel_weight()
(py_dss_interface.models.Loads.Loads.LoadsF method), 43		(py_dss_interface.models.Loads.Loads.LoadsF method), 43
loads_read_daily()	loads	read_spectrum()
(py_dss_interface.models.Loads.Loads.LoadsS	10445_	(py_dss_interface.models.Loads.Loads.LoadsS
method), 47		method), 47
<pre>loads_read_duty()</pre>	loads_	read_status()
(py_dss_interface.models.Loads.Loads.LoadsS method), 47		(py_dss_interface.models.Loads.Loads.LoadsI method), 46
<pre>loads_read_growth()</pre>	loads_	_read_vmax_pu()
$(py\_dss\_interface.models.Loads.Loads.LoadsS$		(py_dss_interface.models.Loads.Loads.LoadsF
method), 47		method), 44
loads_read_idx() (py_dss_interface.models.Loads.	o dudsaldosu <u>o</u>	
<pre>method), 46 loads_read_is_delta()</pre>		(py_dss_interface.models.Loads.Loads.LoadsF method), 44
(py_dss_interface.models.Loads.Loads.LoadsI	loads	read_vmin_norm()
method), 46		(py_dss_interface.models.Loads.Loads.LoadsF
loads_read_kv() (py_dss_interface.models.Loads.L	ads.Loads	
method), 43	loads_	_read_vmin_pu()
<pre>loads_read_kva() (py_dss_interface.models.Loads.L</pre>	oads.Load	ds@y_dss_interface.models.Loads.Loads.LoadsF

<pre>method), 44 loads_read_x_neut()</pre>	<pre>loads_write_kw() (py_dss_interface.models.Loads.Loads.LoadsF     method), 44</pre>
(py_dss_interface.models.Loads.Loads.LoadsF	loads_write_kwh()
method), 44	(py_dss_interface.models.Loads.Loads.LoadsF
loads_read_xfkva()	method), 44
(py_dss_interface.models.Loads.Loads.LoadsF	loads_write_kwh_days()
method), 44	(py_dss_interface.models.Loads.Loads.LoadsF
<pre>loads_read_yearly()</pre>	method), 44
(py_dss_interface.models.Loads.Loads.LoadsS	<pre>loads_write_model()</pre>
method), 47	(py_dss_interface.models.Loads.Loads.LoadsI
loads_read_zipv()	method), 46
(py_dss_interface.models.Loads.Loads.LoadsV	<pre>loads_write_name()</pre>
method), 48	(py_dss_interface.models.Loads.Loads.LoadsS
<pre>loads_write_allocation_factor()</pre>	method), 47
(py_dss_interface.models.Loads.Loads.LoadsF	loads_write_num_cust()
method), 44	(py_dss_interface.models.Loads.Loads.LoadsI
<pre>loads_write_c_factor()</pre>	method), 46
(py_dss_interface.models.Loads.Loads.LoadsF	loads_write_pct_mean()
method), 44	(py_dss_interface.models.Loads.Loads.LoadsF
<pre>loads_write_class()</pre>	method), 44
(py_dss_interface.models.Loads.Loads.LoadsI	loads_write_pct_series_rl()
method), 46	(py_dss_interface.models.Loads.Loads.LoadsF
<pre>loads_write_cvr_curve()</pre>	method), 45
(py_dss_interface.models.Loads.Loads.LoadsS	loads_write_pct_std_dev()
method), 47	(py_dss_interface.models.Loads.Loads.LoadsF
loads_write_cvr_vars()	method), 45
(py_dss_interface.models.Loads.Loads.LoadsF	loads_write_pf() (py_dss_interface.models.Loads.Loads.LoadsF
method), 44	method), 45
<pre>loads_write_cvr_watts()</pre>	<pre>loads_write_r_neut()</pre>
(py_dss_interface.models.Loads.Loads.LoadsF	<pre>(py_dss_interface.models.Loads.Loads.LoadsF</pre>
method), 44	method), 45
<pre>loads_write_daily()</pre>	loads_write_rel_weight()
$(py\_dss\_interface.models.Loads.Loads.LoadsS$	(py_dss_interface.models.Loads.Loads.LoadsF
method), 47	method), 45
<pre>loads_write_duty()</pre>	<pre>loads_write_spectrum()</pre>
(py_dss_interface.models.Loads.Loads.LoadsS	(py_dss_interface.models.Loads.Loads.LoadsS
method), 47	method), 47
loads_write_growth()	loads_write_status()
(py_dss_interface.models.Loads.Loads.LoadsS	(py_dss_interface.models.Loads.Loads.LoadsI
method), 47	method), 46
<pre>loads_write_idx()</pre>	loads_write_vmax_pu()
(py_dss_interface.models.Loads.Loads.LoadsI	$(py\_dss\_interface.models.Loads.Loads.LoadsF$
method), 46	method), 45
<pre>loads_write_is_delta()</pre>	<pre>loads_write_vmin_emerg()</pre>
(py_dss_interface.models.Loads.Loads.LoadsI	(py_dss_interface.models.Loads.Loads.LoadsF
method), 46	method), 45
loads_write_kv() (py_dss_interface.models.Loads.L	oddxdxdxdxfrite_vmin_norm()
method), 44	(py_dss_interface.models.Loads.Loads.LoadsF
loads_write_kva()	method), 45
(py_dss_interface.models.Loads.Loads.LoadsF	loads_write_vmin_pu()
- ·	(py_dss_interface.models.Loads.Loads.LoadsF
method), 44	
loads_write_kvar()	method), 45
(py_dss_interface.models.Loads.Loads.LoadsF	<pre>loads_write_x_neut()</pre>
method) 44	(pv. dss. interface models Loads Loads LoadsF

```
method), 45
                                                                                                               loadshapes_read_s_interval()
loads_write_xfkva()
                                                                                                                                 (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapesF
                  (py_dss_interface.models.Loads.Loads.LoadsF
                                                                                                                                 method), 49
                  method), 45
                                                                                                               loadshapes_read_time_array()
                                                                                                                                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapesV
loads_write_yearly()
                  (py_dss_interface.models.Loads.Loads.LoadsS
                                                                                                                                 method), 50
                  method), 47
                                                                                                               loadshapes_read_use_actual()
loads_write_zipv()
                                                                                                                                 (py_dss_interface.models.LoadShapes.LoadShapesI
                  (py_dss_interface.models.Loads.Loads.LoadsV
                                                                                                                                 method), 49
                  method), 48
                                                                                                               loadshapes_write_hr_interval()
LoadsF (class in py_dss_interface.models.Loads.Loads),
                                                                                                                                 (py\_dss\_interface.models.LoadShapes.LoadShapes.LoadShapesF
                  43
                                                                                                                                 method), 49
LoadShapes
                                                            (class
                                                                                                       in loadshapes_write_min_interval()
                  py_dss_interface.models.LoadShapes.LoadShapes),
                                                                                                                                 (py_dss_interface.models.LoadShapes.LoadShapesFoadShapes)
                                                                                                                                 method), 49
loadshapes_all_names()
                                                                                                               loadshapes_write_name()
                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShap
                  method), 50
                                                                                                                                 method), 50
                                                                                                               loadshapes_write_npts()
loadshapes_count()
                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapes_tdss_interface.models.LoadShapes.LoadShapes.LoadShapes.
                  method), 49
                                                                                                                                 method), 49
loadshapes_first()
                                                                                                               loadshapes_write_p_base()
                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapes_ldss_interface.models.LoadShapes.LoadShapes.LoadShapesF
                  method), 49
                                                                                                                                 method), 49
loadshapes_next()
                                                                                                               loadshapes_write_p_mult()
                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapes_ldss_interface.models.LoadShapes.LoadShapes.LoadShapesV
                  method), 49
                                                                                                                                 method), 50
                                                                                                               loadshapes_write_q_base()
loadshapes_normalize()
                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapesEdss_interface.models.LoadShapes.LoadShapes.LoadShapesF
                  method), 49
                                                                                                                                 method), 49
loadshapes_read_hr_interval()
                                                                                                               loadshapes_write_q_mult()
                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapes_Hss_interface.models.LoadShapes.LoadShapes.LoadShapesV
                  method), 48
                                                                                                                                 method), 50
loadshapes_read_min_interval()
                                                                                                               loadshapes_write_s_interval()
                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapes.Hss_interface.models.LoadShapes.LoadShapes.LoadShapesF
                  method), 48
                                                                                                                                 method), 49
loadshapes_read_name()
                                                                                                               loadshapes_write_time_array()
                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapes_Slss_interface.models.LoadShapes.LoadShapes.LoadShapesV
                  method), 50
                                                                                                                                 method), 50
loadshapes_read_npts()
                                                                                                               loadshapes_write_use_actual()
                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapes_ldss_interface.models.LoadShapes.LoadShapes.LoadShapesI
                  method), 49
                                                                                                                                 method), 49
                                                                                                               LoadShapesF
loadshapes_read_p_base()
                                                                                                                                                                            (class
                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapes.toadShapes.toadShapes),
                  method), 48
loadshapes_read_p_mult()
                                                                                                               LoadShapesI
                                                                                                                                                                            (class
                                                                                                                                                                                                                      in
                  (py\_dss\_interface.models.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes),
                  method), 50
loadshapes_read_q_base()
                                                                                                               LoadShapesS
                                                                                                                                                                            (class
                  (py_dss_interface.models.LoadShapes.LoadShapes.LoadShapes.toadShapes.toadShapes),
                  method), 48
loadshapes_read_q_mult()
                                                                                                               LoadShapesV
                                                                                                                                                                            (class
                  (py\_dss\_interface.models.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadShapes.LoadSh
                                                                                                                                 50
                  method), 50
```

```
Loads I (class in py_dss_interface.models.Loads.Loads),
                                                                                                  method), 51
                                                                                    meters_num_section_customers()
Loads (class in py dss interface.models.Loads.Loads),
                                                                                                   (py_dss_interface.models.Meters.MetersI
                                                                                                   method), 51
LoadsV (class in py_dss_interface.models.Loads.Loads), meters_num_sections()
                                                                                                   (py dss interface.models.Meters.MetersI
                                                                                                   method), 51
M
                                                                                    meters_ocp_device_type()
                                                                                                   (py_dss_interface.models.Meters.Meters.MetersI
Meters (class in py_dss_interface.models.Meters.Meters),
                                                                                                   method), 51
                                                                                    meters_open_all_di_files()
meters all branches in zone()
                                                                                                   (py_dss_interface.models.Meters.Meters.MetersI
              (py_dss_interface.models.Meters.Meters.MetersV
                                                                                                   method), 52
             method), 53
meters_all_end_elements()
                                                                                    meters_read_alloc_factors()
                                                                                                   (py\_dss\_interface.models.Meters.Meters.MetersV
              (py_dss_interface.models.Meters.Meters.MetersV
                                                                                                   method), 53
             method), 53
                                                                                    meters_read_calc_current()
meters_all_names()
                                                                                                   (py_dss_interface.models.Meters.Meters.MetersV
              (py_dss_interface.models.Meters.Meters.MetersV
                                                                                                   method), 53
              method), 53
meters_all_pce_in_zone()
                                                                                    meters read metered element()
              (py_dss_interface.models.Meters.Meters.MetersV
                                                                                                   (py_dss_interface.models.Meters.Meters.MetersS
                                                                                                   method), 52
             method), 53
                                                                                    meters_read_metered_terminal()
meters avg repair time()
                                                                                                   (py dss interface.models.Meters.MetersI
              (py_dss_interface.models.Meters.Meters.MetersF
             method), 50
                                                                                                   method), 52
meters_close_all_di_files()
                                                                                    meters_read_name()
                                                                                                   (py_dss_interface.models.Meters.Meters.MetersS
              (py_dss_interface.models.Meters.MetersI
                                                                                                   method), 53
             method), 51
meters_count() (py_dss_interface.models.Meters.Meters_Meters_I read_peak_current()
                                                                                                   (py_dss_interface.models.Meters.Meters.MetersV
              method), 51
meters_count_branches()
                                                                                                   method), 53
              (py_dss_interface.models.Meters.Meters.MetersI meters_read_sequence_index()
                                                                                                   (py_dss_interface.models.Meters.Meters.MetersI
             method), 51
                                                                                                   method), 52
meters_count_end_elements()
              (py_dss_interface.models.Meters.Meters.MetersI meters_register_names()
                                                                                                   (py_dss_interface.models.Meters.Meters.MetersV
              method), 51
                                                                                                   method), 53
meters_cust_interrupts()
              (py_dss_interface.models.Meters.Meters.MetersF meters_register_values()
                                                                                                   (py_dss_interface.models.Meters.Meters.MetersV
             method), 51
                                                                                                  method), 53
meters di files are open()
              (py\_dss\_interface.models.Meters.Meters.MetersI meters\_reset() (py\_dss\_interface.models.Meters.Meters.MetersI
                                                                                                   method), 52
             method), 51
meters_do_reliability_calc()
                                                                                    meters reset all()
                                                                                                   (py_dss_interface.models.Meters.MetersI
              (py_dss_interface.models.Meters.MetersI
                                                                                                   method), 52
             method), 51
                                                                                    meters_saidi() (py_dss_interface.models.Meters.Meters.MetersF
meters_fault_rate_x_repair_hrs()
              (py_dss_interface.models.Meters.Meters.MetersF
                                                                                                   method), 51
             method), 51
                                                                                    meters_saifi() (py_dss_interface.models.Meters.Meters.MetersF
meters_first() (py_dss_interface.models.Meters.Meters.MetersImethod), 51
                                                                                    meters_saifi_kw()
             method), 51
\verb|meters_next()| (py\_dss\_interface.models.Meters.Meters.MetersI (py\_dss\_interface.models.Meters.MetersFiles)| (py\_dss\_interface.models.Meters.MetersI)| (py\_dss\_interface.models.Meters.MetersFiles)| (py\_dss\_interface.models.Meters.MetersI)| (py\_dss\_interface.models.Meters.MetersI)| (py\_dss\_interface.models.Meters.MetersI)| (py\_dss\_interface.models.Meters.MetersI)| (py\_dss\_interface.models.Meters.MetersI)| (py\_dss\_interface.models.Meters.MetersI)| (py\_dss\_interface.models.Meters.MetersI)| (py\_dss\_interface.models.MetersI)| (py\_dss\_inte
             method), 51
                                                                                                  method), 51
                                                                                    meters_sample() (py_dss_interface.models.Meters.Meters.MetersI
meters_num_section_branches()
                                                                                                  method), 52
              (py_dss_interface.models.Meters.Meters.MetersI
```

```
meters_sample_all()
              (py_dss_interface.models.Meters.Meters.MetersI MetersV (class in py_dss_interface.models.Meters.Meters),
             method), 52
meters_save() (py_dss_interface.models.Meters.Meters.Meters.Mrs (class in py_dss_interface.models.Monitors.Monitors),
             method), 52
meters save all()
                                                                                   monitors all names()
                                                                                                 (py\_dss\_interface.models.Monitors.Monitors.MonitorsV
              (py dss interface.models.Meters.MetersI
             method), 52
                                                                                                 method), 56
meters_sect_seq_idx()
                                                                                   monitors_byte_stream()
              (py_dss_interface.models.Meters.MetersI
                                                                                                 (py_dss_interface.models.Monitors.Monitors.MonitorsV
             method), 52
                                                                                                 method), 56
meters_sect_total_cust()
                                                                                   monitors_channel()
              (py_dss_interface.models.Meters.MetersI
                                                                                                 (py_dss_interface.models.Monitors.Monitors.MonitorsV
             method), 52
                                                                                                 method), 56
meters_seq_list_size()
                                                                                   monitors_count() (py_dss_interface.models.Monitors.Monitors.Moni
              (py_dss_interface.models.Meters.MetersI
                                                                                                 method), 54
             method), 52
                                                                                   monitors_dbl_freq()
meters_set_active_section()
                                                                                                 (py_dss_interface.models.Monitors.Monitors.MonitorsV
              (py_dss_interface.models.Meters.Meters.MetersI
                                                                                                 method), 56
             method), 52
                                                                                   monitors_dbl_hour()
meters_sum_branch_flt_rates()
                                                                                                 (py\_dss\_interface.models.Monitors.Monitors.MonitorsV
              (py_dss_interface.models.Meters.Meters.MetersF
                                                                                                 method), 56
             method), 51
                                                                                   monitors_file_name()
                                                                                                 (py\_dss\_interface.models.Monitors.Monitors.MonitorsS
meters_total_customers()
              (py_dss_interface.models.Meters.Meters.MetersI
                                                                                                 method), 55
             method), 52
                                                                                   monitors_file_version()
meters_totals()(py_dss_interface.models.Meters.Meters.Metersyby_dss_interface.models.Monitors.Monitors.Monitors.
             method), 53
                                                                                                 method), 54
                                                                                   monitors_first() (py_dss_interface.models.Monitors.Monitors.Moni
meters_write_alloc_factors()
             (py_dss_interface.models.Meters.Meters.MetersV
                                                                                                 method), 54
             method), 53
                                                                                   monitors_header()
meters_write_calc_current()
                                                                                                 (py_dss_interface.models.Monitors.Monitors.MonitorsV
              (py_dss_interface.models.Meters.Meters.MetersV
                                                                                                 method), 56
                                                                                   monitors_next() (py_dss_interface.models.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitors.Monitor
             method), 53
meters_write_metered_element()
                                                                                                 method), 54
             (py_dss_interface.models.Meters.Meters.MetersS monitors_num_channels()
             method), 53
                                                                                                  (py dss interface.models.Monitors.Monitors.MonitorsI
meters_write_metered_terminal()
                                                                                                 method), 54
              (py_dss_interface.models.Meters.Meters.MetersI monitors_process()
             method), 52
                                                                                                 (py_dss_interface.models.Monitors.Monitors.MonitorsI
meters write name()
                                                                                                 method), 54
              (py_dss_interface.models.Meters.Meters.MetersS monitors_process_all()
                                                                                                 (py_dss_interface.models.Monitors.Monitors.MonitorsI
             method), 53
meters_write_peak_current()
                                                                                                 method), 54
              (py_dss_interface.models.Meters.Meters.MetersVmonitors_read_element()
             method), 53
                                                                                                  (py_dss_interface.models.Monitors.Monitors.MonitorsS
meters_write_sequence_index()
                                                                                                 method), 55
              (py_dss_interface.models.Meters.Meters.MetersI monitors_read_mode()
             method), 52
                                                                                                  (py_dss_interface.models.Monitors.Monitors.MonitorsI
MetersF (class in py_dss_interface.models.Meters.Meters),
                                                                                                 method), 54
                                                                                   monitors_read_name()
MetersI (class in py_dss_interface.models.Meters.Meters),
                                                                                                 (py_dss_interface.models.Monitors.Monitors.MonitorsS
              51
                                                                                                 method), 55
MetersS (class in py_dss_interface.models.Meters.Meters)monitors_read_terminal()
```

```
(py dss interface.models.Monitors.Monitors.Monitors.dllel actor status()
        method), 54
                                                              (py_dss_interface.models.Parallel.Parallel.ParallelV
monitors record size()
                                                              method), 57
         (py_dss_interface.models.Monitors.Monitors.Monitors.Monitors.dllel_create_actor()
        method), 54
                                                              (py_dss_interface.models.Parallel.Parallel.ParallelI
monitors reset() (py dss interface.models.Monitors.Monitors.Methid);s56
                                                     parallel num actors()
        method), 54
                                                              (py_dss_interface.models.Parallel.Parallel.ParallelI
monitors reset all()
         (py dss interface.models.Monitors.Monitors.MonitorsI
                                                              method), 56
        method), 54
                                                     parallel_num_cores()
monitors_sample()
                                                              (py_dss_interface.models.Parallel.Parallel.ParallelI
         (py_dss_interface.models.Monitors.Monitors.MonitorsI
                                                              method), 56
        method), 54
                                                     parallel_num_cpus()
                                                              (py_dss_interface.models.Parallel.Parallel.ParallelI
monitors_sample_all()
         (py_dss_interface.models.Monitors.Monitors.MonitorsI
                                                              method), 57
        method), 54
                                                     parallel_read_active_actor()
                                                              (py_dss_interface.models.Parallel.Parallel.ParallelI
monitors_sample_count()
         (py_dss_interface.models.Monitors.Monitors.MonitorsI
                                                              method), 57
                                                     parallel_read_active_parallel()
        method), 54
monitors_save() (py_dss_interface.models.Monitors.Monitors.Monitors.Monitors_interface.models.Parallel.Parallel.Parallel.
        method), 55
                                                              method), 57
                                                     parallel_read_actor_cpu()
monitors save all()
         (py_dss_interface.models.Monitors.Monitors.MonitorsI
                                                              (py_dss_interface.models.Parallel.Parallel.ParallelI
        method), 55
                                                              method), 57
monitors_show() (py_dss_interface.models.Monitors.Monitors.Monitors.bl_concatenate_reports1()
                                                              (py_dss_interface.models.Parallel.Parallel.ParallelI
        method), 55
monitors_write_element()
                                                              method), 57
         (py_dss_interface.models.Monitors.Monitors.Monitors.Sllel_wait() (py_dss_interface.models.Parallel.Parallel.ParallelI
        method), 55
                                                              method), 57
monitors_write_mode()
                                                     parallel_write_active_actor()
         (py_dss_interface.models.Monitors.Monitors.MonitorsI
                                                              (py_dss_interface.models.Parallel.Parallel.ParallelI
        method), 55
                                                              method), 57
monitors_write_name()
                                                     parallel_write_active_parallel()
         (py_dss_interface.models.Monitors.Monitors.MonitorsS
                                                              (py_dss_interface.models.Parallel.Parallel.ParallelI
        method), 55
                                                              method), 57
monitors_write_terminal()
                                                     parallel_write_actor_cpu()
         (py dss interface.models.Monitors.Monitors.MonitorsI
                                                              (py dss interface.models.Parallel.Parallel.ParallelI
        method), 55
                                                              method), 57
MonitorsI
                            (class
                                                     parallel_write_concatenate_reportsl()
        py_dss_interface.models.Monitors.Monitors),
                                                              (py_dss_interface.models.Parallel.Parallel.ParallelI
                                                              method), 57
                            (class
                                                    ParallelI
                                                                                 (class
MonitorsS
                                                                                                      in
        py_dss_interface.models.Monitors.Monitors),
                                                              py_dss_interface.models.Parallel.Parallel),
        55
                                                              56
MonitorsV
                            (class
                                                 in ParallelV
                                                                                 (class
                                                                                                       in
        py_dss_interface.models.Monitors.Monitors),
                                                              py_dss_interface.models.Parallel.Parallel),
        55
                                                              57
                                                     Parser (class in py_dss_interface.models.Parser.Parser),
Р
Parallel(class in py_dss_interface.models.Parallel.Parallel;ser_dbl_value()
                                                              (py_dss_interface.models.Parser.Parser.ParserF
         56
parallel_actor_progress()
                                                              method), 58
         (py_dss_interface.models.Parallel.Parallel.ParalleФ∀rser_int_value()
                                                              (py dss interface.models.Parser.Parser.ParserI
        method), 57
```

```
method), 58
                                                                                    parser_write_end_quote()
parser_matrix() (py_dss_interface.models.Parser.Parser.Parser(py_dss_interface.models.Parser.Parser.ParserS
              method), 58
                                                                                                  method), 59
parser_matrix() (py_dss_interface.models.Parser.ParserParserVwrite_white_space()
              method), 59
                                                                                                  (py_dss_interface.models.Parser.Parser.ParserS
                                                                                                  method), 59
parser_next_param()
              (py_dss_interface.models.Parser.Parser.ParserS ParserF (class in py_dss_interface.models.Parser.Parser),
             method), 59
parser_read_auto_increment()
                                                                                    ParserI (class in py_dss_interface.models.Parser.Parser),
              (py_dss_interface.models.Parser.Parser.ParserI
             method), 58
                                                                                    ParserS (class in py_dss_interface.models.Parser.Parser),
parser_read_begin_quote()
              (py_dss_interface.models.Parser.Parser.ParserS ParserV (class in py_dss_interface.models.Parser.Parser),
             method), 59
                                                                                                  59
parser_read_cmd_string()
                                                                                    PDElements
                                                                                                                                 (class
                                                                                                                                                                  in
              (py_dss_interface.models.Parser.Parser.ParserS
                                                                                                  py_dss_interface.models.PDElements.PDElements),
             method), 59
parser_read_delimiters()
                                                                                    pdelements_accumulated_1()
              (py_dss_interface.models.Parser.Parser.ParserS
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements.P
             method), 59
                                                                                                  method), 60
parser_read_end_quote()
                                                                                    pdelements_count()
              (py_dss_interface.models.Parser.Parser.ParserS
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements.PDE
             method), 59
                                                                                                  method), 60
parser_read_white_space()
                                                                                    pdelements_first()
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements1
              (py_dss_interface.models.Parser.Parser.ParserS
             method), 59
                                                                                                  method), 61
parser_reset_delimeters()
                                                                                    pdelements_from_terminal()
              (py_dss_interface.models.Parser.Parser.ParserI
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements.P
                                                                                                  method), 61
             method), 58
parser_str_value()
                                                                                    pdelements_is_shunt()
              (py_dss_interface.models.Parser.Parser.ParserS
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements.PDE
             method), 59
                                                                                                  method), 61
                                                                                    pdelements_lambda()
parser_sym_matrix()
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements.P
              (py_dss_interface.models.Parser.Parser.ParserV
             method), 59
                                                                                                  method), 60
parser_symmatrix()
                                                                                    pdelements_next()
              (py_dss_interface.models.Parser.Parser.Parser
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements)
             method), 58
                                                                                                  method), 61
parser_vector() (py_dss_interface.models.Parser.ParsedBarsarents_num_customers()
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements.PDE
             method), 58
\verb|parser_vector|| (py\_dss\_interface.models.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser.Parser
              method), 60
                                                                                    pdelements_parent_pd_element()
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements1
parser_write_auto_increment()
              (py_dss_interface.models.Parser.ParserI
                                                                                                  method), 61
             method), 58
                                                                                    pdelements_read_fault_rate()
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements.PDE
parser_write_begin_quote()
              (py_dss_interface.models.Parser.Parser.ParserS
                                                                                                  method), 60
             method), 59
                                                                                    pdelements_read_name()
parser_write_cmd_string()
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements.
              (py_dss_interface.models.Parser.Parser.ParserS
                                                                                                  method), 61
             method), 59
                                                                                    pdelements_read_pct_permanent()
                                                                                                  (py_dss_interface.models.PDElements.PDElements.PDElements.P
parser_write_delimiters()
              (py_dss_interface.models.Parser.Parser.ParserS
                                                                                                  method), 60
             method), 59
                                                                                    pdelements_repair_time()
```

```
(py_dss_interface.models.PDElements.PDElementsxPSDElementsxFead kvar()
                   method), 60
                                                                                                                                        (py dss interface.models.PVSystems.PVSystems.PVSystemsF
                                                                                                                                        method), 62
pdelements_section_id()
                   (py_dss_interface.models.PDElements.PDElementsxPB)ElementsLead_name()
                   method), 61
                                                                                                                                        (py_dss_interface.models.PVSystems.PVSystems.PVSystemsS
                                                                                                                                        method), 63
pdelements total customers()
                   (py dss interface.models.PDElements.PDElementsxPDElementskread pf ()
                                                                                                                                        (py_dss_interface.models.PVSystems.PVSystems.PVSystemsF
                   method), 61
pdelements_total_miles()
                                                                                                                                        method), 62
                   (py_dss_interface.models.PDElements.PDElementsxPDElementsxFead_pmpp()
                   method), 60
                                                                                                                                        (py_dss_interface.models.PVSystems.PVSystems.PVSystemsF
                                                                                                                                        method), 62
pdelements_write_fault_rate()
                   (py_dss_interface.models.PDElements.PDElementsxPDElementsxForite_idx()
                   method), 60
                                                                                                                                        (py_dss_interface.models.PVSystems.PVSystemsI
pdelements_write_name()
                                                                                                                                        method), 62
                   (py_dss_interface.models.PDElements.PDElementss.PDElementssSvrite_irradiance()
                                                                                                                                        (py_dss_interface.models.PVSystems.PVSystems.PVSystemsF
                   method), 61
pdelements_write_pct_permanent()
                                                                                                                                        method), 62
                   (py_dss_interface.models.PDElements.PDElementss.PDElementss.Firite_kva_rated()
                                                                                                                                        (py_dss_interface.models.PVSystems.PVSystems.PVSystemsF
                   method), 60
PDElementsF
                                                                (class
                                                                                                            in
                                                                                                                                        method), 62
                   py_dss_interface.models.PDElements.PDElementspysystems_write_kvar()
                   60
                                                                                                                                        (py_dss_interface.models.PVSystems.PVSystems.PVSystemsF
                                                                (class
                                                                                                                                       method), 62
PDElementsI
                   py_dss_interface.models.PDElements.PDElementspysystems_write_name()
                                                                                                                                        (py_dss_interface.models.PVSystems.PVSystems.PVSystemsS
                   60
PDElementsS
                                                                (class
                                                                                                                                        method), 63
                                                                                                            in
                   py_dss_interface.models.PDElements.PDElementspysystems_write_pf()
                                                                                                                                        (py_dss_interface.models.PVSystems.PVSystems.PVSystemsF
                                                             (class
                                                                                                            in
                                                                                                                                        method), 62
PVSystems
                   py_dss_interface.models.PVSystems.PVSystems), pvsystems_write_pmpp()
                                                                                                                                        (py_dss_interface.models.PVSystems.PVSystems.PVSystemsF
pvsystems_all_names()
                                                                                                                                        method), 62
                   (py_dss_interface.models.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSystems.PVSSystems.PVSystems.PVSystems.PVSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSS
                                                                                                                                                                                   (class
                                                                                                                                                                                                                                 in
                                                                                                                                       py_dss_interface.models.PVSystems.PVSystems),
                   method), 63
pvsystems count()
                   (py dss interface.models.PVSystems.PVSystems.PVSystemsInsI
                                                                                                                                                                                   (class
                                                                                                                                                                                                                                 in
                   method), 62
                                                                                                                                       py_dss_interface.models.PVSystems.PVSystems),
pvsystems_first()
                                                                                                                                        62
                   (py_dss_interface.models.PVSystems.PVSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems.PVSSystems
                                                                                                                                                                                   (class
                                                                                                                                       py dss interface.models.PVSystems.PVSystems),
                   method), 62
pvsystems_kw() (py_dss_interface.models.PVSystems.PVSystems@2VSystemsF
                                                                                                                    PVSystemsV
                   method), 62
                                                                                                                                                                                   (class
pvsystems_next() (py_dss_interface.models.PVSystems.PVSystems.PVSystems.e.models.PVSystems.PVSystems),
                   method), 62
pvsystems_read_idx()
                                                                                                                    py_dss_interface.models.ActiveClass.ActiveClass
                   (py_dss_interface.models.PVSystems.PVSystems.PVSystems@module), 7
                   method), 62
                                                                                                                    py_dss_interface.models.Bus.Bus (module),
pvsystems_read_irradiance()
                   (py_dss_interface.models.PVSystems.PVSystems.PVSystems.Ft.ftnterface.models.Capacitors.Capacitors
                   method), 62
                                                                                                                                        (module), 11
pvsystems_read_kva_rated()
                                                                                                                    py_dss_interface.models.CapControls.CapControls
                   (py_dss_interface.models.PVSystems.PVSystems.PVSystems@module), 13
                   method), 62
                                                                                                                    py dss interface.models.Circuit.Circuit
```

```
(module), 72
                  (module), 16
py_dss_interface.models.CktElement.CktElemedts_interface.models.Solution.Solution
                                                                                                                              (module), 74
                  (module), 20
py_dss_interface.models.CMathLib.CMathLiby_dss_interface.models.SwtControls.SwtControls
                  (module), 23
                                                                                                                              (module), 80
py_dss_interface.models.CtrlQueue.CtrlQueyedss_interface.models.Text.Text (mod-
                  (module), 23
                                                                                                                              ule), 82
py_dss_interface.models.DSSElement.DSSElemedts_interface.models.Topology.Topology
                  (module), 25
                                                                                                                              (module), 82
py_dss_interface.models.DSSExecutive.DSSFxedssiveterface.models.Transformers.Transformers
                                                                                                                              (module), 83
                  (module), 26
py_dss_interface.models.DSSInterface.DSSpytdssaceterface.models.VSources.VSources
                                                                                                                              (module), 87
                  (module), 26
py_dss_interface.models.DSSProgress.DSSPpygdess_interface.models.XYCurves.XYCurves
                  (module), 28
                                                                                                                              (module), 88
py_dss_interface.models.DSSProperties.DSSProperties
                  (module), 29
py_dss_interface.models.ErrorInterface.ErrorInterface.ErrorInterface.
                                                                                                                                                                     (class
                                                                                                                                                                                                                in
                  (module), 29
                                                                                                                              py_dss_interface.models.Reclosers.Reclosers),
py_dss_interface.models.Fuses.Fuses
                 (module), 30
                                                                                                            reclosers_all_names()
py_dss_interface.models.Generators.Generators
                                                                                                                             (py_dss_interface.models.Reclosers.Reclosers.ReclosersV
                  (module), 32
                                                                                                                              method), 65
py_dss_interface.models.ISources.ISourceseclosers_close()
                  (module), 35
                                                                                                                              (py_dss_interface.models.Reclosers.Reclosers.ReclosersI
py_dss_interface.models.LineCodes.LineCodes
                                                                                                                              method), 64
                  (module), 36
                                                                                                            reclosers_count()
py_dss_interface.models.Lines.Lines
                                                                                                                              (py dss interface.models.Reclosers.Reclosers.ReclosersI
                  (module), 39
                                                                                                                              method), 64
py_dss_interface.models.Loads.Loads
                                                                                                            reclosers_first()
                  (module), 42
                                                                                                                              (py_dss_interface.models.Reclosers.Reclosers.ReclosersI
py_dss_interface.models.LoadShapes.LoadShapes
                                                                                                                             method), 64
                  (module), 48
                                                                                                            reclosers_next() (py_dss_interface.models.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Recl
py_dss_interface.models.Meters.Meters
                                                                                                                              method), 64
                  (module), 50
                                                                                                            reclosers_open() (py_dss_interface.models.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Reclosers.Recl
py_dss_interface.models.Monitors.Monitors
                                                                                                                              method), 64
                  (module), 54
                                                                                                            reclosers_read_ground_inst()
py_dss_interface.models.Parallel.Parallel
                                                                                                                              (py\_dss\_interface.models.Reclosers.Reclosers.ReclosersF
                  (module), 56
                                                                                                                              method), 63
py_dss_interface.models.Parser.Parser
                                                                                                            reclosers_read_ground_trip()
                  (module), 58
                                                                                                                              (py_dss_interface.models.Reclosers.Reclosers.ReclosersF
py_dss_interface.models.PDElements.PDElements
                                                                                                                             method), 63
                  (module), 60
                                                                                                            reclosers_read_idx()
py_dss_interface.models.PVSystems.PVSystems
                                                                                                                              (py_dss_interface.models.Reclosers.Reclosers.ReclosersI
                  (module), 61
                                                                                                                              method), 64
py_dss_interface.models.Reclosers.Reclosers_read_monitored_obj()
                  (module), 63
                                                                                                                              (py_dss_interface.models.Reclosers.Reclosers.ReclosersS
py_dss_interface.models.RegControls.RegControlsmethod),65
                  (module), 65
                                                                                                            reclosers_read_monitored_term()
py_dss_interface.models.Relays.Relays
                                                                                                                              (py dss interface.models.Reclosers.Reclosers.ReclosersI
                  (module), 68
                                                                                                                              method), 64
py_dss_interface.models.Sensors.Sensors reclosers_read_name()
                  (module), 70
                                                                                                                              (py_dss_interface.models.Reclosers.Reclosers.ReclosersS
py_dss_interface.models.Settings.Settings
                                                                                                                              method), 65
```

reclosers\_write\_switched\_term()

```
(py_dss_interface.models.Reclosers.Reclosers.ReclosersI (py_dss_interface.models.Reclosers.Reclosers.ReclosersI
                                                                                                                                                                                                                                                                                                                                                                                                                          method), 64
                                                         method), 64
 reclosers_read_phase_inst()
                                                                                                                                                                                                                                                                                                                                                               ReclosersF
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      in
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              (class
                                                           (py_dss_interface.models.Reclosers.Reclosers.ReclosersF py_dss_interface.models.Reclosers.Reclosers),
                                                         method), 63
reclosers_read_phase_trip()
                                                                                                                                                                                                                                                                                                                                                               ReclosersI
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      in
                                                           (py_dss_interface.models.Reclosers.Reclosers.ReclosersF py_dss_interface.models.Reclosers.Reclosers),
                                                         method), 63
                                                                                                                                                                                                                                                                                                                                                                                                                            64
 reclosers_read_shots()
                                                                                                                                                                                                                                                                                                                                                               ReclosersS
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      in
                                                           (py_dss_interface.models.Reclosers.Reclosers.ReclosersI py_dss_interface.models.Reclosers.Reclosers),
                                                          method), 64
 reclosers_read_switched_obj()
                                                                                                                                                                                                                                                                                                                                                               ReclosersV
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      in
                                                           (py_dss_interface.models.Reclosers.Reclosers.ReclosersS py_dss_interface.models.Reclosers.Reclosers),
                                                         method), 65
 reclosers_read_switched_term()
                                                                                                                                                                                                                                                                                                                                                                 RegControls
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   (class
                                                          (py_dss_interface.models.Reclosers.Reclosers.ReclosersI py_dss_interface.models.RegControls.RegControls),
                                                         method), 64
reclosers_reclose_intervals()
                                                                                                                                                                                                                                                                                                                                                               regcontrols_all_names()
                                                           (py_dss_interface.models.Reclosers.Reclosers.ReclosersV (py_dss_interface.models.RegControls.RegControls.RegControls.
                                                                                                                                                                                                                                                                                                                                                                                                                          method), 68
                                                         method), 65
 reclosers_write_ground_inst()
                                                                                                                                                                                                                                                                                                                                                               regcontrols_count()
                                                          (py_dss_interface.models.Reclosers.Reclosers.ReclosersF (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.R
                                                                                                                                                                                                                                                                                                                                                                                                                          method), 67
                                                         method), 63
reclosers_write_ground_trip()
                                                                                                                                                                                                                                                                                                                                                                 regcontrols_first()
                                                           (py_dss_interface.models.Reclosers.Reclosers.ReclosersF (py_dss_interface.models.RegControls.RegControls.RegControls.
                                                         method), 63
                                                                                                                                                                                                                                                                                                                                                                                                                           method), 67
                                                                                                                                                                                                                                                                                                                                                               regcontrols_next()
reclosers_write_idx()
                                                          (py_dss_interface.models.Reclosers.Reclosers.ReclosersI (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegCon
                                                         method), 64
                                                                                                                                                                                                                                                                                                                                                                                                                           method), 67
 reclosers_write_monitored_obj()
                                                                                                                                                                                                                                                                                                                                                                 regcontrols_read_ct_primary()
                                                          (py\_dss\_interface.models.Reclosers.Reclosers.Reclosers.ReclosersS \\ (py\_dss\_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegCont
                                                                                                                                                                                                                                                                                                                                                                                                                          method), 66
                                                         method), 65
                                                                                                                                                                                                                                                                                                                                                               regcontrols_read_delay()
 reclosers_write_monitored_term()
                                                           (py_dss_interface.models.Reclosers.Reclosers.ReclosersI (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegCon
                                                                                                                                                                                                                                                                                                                                                                                                                          method), 66
                                                         method), 64
reclosers_write_name()
                                                                                                                                                                                                                                                                                                                                                                 regcontrols_read_forward_band()
                                                           (py_dss_interface.models.Reclosers.Reclosers.ReclosersS (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegCo
                                                         method), 65
                                                                                                                                                                                                                                                                                                                                                                                                                           method), 66
reclosers_write_num_fast()
                                                                                                                                                                                                                                                                                                                                                               regcontrols_read_forward_r()
                                                           (py_dss_interface.models.Reclosers.Reclosers.ReclosersI (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegCon
                                                          method), 64
                                                                                                                                                                                                                                                                                                                                                                                                                          method), 66
                                                                                                                                                                                                                                                                                                                                                               regcontrols_read_forward_vreg()
 reclosers_write_phase_inst()
                                                           (py_dss_interface.models.Reclosers.Reclosers.ReclosersF (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegCon
                                                         method), 64
                                                                                                                                                                                                                                                                                                                                                                                                                          method), 66
 reclosers_write_phase_trip()
                                                                                                                                                                                                                                                                                                                                                                 regcontrols_read_forward_x()
                                                          (py_dss_interface.models.Reclosers.Reclosers.ReclosersF (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegCo
                                                         method), 64
                                                                                                                                                                                                                                                                                                                                                               regcontrols_read_is_inverse_time()
reclosers_write_shots()
                                                           (py_dss_interface.models.Reclosers.Reclosers.ReclosersI (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegCon
                                                         method), 64
                                                                                                                                                                                                                                                                                                                                                                                                                           method), 67
 reclosers_write_switched_obj()
                                                                                                                                                                                                                                                                                                                                                               regcontrols_read_is_reversible()
                                                          (py_dss_interface.models.Reclosers.Reclosers.ReclosersS (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.RegControl.R
                                                         method), 65
                                                                                                                                                                                                                                                                                                                                                                                                                          method), 67
```

reclosers\_read\_num\_fast()

method), 66

method), 66

regcontrols\_write\_forward\_r()

```
regcontrols_read_max_tap_change()
                                                                                                                                                                                                                                                                                                                                                                                                                                  regcontrols_write_forward_vreg()
                                                                       (py\_dss\_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.Reg
                                                                    method), 67
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        method), 66
 regcontrols_read_monitored_bus()
                                                                                                                                                                                                                                                                                                                                                                                                                                  regcontrols_write_forward_x()
                                                                       (py_dss_interface.models.RegControls.RegControls.RegContpols@ss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl
                                                                    method), 68
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        method), 66
regcontrols_read_name()
                                                                                                                                                                                                                                                                                                                                                                                                                                   regcontrols_write_is_inverse_time()
                                                                       (py_dss_interface.models.RegControls.RegControls.RegContpols@ss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl
                                                                    method), 68
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         method), 67
 regcontrols_read_pt_ratio()
                                                                                                                                                                                                                                                                                                                                                                                                                                   regcontrols_write_is_reversible()
                                                                       (py_dss_interface.models.RegControls.RegControls.RegContpols#ss_interface.models.RegControls.RegControls.RegControls
                                                                     method), 66
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         method), 67
 regcontrols_read_reverse_band()
                                                                                                                                                                                                                                                                                                                                                                                                                                  regcontrols_write_max_tap_change()
                                                                     (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegCon
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         method), 68
                                                                    method), 66
                                                                                                                                                                                                                                                                                                                                                                                                                                   regcontrols_write_monitored_bus()
 regcontrols_read_reverse_r()
                                                                     (py\_dss\_interface.models. RegControls. Reg
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         method), 68
                                                                    method), 66
regcontrols_read_reverse_vreg()
                                                                                                                                                                                                                                                                                                                                                                                                                                  regcontrols_write_name()
                                                                       (py_dss_interface.models.RegControls.RegControls.RegContpolsdfss_interface.models.RegControls.RegControls.RegControls.
                                                                    method), 66
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        method), 68
 regcontrols_read_reverse_x()
                                                                                                                                                                                                                                                                                                                                                                                                                                  regcontrols_write_pt_ratio()
                                                                     (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegCon
                                                                    method), 66
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        method), 66
                                                                                                                                                                                                                                                                                                                                                                                                                                   regcontrols_write_reverse_band()
 regcontrols_read_tap_delay()
                                                                       (py_dss_interface.models.RegControls.RegControls.RegContp<u>ols</u>#ss_interface.models.RegControls.RegControls.RegControls.
                                                                    method), 66
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         method), 67
                                                                                                                                                                                                                                                                                                                                                                                                                                  regcontrols_write_reverse_r()
 regcontrols_read_tap_number()
                                                                     (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegCon
                                                                    method), 67
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         method), 67
 regcontrols_read_tap_winding()
                                                                                                                                                                                                                                                                                                                                                                                                                                   regcontrols_write_reverse_vreg()
                                                                     (py\_dss\_interface.models. RegControls. Reg
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        method), 67
                                                                    method), 67
 regcontrols_read_transformer()
                                                                                                                                                                                                                                                                                                                                                                                                                                  regcontrols_write_reverse_x()
                                                                       (py_dss_interface.models.RegControls.RegControls.RegContpols&ss_interface.models.RegControls.RegControls.RegControls
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        method), 67
                                                                    method), 68
                                                                                                                                                                                                                                                                                                                                                                                                                                  regcontrols_write_tap_delay()
 regcontrols_read_voltage_limit()
                                                                       (py\_dss\_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.Reg
                                                                    method), 66
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         method), 67
regcontrols_read_winding()
                                                                                                                                                                                                                                                                                                                                                                                                                                  regcontrols_write_tap_number()
                                                                       (py\_dss\_interface.models. RegControls. Reg
                                                                    method), 67
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         method), 68
 regcontrols_write_ct_primary()
                                                                                                                                                                                                                                                                                                                                                                                                                                  regcontrols_write_tap_winding()
                                                                     (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegCon
                                                                    method), 66
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         method), 68
 regcontrols_write_delay()
                                                                                                                                                                                                                                                                                                                                                                                                                                   regcontrols_write_transformer()
                                                                     (py_dss_interface.models.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControl.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegControls.RegCon
                                                                    method), 66
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         method), 68
 regcontrols_write_forward_band()
                                                                                                                                                                                                                                                                                                                                                                                                                                  regcontrols_write_voltage_limit()
                                                                       (py\_dss\_interface.models. RegControls. Reg
```

124 Index

method), 67

 $(py\_dss\_interface.models.RegControls.Reg$ method), 68

regcontrols\_write\_winding()

```
RegControlsF
                             (class
                                                             (py_dss_interface.models.Relays.Relays.RelaysS
        py_dss_interface.models.RegControls.RegControls),
                                                             method), 70
                                                    relays write switched term()
                                                             (py_dss_interface.models.Relays.Relays.RelaysI
RegControlsI
                             (class
                                                in
        py_dss_interface.models.RegControls.RegControls),
                                                             method), 69
                                                    RelaysI (class in py dss interface.models.Relays.Relays),
RegControlsS
                             (class
                                                in
        py_dss_interface.models.RegControls.RegControls@elaysS (class in py_dss_interface.models.Relays.Relays),
        68
                                                in RelaysV (class in py_dss_interface.models.Relays.Relays),
RegControlsV
                             (class
        py_dss_interface.models.RegControls.RegControls),
Relays (class in py_dss_interface.models.Relays.Relays), S
        68
                                                    Sensors (class in py_dss_interface.models.Sensors.Sensors),
relays_all_names()
         (py_dss_interface.models.Relays.Relays.RelaysV sensors_all_names()
        method), 70
                                                             (py_dss_interface.models.Sensors.SensorsV
relays_count() (py_dss_interface.models.Relays.Relays.RelaysImethod), 72
        method), 69
                                                    sensors_count() (py_dss_interface.models.Sensors.Sensors.SensorsI
relays_first() (py_dss_interface.models.Relays.Relays.RelaysImethod), 71
        method), 69
                                                    sensors_first() (py_dss_interface.models.Sensors.Sensors.SensorsI
relays_next() (py_dss_interface.models.Relays.Relays.RelaysI method), 71
        method), 69
                                                    sensors_next() (py_dss_interface.models.Sensors.Sensors.SensorsI
relays_read_idx()
                                                             method), 71
         (py_dss_interface.models.Relays.Relays.RelaysI sensors read currents()
        method), 69
                                                             (py_dss_interface.models.Sensors.Sensors.SensorsV
relays_read_monitored_obj()
                                                             method), 72
         (py_dss_interface.models.Relays.Relays.RelaysS sensors_read_is_delta()
        method), 69
                                                             (py_dss_interface.models.Sensors.Sensors.SensorsI
relays_read_monitored_term()
                                                             method), 71
         (py_dss_interface.models.Relays.Relays.RelaysI sensors_read_kv_base()
        method), 69
                                                             (py_dss_interface.models.Sensors.Sensors.SensorsF
relays_read_name()
                                                             method), 70
         (py_dss_interface.models.Relays.Relays.RelaysS sensors_read_kvars()
        method), 69
                                                             (py_dss_interface.models.Sensors.Sensors.SensorsV
relays_read_switched_obj()
                                                             method), 72
         (py_dss_interface.models.Relays.Relays.RelaysS sensors_read_kws()
        method), 69
                                                             (py_dss_interface.models.Sensors.Sensors.SensorsV
relays_read_switched_term()
                                                             method), 72
        (py_dss_interface.models.Relays.Relays.RelaysI sensors_read_metered_element()
        method), 69
                                                             (py_dss_interface.models.Sensors.Sensors.SensorsS
relays write idx()
                                                             method), 71
         (py_dss_interface.models.Relays.Relays.RelaysI sensors_read_metered_terminal()
        method), 69
                                                             (py_dss_interface.models.Sensors.Sensors.SensorsI
relays_write_monitored_obj()
                                                             method), 71
         (py_dss_interface.models.Relays.Relays.RelaysS sensors_read_name()
        method), 69
                                                             (py_dss_interface.models.Sensors.Sensors.SensorsS
relays_write_monitored_term()
                                                             method), 71
         (py_dss_interface.models.Relays.Relays.RelaysI sensors_read_pct_error()
        method), 69
                                                             (py_dss_interface.models.Sensors.Sensors.SensorsF
relays_write_name()
                                                             method), 70
        (py_dss_interface.models.Relays.Relays.RelaysS sensors_read_reverse_delta()
        method), 70
                                                             (py_dss_interface.models.Sensors.Sensors.SensorsI
relays_write_switched_obj()
                                                             method), 71
```

```
sensors_read_weight()
                                                                                                                   settings_read_allow_duplicates()
                   (py_dss_interface.models.Sensors.Sensors.SensorsF
                                                                                                                                      (py_dss_interface.models.Settings.SettingsI
                  method), 70
                                                                                                                                      method), 73
sensors_reset()(py_dss_interface.models.Sensors.Sensors.ts_ausors_tead_auto_bus_list()
                  method), 71
                                                                                                                                      (py_dss_interface.models.Settings.Settings.SettingsS
                                                                                                                                      method), 74
sensors reset all()
                   (py dss interface.models.Sensors.Sensors.Sensorsdettings read ckt model()
                  method), 71
                                                                                                                                      (py_dss_interface.models.Settings.Settings.SettingsI
sensors_write_currents()
                                                                                                                                      method), 73
                   (py_dss_interface.models.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Senso
                  method), 72
                                                                                                                                      (py\_dss\_interface.models.Settings.Settings.SettingsF
sensors_write_is_delta()
                                                                                                                                      method), 72
                   (py_dss_interface.models.Sensors.Sensors.Sensorsdettings_read_emerg_vmin_pu()
                  method), 71
                                                                                                                                      (py_dss_interface.models.Settings.Settings.SettingsF
sensors_write_kv_base()
                                                                                                                                      method), 72
                   (py_dss_interface.models.Sensors.Sensors.SensorsFettings_read_loss_regs()
                                                                                                                                      (py_dss_interface.models.Settings.SettingsV
                  method), 70
sensors_write_kvars()
                                                                                                                                      method), 74
                   (py_dss_interface.models.Sensors.Sensors.Sensors9\text{\text{\text{tings_read_loss_weight()}}}
                  method), 72
                                                                                                                                      (py_dss_interface.models.Settings.Settings.SettingsF
sensors_write_kws()
                                                                                                                                      method), 72
                   (py_dss_interface.models.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.Sensors.
                  method), 72
                                                                                                                                      (py_dss_interface.models.Settings.Settings.SettingsF
sensors write metered element()
                                                                                                                                      method), 73
                   (py_dss_interface.models.Sensors.Sensors.Sensorssettings_read_norm_vmin_pu()
                  method), 71
                                                                                                                                      (py_dss_interface.models.Settings.SettingsF
sensors_write_metered_terminal()
                                                                                                                                      method), 73
                   (py_dss_interface.models.Sensors.Sensors.Sensorsdettings_read_price_curve()
                  method), 71
                                                                                                                                      (py_dss_interface.models.Settings.SettingsS
sensors_write_name()
                                                                                                                                      method), 74
                   (py_dss_interface.models.Sensors.Sensors.Sensorssettings_read_price_signal()
                  method), 71
                                                                                                                                      (py_dss_interface.models.Settings.Settings.SettingsF
sensors_write_pct_error()
                                                                                                                                      method), 73
                   (py_dss_interface.models.Sensors.Sensors.SensorsFettings_read_trapezoidal()
                  method), 70
                                                                                                                                      (py_dss_interface.models.Settings.Settings.SettingsI
sensors_write_reverse_delta()
                                                                                                                                      method), 73
                   (py_dss_interface.models.Sensors.Sensors.Sensors.dettings_read_ue_regs()
                  method), 71
                                                                                                                                      (py_dss_interface.models.Settings.SettingsV
sensors_write_weight()
                                                                                                                                      method), 74
                   (py_dss_interface.models.Sensors.Sensors.Fettings_read_ue_weight()
                                                                                                                                      (py dss interface.models.Settings.SettingsF
                  method), 70
SensorsF (class in py_dss_interface.models.Sensors.Sensors),
                                                                                                                                      method), 73
                                                                                                                   settings_read_voltage_bases()
                                                                                                                                      (py_dss_interface.models.Settings.Settings.SettingsV
SensorsI (class in py_dss_interface.models.Sensors.Sensors),
                                                                                                                                      method), 74
SensorsS(class in py_dss_interface.models.Sensors.Sensorsettings_read_zone_lock()
                                                                                                                                      (py_dss_interface.models.Settings.Settings.SettingsI
SensorsV (class in py_dss_interface.models.Sensors.Sensors),
                                                                                                                                      method), 73
                                                                                                                   settings_write_allow_duplicates()
Settings (class in py_dss_interface.models.Settings.Settings),
                                                                                                                                      (py_dss_interface.models.Settings.Settings.SettingsI
                                                                                                                                      method), 73
                   72
settings allocation factors()
                                                                                                                   settings write auto bus list()
                   (py_dss_interface.models.Settings.SettingsF
                                                                                                                                      (py_dss_interface.models.Settings.SettingsS
                  method), 72
                                                                                                                                      method), 74
```

```
settings_write_ckt_model()
                                                    Solution (class in py_dss_interface.models.Solution.Solution),
        (py dss interface.models.Settings.SettingsI
                                                             74
        method), 73
                                                    solution build y matrix()
                                                             (py_dss_interface.models.Solution.Solution.SolutionI
settings_write_emerg_vmax_pu()
        (py_dss_interface.models.Settings.Settings.SettingsF
                                                             method), 76
        method), 73
                                                    solution bus levels()
settings write emerg vmin pu()
                                                             (py dss interface.models.Solution.SolutionV
        (py_dss_interface.models.Settings.Settings.SettingsF
                                                             method), 79
        method), 73
                                                    solution_calc_inc_matrix()
                                                             (py\_dss\_interface.models.Solution.Solution.SolutionI
settings_write_loss_regs()
        (py_dss_interface.models.Settings.Settings.SettingsV
                                                             method), 76
        method), 74
                                                    solution_calc_inc_matrix_0()
                                                             (py_dss_interface.models.Solution.Solution.SolutionI
settings_write_loss_weight()
        (py_dss_interface.models.Settings.Settings.SettingsF
                                                             method), 76
                                                    solution_check_controls()
        method), 73
settings_write_norm_vmax_pu()
                                                             (py_dss_interface.models.Solution.Solution.SolutionI
        (py_dss_interface.models.Settings.Settings.SettingsF
                                                             method), 77
        method), 73
                                                    solution check fault status()
settings_write_norm_vmin_pu()
                                                             (py_dss_interface.models.Solution.Solution.SolutionI
        (py dss interface.models.Settings.Settings.SettingsF
                                                             method), 77
        method), 73
                                                    solution_clean_up()
settings_write_price_curve()
                                                             (py_dss_interface.models.Solution.Solution.SolutionI
        (py_dss_interface.models.Settings.Settings.SettingsS
                                                             method), 77
        method), 74
                                                    solution do control actions()
                                                             (py_dss_interface.models.Solution.Solution.SolutionI
settings_write_price_signal()
        (py_dss_interface.models.Settings.Settings.SettingsF
                                                             method), 77
        method), 73
                                                    solution_event_log()
                                                             (py_dss_interface.models.Solution.Solution.SolutionV
settings_write_trapezoidal()
        (py_dss_interface.models.Settings.Settings.SettingsI
                                                             method), 80
        method), 73
                                                    solution_finish_time_step()
settings_write_ue_regs()
                                                             (py_dss_interface.models.Solution.Solution.SolutionI
        (py_dss_interface.models.Settings.Settings.SettingsV
                                                             method), 77
        method), 74
                                                    solution_inc_matrix_cols()
                                                             (py_dss_interface.models.Solution.Solution.SolutionV
settings_write_ue_weight()
        (py dss interface.models.Settings.Settings.SettingsF
                                                             method), 80
        method), 73
                                                    solution_inc_matrix_rows()
settings_write_voltage_bases()
                                                             (py dss interface.models.Solution.Solution.SolutionV
        (py_dss_interface.models.Settings.Settings.SettingsV
                                                             method), 80
        method), 74
                                                    solution_init_snap()
settings_write_zone_lock()
                                                             (py_dss_interface.models.Solution.Solution.SolutionI
        (py dss interface.models.Settings.SettingsI
                                                             method), 77
        method), 74
                                                    solution iterations()
                                                             (py\_dss\_interface.models.Solution.Solution.SolutionI
SettingsF
                           (class
                                                in
        py_dss_interface.models.Settings.Settings),
                                                             method), 77
                                                    solution_laplacian()
                           (class
                                                             (py_dss_interface.models.Solution.Solution.SolutionV
SettingsI
                                                in
        py_dss_interface.models.Settings.Settings),
                                                             method), 80
        73
                                                    solution_mode_id()
                           (class
                                                in
                                                             (py_dss_interface.models.Solution.Solution.SolutionS
SettingsS
        py_dss_interface.models.Settings.Settings),
                                                             method), 79
        74
                                                    solution_most_iterations_done()
                                                             (py_dss_interface.models.Solution.Solution.SolutionI
SettingsV
                           (class
                                                in
                                                             method), 77
        py_dss_interface.models.Settings.Settings),
        74
                                                    solution_nc_matrix()
```

```
(py_dss_interface.models.Solution.Solution.SolutionV
                                                           (py_dss_interface.models.Solution.SolutionS
        method), 80
                                                           method), 79
solution_process_time()
                                                   solution read load model()
        (py_dss_interface.models.Solution.Solution.SolutionF
                                                           (py_dss_interface.models.Solution.Solution.SolutionI
        method), 75
                                                           method), 77
solution process time step()
                                                   solution read load mult()
        (py dss interface.models.Solution.SolutionF
                                                           (py_dss_interface.models.Solution.SolutionF
        method), 75
                                                           method), 75
solution_read_add_type()
                                                   solution_read_max_control_iterations()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                           (py_dss_interface.models.Solution.Solution.SolutionI
        method), 77
                                                           method), 77
solution_read_algorithm()
                                                   solution_read_max_iterations()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                           (py_dss_interface.models.Solution.SolutionI
        method), 77
                                                           method), 77
solution_read_cap_kvar()
                                                   solution_read_mode()
        (py_dss_interface.models.Solution.Solution.FolutionF
                                                           (py_dss_interface.models.Solution.Solution.SolutionI
                                                           method), 77
        method), 75
solution_read_control_actions_done()
                                                   solution read number()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                           (py_dss_interface.models.Solution.Solution.SolutionI
        method), 77
                                                           method), 77
solution_read_control_iterations()
                                                   solution_read_pct_growth()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                           (py\_dss\_interface.models.Solution.Solution.FolutionF
        method), 77
                                                           method), 75
                                                   solution_read_random()
solution read control mode()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                           (py_dss_interface.models.Solution.Solution.SolutionI
        method), 77
                                                           method), 78
solution_read_converged()
                                                   solution_read_seconds()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                           (py_dss_interface.models.Solution.Solution.SolutionF
        method), 77
                                                           method), 75
solution_read_dbl_hour()
                                                   solution_read_step_size()
        (py_dss_interface.models.Solution.SolutionF
                                                           (py_dss_interface.models.Solution.Solution.SolutionF
        method), 75
                                                           method), 75
solution_read_default_daily()
                                                   solution_read_tolerance()
        (py_dss_interface.models.Solution.Solution.SolutionS
                                                           (py_dss_interface.models.Solution.Solution.SolutionF
        method), 79
                                                           method), 75
solution_read_default_yearly()
                                                   solution_read_total_time()
        (py_dss_interface.models.Solution.Solution.SolutionS
                                                           (py dss interface.models.Solution.Solution.SolutionF
        method), 79
                                                           method), 75
solution_read_frequency()
                                                   solution_read_year()
        (py_dss_interface.models.Solution.SolutionF
                                                           (py_dss_interface.models.Solution.Solution.SolutionI
        method), 75
                                                           method), 78
solution_read_gen_kw()
                                                   solution_sample_control_devices()
        (py_dss_interface.models.Solution.Solution.SolutionF
                                                           (py_dss_interface.models.Solution.Solution.SolutionI
        method), 75
                                                           method), 78
                                                   solution_sample_do_control_actions()
solution_read_gen_mult()
        (py_dss_interface.models.Solution.SolutionF
                                                           (py_dss_interface.models.Solution.Solution.SolutionI
        method), 75
                                                           method), 78
                                                   solution_solve() (py_dss_interface.models.Solution.Solution.Solution
solution_read_gen_pf()
        (py_dss_interface.models.Solution.SolutionF
                                                           method), 78
        method), 75
                                                   solution_solve_all()
                                                           (py_dss_interface.models.Solution.Solution.SolutionI
solution_read_hour()
        (py_dss_interface.models.Solution.SolutionI
                                                           method), 78
                                                   solution_solve_direct()
        method), 77
solution_read_ld_curve()
                                                           (py dss interface.models.Solution.SolutionI
```

```
method), 78
                                                          method), 76
solution_solve_no_control()
                                                  solution_write_gen_kw()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                          (py dss interface.models.Solution.SolutionF
        method), 78
                                                          method), 76
solution_solve_plus_control()
                                                  solution_write_gen_mult()
        (py dss interface.models.Solution.Solution.SolutionI
                                                          (py dss interface.models.Solution.SolutionF
        method), 78
                                                          method), 76
                                                  solution_write_gen_pf()
solution_solve_power_flow()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                          (py_dss_interface.models.Solution.Solution.SolutionF
                                                          method), 76
        method), 78
solution_step_size_hr()
                                                  solution_write_hour()
        (py_dss_interface.models.Solution.SolutionF
                                                          (py_dss_interface.models.Solution.Solution.SolutionI
        method), 75
                                                          method), 78
solution_step_size_min()
                                                  solution_write_ld_curve()
        (py\_dss\_interface.models.Solution.Solution.SolutionF
                                                          (py\_dss\_interface.models.Solution.Solution.SolutionS
        method), 76
                                                          method), 79
solution_system_y_changed()
                                                  solution_write_load_model()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                          (py_dss_interface.models.Solution.SolutionI
        method), 78
                                                          method), 79
solution total iterations()
                                                  solution_write_load_mult()
        (py_dss_interface.models.Solution.SolutionI
                                                          (py\_dss\_interface.models.Solution.Solution.FolutionF
        method), 78
solution_write_add_type()
                                                  solution_write_max_control_iterations()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                          (py dss interface.models.Solution.SolutionI
        method), 78
                                                          method), 79
solution_write_algorithm()
                                                  solution_write_max_iterations()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                          (py_dss_interface.models.Solution.Solution.SolutionI
        method), 78
                                                          method), 79
solution_write_cap_kvar()
                                                  solution_write_mode()
        (py_dss_interface.models.Solution.SolutionF
                                                          (py_dss_interface.models.Solution.Solution.SolutionI
        method), 76
                                                          method), 79
solution_write_control_actions_done()
                                                  solution_write_number()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                          (py_dss_interface.models.Solution.SolutionI
        method), 78
                                                          method), 79
solution write control iterations()
                                                  solution_write_pct_growth()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                          (py\_dss\_interface.models.Solution.Solution.FolutionF
        method), 78
                                                          method), 76
solution_write_control_mode()
                                                  solution_write_random()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                          (py_dss_interface.models.Solution.Solution.SolutionI
        method), 78
                                                          method), 79
solution write converged()
                                                  solution write seconds()
        (py_dss_interface.models.Solution.Solution.SolutionI
                                                          (py_dss_interface.models.Solution.SolutionF
        method), 78
                                                          method), 76
solution_write_dbl_hour()
                                                  solution_write_step_size()
                                                          (py\_dss\_interface.models. Solution. Solution. SolutionF
        (py_dss_interface.models.Solution.SolutionF
        method), 76
                                                          method), 76
solution_write_default_daily()
                                                  solution_write_tolerance()
        (py_dss_interface.models.Solution.SolutionS
                                                          (py_dss_interface.models.Solution.SolutionF
        method), 79
                                                          method), 76
solution_write_default_yearly()
                                                  solution_write_total_time()
        (py_dss_interface.models.Solution.Solution.SolutionS
                                                          (py_dss_interface.models.Solution.Solution.SolutionF
        method), 79
                                                          method), 76
solution_write_frequency()
                                                  solution_write_year()
        (py dss interface.models.Solution.SolutionF
                                                          (py dss interface.models.Solution.SolutionI
```

```
method), 79
                                                                                                                                                                                                                                                                                                                                                                                      swtcontrols_write_switched_obj()
                                                                                                                                                                                                     (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                    (py_dss_interface.models.SwtControls.SwtControls.SwtControlsS
SolutionF
                                                                                                                                                                                                                                                                                                                                                       in
                                                            py_dss_interface.models.Solution.Solution), 75
                                                                                                                                                                                                                                                                                                                                                                                                                                                 method), 81
                                                                                                                                                                                                    (class
                                                                                                                                                                                                                                                                                                                                                                                    swtcontrols_write_switched_term()
SolutionI
                                                                                                                                                                                                                                                                                                                                                       in
                                                            py_dss_interface.models.Solution.Solution), 76
                                                                                                                                                                                                                                                                                                                                                                                                                                                    (py_dss_interface.models.SwtControls.SwtControls.SwtControlsI
                                                                                                                                                                                                    (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                  method), 81
SolutionS
                                                                                                                                                                                                                                                                                                                                                       in
                                                            py_dss_interface.models.Solution.Solution), 79
                                                                                                                                                                                                                                                                                                                                                                                    SwtControlsF
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            in
                                                                                                                                                                                                                                                                                                                                                                                                                                                 py\_dss\_interface.models.SwtControls.SwtControls),
SolutionV
                                                                                                                                                                                                     (class
                                                                                                                                                                                                                                                                                                                                                       in
                                                            py_dss_interface.models.Solution.Solution), 79
                                                                                                                                                                                                                                                                                                                                                                                                                                                  80
SwtControls
                                                                                                                                                                                                             (class
                                                                                                                                                                                                                                                                                                                                                        in
                                                                                                                                                                                                                                                                                                                                                                                  SwtControlsI
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                 py\_dss\_interface.models.SwtControls.SwtControls),
                                                            py_dss_interface.models.SwtControls.SwtControls),
swtcontrols_all_names()
                                                                                                                                                                                                                                                                                                                                                                                   SwtControlsS
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            in
                                                               (py_dss_interface.models.SwtControls.SwtControls.SwtContpol_x\dds_interface.models.SwtControls.SwtControls),
                                                            method), 82
 swtcontrols_count()
                                                                                                                                                                                                                                                                                                                                                                                      SwtControlsV
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             in
                                                             (py_dss_interface.models.SwtControls.SwtControls.SwtControls),
                                                            method), 81
swtcontrols_first()
                                                               (py_dss_interface.models.SwtControls.SwtControls\).SwtControlsI
                                                            method), 81
                                                                                                                                                                                                                                                                                                                                                                                   Text (class in py_dss_interface.models.Text.Text), 82
 swtcontrols next()
                                                                                                                                                                                                                                                                                                                                                                                    text()
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     (py_dss_interface.models.Text.Text.Text
                                                               (py_dss_interface.models.SwtControls.SwtControls.SwtControls), 82
                                                            method), 81
                                                                                                                                                                                                                                                                                                                                                                                   Topology (class in py_dss_interface.models.Topology.Topology),
swtcontrols_read_action()
                                                                                                                                                                                                                                                                                                                                                                                                                                                  82
                                                               (py_dss_interface.models.SwtControls.SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtCo
                                                             method), 81
                                                                                                                                                                                                                                                                                                                                                                                                                                                    (py_dss_interface.models.Topology.TopologyI
swtcontrols_read_delay()
                                                                                                                                                                                                                                                                                                                                                                                                                                                  method), 82
                                                             (py_dss_interface.models.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtCo
                                                            method), 80
                                                                                                                                                                                                                                                                                                                                                                                                                                                    (py_dss_interface.models.Topology.Topology.TopologyI
 swtcontrols_read_is_locked()
                                                                                                                                                                                                                                                                                                                                                                                                                                                  method), 82
                                                               (py_dss_interface.models.SwtControls.SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtCo
                                                            method), 81
                                                                                                                                                                                                                                                                                                                                                                                                                                                    (py_dss_interface.models.Topology.Topology.TopologyV
 swtcontrols_read_name()
                                                                                                                                                                                                                                                                                                                                                                                                                                                  method), 83
                                                               (py_dss_interface.models.SwtControls.SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtCo
                                                            method), 81
                                                                                                                                                                                                                                                                                                                                                                                                                                                    (py_dss_interface.models.Topology.TopologyV
swtcontrols read switched obj()
                                                                                                                                                                                                                                                                                                                                                                                                                                                    method), 83
                                                               (py_dss_interface.models.SwtControls.SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtCo
                                                            method), 81
                                                                                                                                                                                                                                                                                                                                                                                                                                                  (py_dss_interface.models.Topology.TopologyV
swtcontrols_read_switched_term()
                                                                                                                                                                                                                                                                                                                                                                                                                                                  method), 83
                                                               (py_dss_interface.models.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtCo
                                                            method), 81
                                                                                                                                                                                                                                                                                                                                                                                                                                                    (py dss interface.models.Topology.TopologyI
 swtcontrols write action()
                                                                                                                                                                                                                                                                                                                                                                                                                                                  method), 82
                                                               (py_dss_interface.models.SwtControls.SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtControls_SwtCo
                                                            method), 81
                                                                                                                                                                                                                                                                                                                                                                                                                                                  method), 82
 swtcontrols_write_delay()
                                                                                                                                                                                                                                                                                                                                                                                    topology_first_load()
                                                               (py_dss_interface.models.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtCo
                                                            method), 80
                                                                                                                                                                                                                                                                                                                                                                                                                                                 method), 82
swtcontrols_write_is_locked()
                                                                                                                                                                                                                                                                                                                                                                                    topology_forward_branch()
                                                               (py_dss_interface.models.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtControls.SwtCo
                                                            method), 81
                                                                                                                                                                                                                                                                                                                                                                                                                                                 method), 82
 swtcontrols_write_name()
                                                                                                                                                                                                                                                                                                                                                                                   topology_looped_branch()
                                                             (py\_dss\_interface.models. SwtControls. Swt
                                                            method), 81
                                                                                                                                                                                                                                                                                                                                                                                                                                                  method), 82
```

```
topology_next() (py_dss_interface.models.Topology.Topology.Topology.Topology_interface.models.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Tr
                                                         method), 82
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 85
 topology_next_load()
                                                                                                                                                                                                                                                                                                                                                                 transformers_read_kv()
                                                           (py_dss_interface.models.Topology.TopologyI
                                                                                                                                                                                                                                                                                                                                                                                                                             (py_dss_interface.models.Transformers.Transformers.Transforme
                                                         method), 83
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 84
topology_num_isolated_branches()
                                                                                                                                                                                                                                                                                                                                                                transformers_read_kva()
                                                          (py_dss_interface.models.Topology.TopologyI
                                                                                                                                                                                                                                                                                                                                                                                                                            (py_dss_interface.models.Transformers.Transformers.Transforme
                                                         method), 83
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 84
topology_num_isolated_loadss()
                                                                                                                                                                                                                                                                                                                                                                 transformers_read_max_tap()
                                                           (py_dss_interface.models.Topology.TopologyI
                                                                                                                                                                                                                                                                                                                                                                                                                            (py\_dss\_interface.models.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Tran
                                                         method), 83
                                                                                                                                                                                                                                                                                                                                                                                                                             method), 84
topology_num_loops()
                                                                                                                                                                                                                                                                                                                                                                 transformers_read_min_tap()
                                                           (py_dss_interface.models.Topology.TopologyI
                                                                                                                                                                                                                                                                                                                                                                                                                            (py\_dss\_interface.models.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Tran
                                                         method), 83
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 84
topology_parallel_branch()
                                                                                                                                                                                                                                                                                                                                                                 transformers_read_name()
                                                           (py_dss_interface.models.Topology.TopologyI
                                                                                                                                                                                                                                                                                                                                                                                                                            (py\_dss\_interface.models.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Tran
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 86
                                                         method), 83
topology_read_branch_name()
                                                                                                                                                                                                                                                                                                                                                                 transformers_read_num_taps()
                                                           (py_dss_interface.models.Topology.Topology.TopologyS
                                                                                                                                                                                                                                                                                                                                                                                                                            (py_dss_interface.models.Transformers.Transformers.Transforme
                                                         method), 83
topology_read_bus_name()
                                                                                                                                                                                                                                                                                                                                                                transformers_read_num_windings()
                                                           (py_dss_interface.models.Topology.Topology.TopologyS
                                                                                                                                                                                                                                                                                                                                                                                                                            (py\_dss\_interface.models.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Tran
                                                         method), 83
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 85
                                                                                                                                                                                                                                                                                                                                                                 transformers_read_r()
topology_write_branch_name()
                                                           (py_dss_interface.models.Topology.Topology.TopologyS
                                                                                                                                                                                                                                                                                                                                                                                                                            (py_dss_interface.models.Transformers.Transformers.Transforme
                                                         method), 83
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 84
 topology_write_bus_name()
                                                                                                                                                                                                                                                                                                                                                                 transformers_read_r_neut()
                                                           (py_dss_interface.models.Topology.Topology.TopologyS
                                                                                                                                                                                                                                                                                                                                                                                                                            (py\_dss\_interface.models.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Tran
                                                         method), 83
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 84
                                                                                                                                                                                           (class
                                                                                                                                                                                                                                                                                                                                     in transformers_read_tap()
TopologyI
                                                         py_dss_interface.models.Topology.Topology),
                                                                                                                                                                                                                                                                                                                                                                                                                             (py_dss_interface.models.Transformers.Transformers.Transforme
                                                           82
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 84
TopologyS
                                                                                                                                                                                                                                                                                                                                                           transformers_read_wdg()
                                                         py_dss_interface.models.Topology.Topology),
                                                                                                                                                                                                                                                                                                                                                                                                                             (py_dss_interface.models.Transformers.Transformers.Transforme
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 85
                                                                                                                                                                                          (class
TopologyV
                                                                                                                                                                                                                                                                                                                                     in transformers_read_x_neut()
                                                         py_dss_interface.models.Topology.Topology),
                                                                                                                                                                                                                                                                                                                                                                                                                             (py_dss_interface.models.Transformers.Transformers.Transforme
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 84
 Transformers
                                                                                                                                                                                                       (class
                                                                                                                                                                                                                                                                                                                                       in transformers_read_xfmr_code()
                                                         py_dss_interface.models.Transformers.Transformers),
                                                                                                                                                                                                                                                                                                                                                                                                                            (py_dss_interface.models.Transformers.Transformers.Transforme
                                                                                                                                                                                                                                                                                                                                                                                                                           method), 86
transformers_all_Names()
                                                                                                                                                                                                                                                                                                                                                                transformers_read_xhl()
                                                           (py\_dss\_interface.models. Transformers. Tr
                                                         method), 86
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 84
transformers_count()
                                                                                                                                                                                                                                                                                                                                                                transformers_read_xht()
                                                           (py\_dss\_interface.models. Transformers. Tr
                                                         method), 85
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 84
transformers_first()
                                                                                                                                                                                                                                                                                                                                                                 transformers_read_xlt()
                                                           (py\_dss\_interface.models. Transformers. Tr
                                                          method), 85
                                                                                                                                                                                                                                                                                                                                                                                                                             method), 84
transformers_next()
                                                                                                                                                                                                                                                                                                                                                                 transformers_str_wdg_voltages()
                                                          (py_dss_interface.models.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transf
                                                         method), 85
                                                                                                                                                                                                                                                                                                                                                                                                                            method), 86
transformers_read_is_delta()
                                                                                                                                                                                                                                                                                                                                                                transformers_wdg_currents()
```

```
(py_dss_interface.models.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transf
                                       method), 86
                                                                                                                                                                                                                                                                                          method), 85
transformers_wdg_voltages()
                                                                                                                                                                                                                                                  TransformersF
                                                                                                                                                                                                                                                                                                                                                                                              (class
                                         (py_dss_interface.models.Transformers.Transformers.Transformers.Ynterface.models.Transformers.Transformers),
                                       method), 87
transformers write is delta()
                                                                                                                                                                                                                                                  TransformersI
                                                                                                                                                                                                                                                                                                                                                                                              (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  in
                                        (py_dss_interface.models.Transformers.Transformers.Transformers.Transformers),
                                       method), 86
transformers_write_kv()
                                                                                                                                                                                                                                                  TransformersS
                                                                                                                                                                                                                                                                                                                                                                                              (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   in
                                         (py_dss_interface.models.Transformers.Transformers.Transformers.Transformers),
                                        method), 84
transformers_write_kva()
                                                                                                                                                                                                                                                  TransformersV
                                                                                                                                                                                                                                                                                                                                                                                              (class
                                         (py_dss_interface.models.Transformers.Transformers.Transformers.Transformers),
                                       method), 84
transformers_write_max_tap()
                                        (py_dss_interface.models.Transformers.TransformersF
                                       method), 85
                                                                                                                                                                                                                                                  VSources (class in py_dss_interface.models.VSources.VSources),
transformers_write_min_tap()
                                         (py_dss_interface.models.Transformers.Transformers_Transformers_F_names()
                                                                                                                                                                                                                                                                                           (py\_dss\_interface.models.VSources.VSources.VSourcesV
transformers_write_name()
                                                                                                                                                                                                                                                                                           method), 88
                                        (py\_dss\_interface.models. Transformers. Tr
                                       method), 86
                                                                                                                                                                                                                                                                                           method), 88
transformers_write_num_taps()
                                                                                                                                                                                                                                                  vsources_first() (py_dss_interface.models.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSourc
                                         (py_dss_interface.models.Transformers.Transformers.Transfaenhort], 88
                                       method), 86
                                                                                                                                                                                                                                                  vsources_next() (py_dss_interface.models.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSources.VSource
transformers_write_num_windings()
                                                                                                                                                                                                                                                                                           method), 88
                                         (py_dss_interface.models.Transformers.Transformers_Transformers_d_angle_deg()
                                       method), 86
                                                                                                                                                                                                                                                                                           (py_dss_interface.models.VSources.VSourcesF
transformers_write_r()
                                                                                                                                                                                                                                                                                           method), 87
                                         (py_dss_interface.models.Transformers.TransformersoTransformersE_base_kv()
                                       method), 85
                                                                                                                                                                                                                                                                                           (py\_dss\_interface.models.VSources.VSources.VSourcesF
transformers_write_r_neut()
                                                                                                                                                                                                                                                                                           method), 87
                                         (py_dss_interface.models.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transf
                                        method), 85
                                                                                                                                                                                                                                                                                            (py_dss_interface.models.VSources.VSources.VSourcesF
transformers_write_tap()
                                                                                                                                                                                                                                                                                           method), 87
                                        (py_dss_interface.models.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transformers.Transf
                                       method), 85
                                                                                                                                                                                                                                                                                            (py_dss_interface.models.VSources.VSources.VSourcesS
transformers_write_wdg()
                                                                                                                                                                                                                                                                                            method), 88
                                        (py_dss_interface.models.Transformers.TransformersId_phases()
                                       method), 86
                                                                                                                                                                                                                                                                                            (py_dss_interface.models.VSources.VSources.VSourcesI
transformers_write_x_neut()
                                                                                                                                                                                                                                                                                           method), 88
                                         (py_dss_interface.models.Transformers.TransformersoTransformersE_pu()
                                       method), 85
                                                                                                                                                                                                                                                                                            (py_dss_interface.models.VSources.VSources.VSourcesF
transformers_write_xfmr_code()
                                                                                                                                                                                                                                                                                           method), 87
                                         (py_dss_interface.models.Transformers.Transformers.Transformers.\text{\faces} e_angle_deg()
                                                                                                                                                                                                                                                                                            (py_dss_interface.models.VSources.VSources.VSourcesF
                                       method), 86
transformers_write_xhl()
                                                                                                                                                                                                                                                                                           method), 87
                                         (py_dss_interface.models.Transformers.TransformersTransformersTe_base_kv()
                                                                                                                                                                                                                                                                                            (py_dss_interface.models.VSources.VSourcesF
transformers_write_xht()
                                                                                                                                                                                                                                                                                           method), 87
                                        (py_dss_interface.models.Transformers.Transformers.Transformers.Fte_frequency()
                                       method), 85
                                                                                                                                                                                                                                                                                            (py_dss_interface.models.VSources.VSources.VSourcesF
transformers_write_xlt()
                                                                                                                                                                                                                                                                                           method), 87
```

```
vsources write name()
                                                                                                                                                                                                                                    method), 90
                                 (py_dss_interface.models.VSources.VSources.VSources.Srves_read_y_scale()
                                                                                                                                                                                                                                    (py dss interface.models.XYCurves.XYCurves.XYCurvesF
                               method), 88
                                                                                                                                                                                                                                    method), 89
vsources_write_phases()
                                 (py_dss_interface.models.VSources.VSources.VSources_read_y_shift()
                               method), 88
                                                                                                                                                                                                                                    (py dss interface.models.XYCurves.XYCurvesF
                                                                                                                                                                                                                                    method), 89
vsources write pu()
                                 (py dss interface.models.VSources.VSources.VSources.Frves write name()
                               method), 87
                                                                                                                                                                                                                                    (py dss interface.models.XYCurves.XYCurves.XYCurvesS
                                                                                                                                                                                                                                    method), 90
VSourcesF
                                                                                                       (class
                                                                                                                                                                                     in
                               py_dss_interface.models.VSources.VSources),
                                                                                                                                                                                                  xycurves_write_npts()
                                87
                                                                                                                                                                                                                                    (py_dss_interface.models.XYCurves.XYCurves.XYCurvesI
                                                                                                      (class
                                                                                                                                                                                    in
                                                                                                                                                                                                                                    method), 89
VSourcesI
                               py_dss_interface.models.VSources.VSources),
                                                                                                                                                                                                  xycurves_write_x()
                                                                                                                                                                                                                                    (py_dss_interface.models.XYCurves.XYCurves.XYCurvesF
VSourcesS
                                                                                                       (class
                                                                                                                                                                                                                                    method), 89
                                                                                                                                                                                    in
                               py_dss_interface.models.VSources.VSources),
                                                                                                                                                                                                  xycurves_write_x_array()
                                88
                                                                                                                                                                                                                                    (py dss interface.models.XYCurves.XYCurves.XYCurvesV
                                                                                                      (class
                                                                                                                                                                                                                                    method), 90
VSourcesV
                               py_dss_interface.models.VSources.VSources),
                                                                                                                                                                                                  xycurves write x scale()
                                88
                                                                                                                                                                                                                                    (py_dss_interface.models.XYCurves.XYCurves.XYCurvesF
                                                                                                                                                                                                                                    method), 89
X
                                                                                                                                                                                                   xycurves_write_x_shift()
{\tt XYCurves} (class in py_dss_interface.models.XYCurves.XYCurves), (py_dss_interface.models.XYCurves.XYCurvesF
                                                                                                                                                                                                                                   method), 89
xycurves_count() (py_dss_interface.models.XYCurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXYECurvesXXECurvesXYECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurvesXXECurv
                                                                                                                                                                                                                                    (py\_dss\_interface.models.XYCurves.XYCurves.XYCurvesF
                               method), 89
xycurves_first() (py_dss_interface.models.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurv
                               method), 89
                                                                                                                                                                                                  xycurves_write_y_array()
\verb|xycurves_next()| (py\_dss\_interface.models.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XYCurves.XY
                               method), 89
                                                                                                                                                                                                                                    method), 90
                                                                                                                                                                                                   xycurves_write_y_scale()
xycurves_read_name()
                                 (py_dss_interface.models.XYCurves.XYCurves.XYCurvesS (py_dss_interface.models.XYCurves.XYCurves.XYCurvesF
                                                                                                                                                                                                                                    method), 89
                               method), 90
                                                                                                                                                                                                  xycurves_write_y_shift()
xycurves_read_npts()
                                (py\_dss\_interface.models. XYCurves. XYCurves. XYCurves I \quad (py\_dss\_interface.models. XYCurves. XYCurves. XYCurves I \quad (py\_dss\_interface.models. XYCurves. XYCurves. XYCurves I \quad (py\_dss\_interface.models. XYCurves. X
                               method), 89
                                                                                                                                                                                                                                   method), 89
                                                                                                                                                                                                  XYCurvesF
                                                                                                                                                                                                                                                                                                          (class
                                                                                                                                                                                                                                                                                                                                                                                       in
xycurves_read_x()
                                 (py_dss_interface.models.XYCurves.XYCurves,XYCurvesF py_dss_interface.models.XYCurves.XYCurves),
                               method), 89
                                                                                                                                                                                                  XYCurvesI
                                                                                                                                                                                                                                                                                                          (class
xycurves_read_x_array()
                                 (py dss interface.models.XYCurves.XYCurves.XYCurvesV py_dss_interface.models.XYCurves.XYCurves),
                               method), 90
                                                                                                                                                                                                                                                                                                          (class
                                                                                                                                                                                                  XYCurvesS
xycurves_read_x_scale()
                                                                                                                                                                                                                                                                                                                                                                                       in
                                 (py_dss_interface.models.XYCurves.XYCurves.XYCurvesF py_dss_interface.models.XYCurves.XYCurves),
                                                                                                                                                                                                                                    90
                               method), 89
                                                                                                                                                                                                  XYCurvesV
                                                                                                                                                                                                                                                                                                          (class
xycurves_read_x_shift()
                                 (py_dss_interface.models.XYCurves.XYCurves.XYCurvesF py_dss_interface.models.XYCurves.XYCurves),
                               method), 89
                                                                                                                                                                                                                                    90
xycurves_read_y()
                                (py_dss_interface.models.XYCurves.XYCurves.XYCurvesF
                               method), 89
xycurves_read_y_array()
                                 (py_dss_interface.models.XYCurves.XYCurves.XYCurvesV
```