Command

Contents

- OSCommand

class openstaadpy.os_analytical.oscommand.**OSCommand** [source]

Bases: object

CreateSteelDesignCommand(*NDesignCode: int, NCommandNo: int, IntValues: list[int], FloatValues: list, StringValues: list, NAssignList: list[int]*) [source]

Creates a steel design command for the specified design code and command number.

Parameters:

NDesignCode (*int*) –
Index of the design code to be used. Supported codes include:

Code Index	Description
1001	AASHTO ASD
1002	AISC ASD
1003	AS 4100-1998
1004	BS 5950-1:2000
1005	BS 5400:Part 3:1982
1006	CAN/CSA-S16-01
1007	French CM66 1977
1008	DIN 18 800 Part 1
1009	IS 800 1984, ASD
1010	Japan AIJ 2002
1011	AISC LRFD
1012	Norway NS 3472 2001
1014	Norway NPD 1993
1016	API 2A-WSD
1020	ASCE 10-97
1025	Russia SNiP 2.23-81* 1990
1027	Canada S136-94
1028	IS 801 1975
1029	IS 802 1995
1030	Mexico NTC 1987
1032	IS 800 2007, LSD

Code Index	Description
1052	IS 800 2007, WSD
1033	BS 5950-5:1998
1034	South Africa SANS 10162-1:2011
1044	AASHTO LRFD
1045	ANSI/AISC 360-05
1046-1053	ASME NF 3000 (various years)
1060	ASCE 52
1061	ANSI/AISC 360-10
1062	Canadian S16_09
1063	Russia SP 16.13330.2011
1064	South Africa SANS10162-1:1993
1065	Canadian S16_14
1066	NZS3404_1997
1067	ANSI/AISC 360-16
1068	AISI S100-2016
1069	Canadian S16_19
1102	AISC CASTELLATED
1202	AISC N690 1994
1204	AISC N690 1984
1210	Japan AIJ 2005
1220	BS EN 1993-1-1:2005

Code Index	Description
1221	Russia SP 16.13330.2017

NCommandNo : *int*

Index of the specific design command to be applied.

- **Table 1/10:**

Design Parameter	NCommandNo		Design Parameter	NCommand
AISC 360-05, ** **360-10,and 360-16			**AISC LRFD (2nd and 3rd Ed.)**	
CODE	9010		CODE	9010
ALH	9674		AXIS	9892
BEAM	9380		BEAM	9380
BRC	9955		CAN	9883
CAN	9883		CB	9280
CB	9280		DFF	9210
CSPACING	9103		DJ1	9390
DFF	9210		DJ2	9400
DJ1	9390		DMAX	9160
DJ2	9400		DMIN	9170
DMAX	9160		FLX	9781
DMIN	9170		FYLD	9100
DUCT	9964		FU	9705
E5P	9133		KX	9235
FLX	9781		KY	9240
FRM	9954		KZ	9250
FU	9705		LX	9125
FYLD	9100		LY	9130
IMM	9877		LZ	9140

Design Parameter	NCommandNo		Design Parameter	NCommand
KX	9235		MAIN	9330
INT	9957		NSF	9260
KY	9240		PROFILE	9520
KZ	9250		RATIO	9360
LBRC	9968		STIFF	9200
LEG	9500		STP	9894
LX	9125		TMAIN	9335
LY	9130		TRACK	9350
LZ	9140		UNB	9660
MAIN	9330		UNT	9650
METHOD	9923			
MTYP	9956			
NBRC	9972			
NSF	9260			
PROFILE	9520			
RATIO	9360			
SEISMIC	9953			
SGR	9460			
SLF	9922			
SNUG	9102			

Design Parameter	NCommandNo		Design Parameter	NCommand
SOE	9975			
SRT	9131			
STB	9973			
STT	9974			
STIFF	9200			
STP	9894			
TBRC	9969			
TFA	9976			
TSL	9132			
TMAIN	9335			
TND	9959			
TORSION	9670			
TRACK	9350			
UNB	9660			
UNL	9150			
UNR	9970			
UNT	9650			
WTYPE	9958			

- **Table 2/10:**

Design Parameter	NCommandNo		Design Parameter	NCommand
AISI 2016			**API 2A-WSD**	
CODE	9010		CODE	9010
AXIS	9892		BEAM	9380
BEAM	9380		CB	9280
CAN	9883		CMY	9310
DFF	9210		CMZ	9320
DJ1	9390		DMAX	9160
DJ2	9400		DMIN	9170
FLX	9781		FSJ	9503
FU	9705		FYLD	9100
FYLD	9100		KY	9240
KT	9805		KZ	9250
KY	9240		LY	9130
KZ	9250		LZ	9140
LT	9806		MAIN	9330
LY	9130		NSF	9260
LZ	9140		RATIO	9360
METHOD	9923		SSY	9290
NSF	9260		SSZ	9300
RATIO	9360		TRACK	9350

Design Parameter	NCommandNo		Design Parameter	NCommand
SSY	9290		UNF	9270
SSZ	9300		UNL	9150
STIFF	9200		WELD	9370
TRACK	9350		WMIN	9180
TSA	9782		WSTR	9110
UNB	9660			
UNT	9650			

- **Table 3/10:**

Design Parameter	NCommandNo		Design Parameter	NCommand
ANSI/AISC N690 1994			**ANSI/AISC N690 1984**	
CODE	9010		CODE	9010
BEAM	9380		CAN	9883
CAN	9883		CB	9280
CB	9280		CMY	9310
CMY	9310		CMZ	9320
CMZ	9320		CT	9924
COMPOSITE	9710		DFF	9210
CONDIA	9715		DJ1	9390
CONHEIGHT	9720		DJ2	9400
CYCLES	9725		DMAX	9160
DFF	9210		DMIN	9170
DJ1	9390		FU	9705
DJ2	9400		FYLD	9100
DLR2	9735		KY	9240
DLRATIO	9730		KZ	9250
DMAX	9160		LY	9130
DMIN	9170		LZ	9140
EFFWIDTH	9740		MAIN	9330
FYLD	9100		NSF	9260

Design Parameter	NCommandNo		Design Parameter	NComman
FPC	9745		PROFILE	9520
FSS	9882		RATIO	9360
FU	9705		SFC	9942
FYLD	9100		SFT	9943
KX	9235		SMY	9945
KY	9240		SMZ	9944
KZ	9250		STIFF	9200
LX	9125		STYPE	9101
LY	9130		TMAIN	9335
LZ	9140		TRACK	9350
MAIN	9330		UNB	9660
NSF	9260		UNT	9650
OVR	9880			
PLTHICK	9755			
PLTWIDTH	9760			
PROFILE	9520			
RATIO	9360			
RIBHEIGHT	9765			
RIBWIDTH	9770			
SFC	9942			

Design Parameter	NCommandNo		Design Parameter	NComman
SFT	9943			
SHE	9893			
SHORING	9775			
SLABTHICK	9780			
SMY	9945			
SMZ	9944			
SSY	9290			
SSZ	9300			
STIFF	9200			
STYPE	9101			
TAPER	9868			
TMAIN	9335			
TORSION	9670			
TRACK	9350			
UNB	9660			
UNT	9650			
WELD	9370			
WMAX	9881			
WMIN	9180			
WSTR	9110			

- **Table 4/10:**

Design Parameter	NCommandNo		Design Parameter	NCommand
BS 5950			**BS 5400**	
CODE	9010		CODE	9010
AD	9902		ESTIFF	9690
BEAM	9381		KY	9240
CAN	9883		KZ	9250
CB	9281		LY	9130
DFF	9210		LZ	9140
DJ1	9390		MAIN	9332
DJ2	9400		NSF	9260
DMAX	9160		PY	9120
DMIN	9170		RATIO	9360
ESTIFF	9690		SBLT	9472
KY	9240		SGR	9464
KZ	9250		TRACK	9352
LEG	9501		UNL	9150
LVV	9190		WET	9510
LY	9130			
LZ	9140			
MLT	9820			
MX	9840			

Design Parameter	NCommandNo		Design Parameter	NCommand
MY	9850			
MYX	9830			
NSF	9260			
PNL	9700			
PY	9120			
MAIN	9331			
RATIO	9360			
SAME	9870			
SBLT	9472			
SWAY	9860			
SGR	9464			
TB	9530			
TRACK	9357			
UNF	9270			
UNL	9150			
WELD	9371			

Design Parameter	NCommandNo		Design Parameter	NCommand

- **Table 5/10:**

Design Parameter	NCommandNo		Design Parameter	NCommand
Canada S136-94			**French**	
CODE	9010		CODE	9010
BEAM	9380		BEAM	9385
CMZ	9790		C1	9550
CMY	9795		C2	9560
CWY	9783		DFF	9210
DMAX	9160		DJ1	9390
DMIN	9170		DJ2	9400
FLX	9781		DMAX	9160
FU	9705		DMIN	9170
FYLD	9100		FYLD	9100
KT	9805		KY	9240
KY	9240		KZ	9250
KZ	9250		LY	9130
LT	9806		LZ	9140
LY	9130		NSF	9260
LZ	9140		RATIO	9360
NSF	9260		SAME	9870
STIFF	9200		TRACK	9355
TRACK	9350		UNF	9270

Design Parameter	NCommandNo		Design Parameter	NCommand
TSA	9782		UNL	9150

Design Parameter	NCommandNo		Design Parameter	NCommand

- **Table 6/10:**

Design Parameter	NCommandNo		Design Parameter	NCommand
Indian 800 1984			**Indian 801 1975**	
CODE	9010		CODE	9010
BEAM	9380		BEAM	9380
CMY	9310		CB	9280
CMZ	9320		CMY	9310
DFF	9210		CMZ	9320
DJ1	9390		CWY	9783
DJ2	9400		FLX	9781
DMAX	9160		FU	9705
DMIN	9170		FYLD	9100
FYLD	9100		KX	9235
KY	9240		KY	9240
KZ	9250		KZ	9250
LY	9130		LX	9125
LZ	9140		LY	9130
MAIN	9334		LZ	9140
NSF	9260		NSF	9260
PROFILE	9520		RATIO	9360
RATIO	9360		TRACK	9350
SSY	9290		TSA	9782

Design Parameter	NCommandNo		Design Parameter	NCommand
SSZ	9300		UNL	9270
TMAIN	9335			
TRACK	9350			
UNF	9270			
UNL	9150			

- **Table 7/10:**

Design Parameter	NCommandNo		Design Parameter	NCommand
Japanese AIJ 2002			**Mexian**	
CODE	9010		CODE	9010
BEAM	9380		BEAM	9380
CAN	9883		CB	9280
CB	9280		CMB	9878
DFF	9210		DFF	9210
DJ1	9390		DJ1	9390
DJ2	9400		DJ2	9400
DMAX	9160		DMAX	9160
DMIN	9170		DMIN	9170
FYLD	9100		DSD	9879
KY	9240		FU	9705
KZ	9250		FYLD	9100
LY	9130		IMM	9877
LZ	9140		INO	9876
MAIN	9336		IRR	9875
MBG	9823		KX	9235
MISES	9819		KY	9240
NSF	9260		KZ	9250
RATIO	9360		LDR	9874

Design Parameter	NCommandNo		Design Parameter	NCommand
SLF	9922		LX	9125
TMAIN	9335		LY	9130
TRACK	9358		LZ	9140
UNF	9270		NSF	9260
UNL	9150		RATIO	9360
YNG	9824		STIFF	9200
			TRACK	9350
			UNB	9660
			UNT	9650

- **Table 8/10:**

Design Parameter	NCommandNo		Design Parameter	NCommand
Norwegian NORSOK N-004			**Russian SNip 81**	
CODE	9010		CODE	9010
BEAM	9380		BEAM	9380
CMY	9310		CB	9281
CMZ	9320		CMM	9494
DFF	9210		CMN	9494
DJ1	9390		DFF	9210
DJ2	9400		DMAX	9160
DMAX	9160		DMIN	9170
DMIN	9170		GAMC1	9809
FYLD	9100		GAMC2	9810
FU	9705		GMF	9977
HYD	9940		KY	9240
KY	9240		KZ	9250
KZ	9250		LEG	9502
LX	9125		LY	9130
LY	9130		LZ	9140
LZ	9140		MAIN	9333
MAIN	9330		NSF	9260
PSD	9941		PY	9121

Design Parameter	NCommandNo		Design Parameter	NCommand
RATIO	9360		RATIO	9360
SGR	9466		SBLT	9471
TMAIN	9335		SGR	9463
TRACK	9351		TB	9952
			TRACK	9356
			UNL	9151

- **Table 9/10:**

Design Parameter	NCommandNo		Design Parameter	NCommand
EC3			**Australian**	
CODE	9010		CODE	9010
ALH	9674		ALB	9804
ALPHA	9926		ALM	9803
BEAM	9382		BEAM	9380
BETA	9382		DFF	9210
C1	9550		DJ1	9390
C2	9560		DJ2	9400
C3	9565		DMAX	9160
CAN	9883		DMIN	9170
CMM	9482		FU	9705
CMN	9492		FYLD	9100
CMT	9671		IST	9800
DFF	9210		KT	9805
DJ1	9390		KY	9240
DJ2	9400		KZ	9250
DMAX	9160		LHT	9938
DMIN	9170		LX	9125
EFT	9673		LY	9130
ELB	9697		LZ	9140

Design Parameter	NCommandNo		Design Parameter	NCommand
ESTIFF	9690		MAIN	9330
FAB	9508		NSC	9802
FU	9705		NSF	9260
GM0	9504		PBRACE	9939
GM1	9505		PHI	9801
GM2	9506		RATIO	9360
GST	9676		SGR	9466
HGT	9720		SKL	9808
KC	9696		SKR	9809
KY	9240		SKT	9807
KZ	9250		TMAIN	9335
LEG	9501		TRACK	9350
LVV	9190		TSP	9932
LY	9130		UNB	9660
LZ	9140		UNT	9650
MTH	9677			
MU	9695			
NA	9465			
NSF	9260			
PLG	9930			

Design Parameter	NCommandNo		Design Parameter	NCommand
PY	9120			
RATIO	9360			
SBLT	9472			
SGR	9461			
STIFF	9202			
TOM	9672			
TORSION	9688			
TRACK	9353			
UNF	9270			
UNL	9150			
ZG	9507			

- **Table 10/10:**

Design Parameter	NCommandNo		Design Parameter	NCommand
Russian SP 2011			**Russian SP 2017**	
CODE	9010		CODE	9010
BEAM	9380		BMT	9951
BMT	9951		CB	9281
CB	9281		CMM	9494
CMM	9494		CMN	9494
CMN	9494		DFF	9210
DFF	9210		DJ1	9390
DJ1	9390		DJ2	9400
DJ2	9400		DMAX	9160
DMAX	9160		DMIN	9170
DMIN	9170		ENSGR	9818
ENMAIN	9817		FU	9705
ENSGR	9818		GAMC1	9809
GAMC1	9809		GAMC2	9810
GAMC2	9810		GAMM	9978
GAMM	9815		GMF	9977
GMF	9977		KY	9240
KY	9240		KZ	9250
KZ	9250		LEG	9502

Design Parameter	NCommandNo		Design Parameter	NCommand
LEG	9502		LY	9130
LY	9130		LZ	9140
LZ	9140		NSF	9260
MAIN	9333		PY	9121
NSF	9260		SBLT	9471
PY	9121		STP	9894
RATIO	9360		TB	9952
SBLT	9471		TRACK	9356
SGR	9463			
TB	9952			
TRACK	9356			
UNL	9151			

IntValues : *list of int*

Integer parameter values for the design command.

FloatValues : *list of float*

Floating-point parameter values for the design command.

StringValues : *list of str*

String parameter values for the design command.

NAssignList : *list*

List of reference IDs of objects to which the design command should be assigned.

Return type:

None

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.CreateSteelDesignCommand(1, 9010, [9380, 9310, 93
```

DeleteAllAnalysisCommands() [source]

Delete all analysis commands from the current STAAD model.

Returns:

Status code ‘1’ for OK and ‘0’ if failed to delete.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> result = staad_obj.Command.DeleteAllAnalysisCommands()
>>> print(result)
```

DeleteCheckIrregularitiesCommand() [source]

Delete the check irregularities command.

Returns:

Status code ‘1’ for OK and ‘0’ if failed to delete.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> result = staad_obj.Command.DeleteCheckIrregularitiesCommand()
>>> print(result)
```

DeleteCheckSoftStoryCommand() [source]

Delete the check soft story command.

Returns:

Status code ‘1’ for OK and ‘0’ if failed to delete.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> result = staad_obj.Command.DeleteCheckSoftStoryCommand()
>>> print(result)
```

DeleteFloorDiaphragmBaseCommand() [source]

Delete the floor diaphragm base command.

Returns:

Status code ‘1’ for OK and ‘0’ if failed to delete.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> status = staad_obj.Command.DeleteFloorDiaphragmBaseCommand()
>>> print(status)
```

PerformAnalysis(*printOption: int*) [source]

Perform a standard analysis on the current STAAD model.

Parameters:

printOption (*int*) –

The print option for the analysis results. Possible values are:

Value	Print Option
1	Print Load Data
2	Print Statics Check
3	Print Statics Load
4	Print Mode Shapes
5	Print Both
6	Print All
0	No Print

Return type:

None

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.PerformAnalysis()
```

PerformBucklingAnalysis(*MaxNoOfIterations: int, PrintOption: int*)

Creates the commands required to perform BUCKLING ANALYSIS. [source]

Parameters:

- **iNoOfIterations** (*int*) – Maximum number of iterations desired.

- **iPrintOption** (*int*) –

  Option for specifying the print output. Choose from the following:

Value	Print Option
1	Print Load Data
2	Print Statics Check
3	Print Statics Load
5	Print Both
6	Print All
0	No Print

Return type:

None

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.PerformBucklingAnalysis(5, 6)
```

PerformBucklingAnalysisEx(*Method: int, MaxNoOfIterations: int, PrintOption: int*) [source]

Assigns the commands required to perform a BUCKLING ANALYSIS on the model. This requires the presence of nonlinear cables in the structure.

Parameters:

- **Method** (*int*) –

  Buckling Analysis method to use.

Value	Method
0	BucklingAnalysisMethod.Iterative
1	BucklingAnalysisMethod.Eigen

- **MaxNoOfIterations** (*int*) – Maximum number of iterations desired. Default is 10; 15 is recommended. This is used only for the Basic Solver.
- **PrintOption** (*int*) –

  Option index for specifying the print output. Choose from the following:

Value	Print Option
1	Print Load Data
2	Print Statics Check
3	Print Statics Load
4	Reserved (do not use)
5	Print Both
6	Print All
0	No Print

Returns:

True if the command was successfully added or updated, *False* otherwise.

Return type:

bool

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> result = staad_obj.Command.PerformBucklingAnalysisEx(0, 15, 6)
```

PerformCableAnalysis(*NoOfIterations: int, PrintOption: int*)

Creates the commands required to perform a CABLE ANALYSIS. This requires the presence of cable members in the structure. [source]

> **Note**
>
> The advanced algorithm will be used only if an Advanced license is enabled. For further details, refer to TR.37.3 of the STAAD.Pro Help manual.

Parameters:

- **iNoOfIterations** (*int*) – Desired number of iterations.
- **iPrintOption** (*int*) –

  Option for specifying the print output. Choose from the following:

Value	Print Option
1	Print Load Data
2	Print Statics Check
3	Print Statics Load
5	Print Both
6	Print All
0	No Print

Return type:

None

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.PerformCableAnalysis(25, 6)
```

`PerformCableAnalysisEx(AdvancedCableAnalysis: int, AdvOptions: list, Params: list, PrintOption: int)` [source]

Assigns the commands required to perform a CABLE ANALYSIS on the model. This requires the presence of nonlinear cables in the structure.

Parameters:

- **AdvancedCableAnalysis** (*int*) – Whether to perform Advanced Cable Analysis (*True* = 1 or *False* = 0).
- **AdvOptions** (*list of int*) –

  Additional options for Advanced Cable Analysis.

Array Index	Default	STAAD Command	Description
0	1	REFORM f11	Use Full Newton-Raphson method? 0 = False, 1 = True
1	0	KGEOM f12	Use Geometric Matrix (Kg)? 0 = False, 1 = True

  Notes: - If array size = 1, second value defaults to *False*. - If array size > 2, only the first two values are used.
- **Params** (*list of float*) –

  Additional parameters required for Cable Analysis. Some values represent integer flags.
  For **Basic Cable Analysis**:

Array Index	Default Value	STAAD Command	Description
0	145	STEPS f1	The number of load steps. The applied loads will be applied gradually in this many steps. Each step will be iterated to convergence. Should be in the range 5 to 145, with 145 as the default number.
1	300	EQITERATIONS f2	Maximum number of iterations permitted in each load step. Should be in the range of 10 to 500
2	1.0E-4	EQTOLERANCE f3	The convergence tolerance for the above iterations
3	0.0	SAGMINIMUM f4	Sag Minimum. Default is 0.0
4	1.0	STABILITY f5	A stiffness matrix value to be added to the global matrix at each translational direction for joints connected to cables and nonlinear trusses for the first (f6) Load Steps. Should be within the range of 0.0 to 1000.0
5	1	f6	The number of load steps over which the Stability stiffness matrix(f5) is gradually applied
6	0.0	KSMALL f7	A stiffness matrix value to be added to the global matrix at each translational direction for joints connected to cables and

Array Index	Default Value	STAAD Command	Description
			nonlinear trusses for every load step. Should be within the range of 0.0 and 1.0

For **Advanced Cable Analysis**:

Array Index	Default Value	STAAD Command	Description
0	1	STEPS f1	Number of load increments.
1	300	EQITERATIONS f2	Max iterations per increment.
2	1.0E-6	EQTOLERANCE f3	Convergence tolerance for residual force norm.

- **PrintOption** (*int*) –

  Option index for specifying the print output. Choose from the following:

Value	Print Option
1	Print Load Data
2	Print Statics Check
3	Print Statics Load
5	Print Both
6	Print All
0	No Print

Returns:

True if the command was successfully added or updated, *False* otherwise.

Return type:

bool

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> result = staad_obj.Command.PerformCableAnalysisEx(1, [0, 1], [125,
>>> print(result)
```

PerformDirectAnalysis(*Option: int, Params: list, AddOptions: list, PrintOption: int*) [source]

Assigns the commands required to perform a DIRECT ANALYSIS for AISC on the model.

Parameters:

- **Option** (*DirectAnalysisOption*) –

  Direct Analysis method. Default is *DirectAnalysisOption.LRFD*.

Value	Method
1	DirectAnalysisOption.LRFD
2	DirectAnalysisOption.ASD

- **Params** (*list of float*) –

  Additional parameters required for Direct Analysis. Some values represent integer flags.

Array Index	Default Value	STAAD Command	Description
0	0.01	TAUTOL f1	Tau-b tolerance (typically 0.001 to 1.0).
1	0.01	DISPTOL f2	Displacement tolerance: - 0.01 in (displacement) - 0.01 radians (rotation) The value should not be too tight
2	1	ITERDIRECT i3	Max number of iterations (typically 1 to 10).
3	15	PDiter i5	Iterations for PDelta with SmallDelta (5 to 15) (max 15).

- **AddOptions** (*list of int*) –

  Additional boolean options for Direct Analysis.

Array Index	Default	STAAD Command	Description
0	0	REDUCEDEI i4	Use reduced EI (Tau-b * 0.8 * EI)? 0 = False, 1 = True
1	0	TBITER	Iterate Tau-b? 0 = False, 1 = True

  Notes: - If array size = 1, second value defaults to *False*. - If array size > 2, only the first two values are used.

- **PrintOption** (*int*) –

  Option index for specifying the print output. Choose from the following:

Value	Print Option
1	Print Load Data
2	Print Statics Check
3	Print Statics Load
4	Print Mode Shapes
5	Print Both
6	Print All
0	No Print

Returns:

True if the command was successfully added or updated, *False* otherwise.

Return type:

bool

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> result = staad_obj.Command.PerformDirectAnalysis(1, [0.015, 0.02, 7
>>> print(result)
```

PerformNonlinearAnalysisEx(*PrintOption: int, Arclength: float, NoOfIterations: int, Tolerance: float, Steps: int, Rebuild: int, AddGeometricStiffness: int, DispLimitData: list*) [source]

Assigns the commands required to perform NONLINEAR ANALYSIS of the current structure with the specified print option.

Parameters:

- **PrintOption** (*int*) –

  Option index for specifying the print output. Choose from the following:

Value	Print Option
1	Print Load Data
2	Print Statics Check
3	Print Statics Load
4	Print Mode Shapes
5	Print Both
6	Print All
0	No Print

- **Arclength** (*float*) – Absolute displacement limit for the first analysis step for Displacement Control. Set to 0 for no Displacement Control.
- **NoOfIterations** (*int*) – Maximum number of iterations to achieve equilibrium in the deformed position within the specified tolerance.
- **Tolerance** (*float*) – Tolerance value for determining convergence.
- **Steps** (*int*) – Number of load steps. The load is applied incrementally in these stages.
- **Rebuild** (*int*) –

  Frequency of rebuilding the Tangent or Stiffness Matrix (K) per load step and iteration:

  - 0 : Once per load step
  - 1 : Every load step and iteration
- **AddGeometricStiffness** (*int*) – Whether to add the geometric stiffness matrix to the stiffness matrix (K). Set to *1* to include, or *0* to exclude.
- **varDispLimitData** (*list of float*) –

  Displacement limit data to specify the target displacement. The array should follow this structure:

Array Index	Default	STAAD Command	Description
0	1	JOINT_TARGET i1	Joint being monitored in a displacement target analysis
1	1	i2	Global degree of freedom (1–6): 1: Global X, 2: Global Y, 3: Global Z, 4: Moment about Global X, 5: Moment about Global Y, 6: Moment about Global Z
2	0	f1	Displacement target value in current length units

Returns:

True if the command was successfully added or updated, *False* otherwise.

Return type:

Int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> result = staad_obj.Command.PerformNonlinearAnalysisEx(6, 0, 5, 0.00
>>> print(result)
```

`PerformPDeltaAnalysisEx(NoOfIterations: int, PrintOption: int, bSmallDelta: int, AddGeometricStiffness: int)` [source]

Perform an extended P-Delta analysis with options.

Parameters:

- **NoOfIterations** (*int*) – Desired number of iterations.
- **PrintOption** (*int*) –

Option index for specifying the print output. Choose from the following:

Value	Print Option
1	Print Load Data
2	Print Statics Check
3	Print Statics Load
4	Print Mode Shapes
5	Print Both
6	Print All
0	No Print

- **bSmallDelta** (*int*) – Set to 1 to include P-SMALL-Delta effect, or 0 to include P-LARGE-Delta effect.
- **AddGeometricStiffness** (*int*) – Set to 1 to add geometric stiffness, or 0 to exclude it.

Return type:

None

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.PerformPDeltaAnalysisEx(20, 5, 0, 1)
```

PerformPDeltaAnalysisNoConverge(*NoOfIterations: int*, *PrintOption: int*) [source]

Perform a P-Delta analysis without convergence check.

Parameters:

- **NoOfIterations** (*int*) – The number of iterations to perform.
- **PrintOption** (*int*) –

The print option for the analysis results. Possible values are:

Value	Print Option
1	Print Load Data
2	Print Statics Check
3	Print Statics Load
4	Print Mode Shapes
5	Print Both
6	Print All
0	No Print

Return type:

None

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.PerformPDeltaAnalysisNoConverge(5, 1)
```

SetCheckIrregularitiesCommand(*DesignCode*) [source]

Set a check irregularities command.

Parameters:

DesignCode (*int*) – Index number for specifying code

Returns:

Status code ‘1’ for OK and ‘0’ if failed to add or update.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> result = staad_obj.Command.SetCheckIrregularitiesCommand(3)
>>> print(result)
```

SetCheckSoftStoryCommand(*DesignCode: int*) [source]

Set a check soft story command.

Parameters:

DesignCode (*int*) – Index number for specifying code

Returns:

Status code ‘1’ for OK and ‘0’ if failed to add or update

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> result = staad_obj.Command.SetCheckSoftStoryCommand(3)
>>> print(result)
```

SetFloorDiaphragmBaseCommand(*baseElevationValue: float*) [source]

Set a floor diaphragm base command.

Parameters:

baseElevationValue (*int*) – The diaphragm base command string.

Returns:

Status code ‘1’ for OK and ‘0’ if failed to add or update

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> status = staad_obj.Command.SetFloorDiaphragmBaseCommand(100.0)
>>> print(status)
```

__init__(*staadObj*) [source]