



```
#-----
# Copyright (c) Bentley Systems, Incorporated. All rights reserved.
# See COPYRIGHT.md in the repository root for full copyright notice
#-----
from .openStaadHelper import *
from comtypes import CoInitialize
```

```
class OSCommand: \[docs\]
    CoInitialize()
```

```
def __init__(self, staadObj): \[docs\]
    self._staad = staadObj
    self._command = self._staad.Command

    self._functions= [
        "PerformAnalysis",
        "PerformPDeltaAnalysisNoConverge",
        "CreateSteelDesignCommand",
        "PerformCableAnalysis",
        "PerformBucklingAnalysis",
        "SetFloorDiaphragmBaseCommand",
        "DeleteFloorDiaphragmBaseCommand",
        "SetCheckSoftStoryCommand",
        "DeleteCheckSoftStoryCommand",
        "SetCheckIrregularitiesCommand",
        "DeleteCheckIrregularitiesCommand",
        "PerformBucklingAnalysisEx",
        "PerformCableAnalysisEx",
        "PerformDirectAnalysis",
        "PerformNonlinearAnalysisEx",
        "DeleteAllAnalysisCommands",
        "PerformPDeltaAnalysisEx"
    ]

    for function_name in self._functions:
        self._command._FlagAsMethod(function_name)
```

```
def PerformAnalysis(self, printOption:int): \[docs\]
    """
    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

### Returns

-----

None

### Examples

-----

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.PerformAnalysis()
"""
self._command.PerformAnalysis(printOption)
```

[\[docs\]](#)

```
def PerformPDeltaAnalysisNoConverge(self, NoOfIterations:int, PrintOption:...
```

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	
+-----+	+-----+	+-----+

Returns

-----

None

Examples

-----

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.PerformPDeltaAnalysisNoConverge(5, 1)
"""
self._command.PerformPDeltaAnalysisNoConverge(NoOfIterations, PrintOptio
```

[\[docs\]](#)

```
def CreateSteelDesignCommand(self, NDesignCode:int, NCommandNo:int, IntValue
```

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

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
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
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	NCommandNo	De Pa
*****				
**AISC 360-05, **		**AISC LRFD (2nd and**		**
**360-10, and 360-16**		**3rd Ed.）**		
CODE	9010	CODE	9010	CO
ALH	9674	AXIS	9892	BE
BEAM	9380	BEAM	9380	CE
BRC	9955	CAN	9883	CM
CAN	9883	CB	9280	CM
CB	9280	DFF	9210	DF
CSPACING	9103	DJ1	9390	DI
DFF	9210	DJ2	9400	DI
DJ1	9390	DMAX	9160	DM
DJ2	9400	DMIN	9170	DM
DMAX	9160	FLX	9781	FY
DMIN	9170	FYLD	9100	KY
DUCT	9964	FU	9705	KZ
ESP	9133	KX	9235	LY
FLX	9781	KY	9240	LZ
FRM	9954	KZ	9250	MA
FU	9705	LX	9125	NS
FYLD	9100	LY	9130	PE
IMM	9877	LZ	9140	PU

KX	9235	MAIN	9330	RA
+-----+	+-----+	+-----+	+-----+	+-----+
INT	9957	NSF	9260	SS
+-----+	+-----+	+-----+	+-----+	+-----+
KY	9240	PROFILE	9520	SS
+-----+	+-----+	+-----+	+-----+	+-----+
KZ	9250	RATIO	9360	ST
+-----+	+-----+	+-----+	+-----+	+-----+
LBRC	9968	STIFF	9200	TF
+-----+	+-----+	+-----+	+-----+	+-----+
LEG	9500	STP	9894	UN
+-----+	+-----+	+-----+	+-----+	+-----+
LX	9125	TMAIN	9335	UN
+-----+	+-----+	+-----+	+-----+	+-----+
LY	9130	TRACK	9350	WS
+-----+	+-----+	+-----+	+-----+	+-----+
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			
+-----+	+-----+	+-----+	+-----+	+-----+
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	NCommandNo			Design	NCommandNo		Desi
Parameter				Parameter			Para
+=====+		+=====+		+=====+		+=====+	
**AISI 2016**				**API 2A-WSD**			
+-----+		+-----+		+-----+		+-----+	
CODE	9010			CODE	9010		CODE
+-----+		+-----+		+-----+		+-----+	
AXIS	9892			BEAM	9380		BEAM
+-----+		+-----+		+-----+		+-----+	
BEAM	9380			CB	9280		CMY
+-----+		+-----+		+-----+		+-----+	
CAN	9883			CMY	9310		CMZ
+-----+		+-----+		+-----+		+-----+	
DFF	9210			CMZ	9320		DMAX
+-----+		+-----+		+-----+		+-----+	
DJ1	9390			DMAX	9160		DBL
+-----+		+-----+		+-----+		+-----+	
DJ2	9400			DMIN	9170		DMIN
+-----+		+-----+		+-----+		+-----+	
FLX	9781			FSJ	9503		ELA
+-----+		+-----+		+-----+		+-----+	
FU	9705			FYLD	9100		ELB
+-----+		+-----+		+-----+		+-----+	
FYLD	9100			KY	9240		FVB
+-----+		+-----+		+-----+		+-----+	
KT	9805			KZ	9250		FYB
+-----+		+-----+		+-----+		+-----+	
KY	9240			LY	9130		FYLD
+-----+		+-----+		+-----+		+-----+	
KZ	9250			LZ	9140		KT
+-----+		+-----+		+-----+		+-----+	
LT	9806			MAIN	9330		KY



LY	9130	NSF	9260	KZ
LZ	9140	RATIO	9360	LEG
METHOD	9923	SSY	9290	LT
NSF	9260	SSZ	9300	LY
RATIO	9360	TRACK	9350	LZ
SSY	9290	UNF	9270	MAIN
SSZ	9300	UNL	9150	NHL
STIFF	9200	WELD	9370	NSF
TRACK	9350	WMIN	9180	RAT
TSA	9782	WSTR	9110	SSY
UNB	9660			SSZ
UNT	9650			TRAC
				UNB
				UNT

- Table 3/10:

Design Parameter	NCommandNo	Design Parameter	NCommandNo	Desi Para
**ANSI/AISC N690 1994**		**ANSI/AISC N690 1984**		**A
CODE	9010	CODE	9010	CODE
BEAM	9380	CAN	9883	CB
CAN	9883	CB	9280	CMZ
CB	9280	CMY	9310	EOPE
CMY	9310	CMZ	9320	FYLD
CMZ	9320	CT	9924	RAT

COMPOSITE	9710		DFF	9210		SOPE
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
CONDIA	9715		DJ1	9390		TRAC
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
CONHEIGHT	9720		DJ2	9400		UNL
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
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		
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
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				
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	NCommandNo	Design	NCommandNo	Design
Parameter		Parameter		Parameter

**BS 5950**		**BS 5400**		**Ca	
CODE	9010	CODE	9010	CODE	
AD	9902	ESTIFF	9690	BEAM	
BEAM	9381	KY	9240	CB	
CAN	9883	KZ	9250	CMY	
CB	9281	LY	9130	CMZ	
DFF	9210	LZ	9140	DFF	
DJ1	9390	MAIN	9332	DJ1	
DJ2	9400	NSF	9260	DJ2	
DMAX	9160	PY	9120	DMAX	
DMIN	9170	RATIO	9360	DMIN	
ESTIFF	9690	SBLT	9472	FU	
KY	9240	SGR	9464	FYLD	
KZ	9250	TRACK	9352	KT	
LEG	9501	UNL	9150	KY	
LVV	9190	WET	9510	KZ	
LY	9130			LT	
LZ	9140			LY	
MLT	9820			LZ	
MX	9840			MAIN	
MY	9850			NSF	
MYX	9830			RATIO	
NSF	9260			SHEAR	
PNL	9700			SSY	
PY	9120			SSZ	
MAIN	9331			TRACK	
RATIO	9360			UNB	
SAME	9870			UNT	

SBLT	9472				
SWAY	9860				
SGR	9464				
TB	9530				
TRACK	9357				
UNF	9270				
UNL	9150				
WELD	9371				

- Table 5/10:

Design Parameter	NCommandNo	Design Parameter	NCommandNo	Desi Para
**Canada S136-94**		**French**		**Ge
CODE	9010	CODE	9010	CODE
BEAM	9380	BEAM	9385	BEAM
CMZ	9790	C1	9550	CB
CMY	9795	C2	9560	CMM
CWY	9783	DFF	9210	DMA
DMAX	9160	DJ1	9390	DMIN
DMIN	9170	DJ2	9400	KY
FLX	9781	DMAX	9160	KZ
FU	9705	DMIN	9170	LY
FYLD	9100	FYLD	9100	LZ

[illegible]

+	+	+	+	+	+
+	+	+	+	+	+
+	+	+	+	+	+
+	+	+	+	+	+
+	+	+	+	+	+

- Table 6/10:

+	+	+	+	+	+
Design	NCommandNo	Design	NCommandNo	Design	NCommandNo
Parameter		Parameter		Parameter	
+	+	+	+	+	+
**Indian 800 1984**		**Indian 801 1975**		**Indian 802 1975**	
+	+	+	+	+	+
CODE	9010	CODE	9010	CODE	9010
+	+	+	+	+	+
BEAM	9380	BEAM	9380	CNSF	9380
+	+	+	+	+	+
CMY	9310	CB	9280	DANC	9280
+	+	+	+	+	+
CMZ	9320	CMY	9310	DBL	9310
+	+	+	+	+	+
DFF	9210	CMZ	9320	DMAX	9210
+	+	+	+	+	+
DJ1	9390	CWY	9783	DMIN	9390
+	+	+	+	+	+
DJ2	9400	FLX	9781	ELA	9400
+	+	+	+	+	+
DMAX	9160	FU	9705	FVB	9160
+	+	+	+	+	+
DMIN	9170	FYLD	9100	FYB	9170
+	+	+	+	+	+
FYLD	9100	KX	9235	FYLD	9100
+	+	+	+	+	+
KY	9240	KY	9240	GUSS	9240
+	+	+	+	+	+
KZ	9250	KZ	9250	KY	9250
+	+	+	+	+	+
LY	9130	LX	9125	KZ	9130
+	+	+	+	+	+
LZ	9140	LY	9130	LEG	9140
+	+	+	+	+	+
MAIN	9334	LZ	9140	LY	9334
+	+	+	+	+	+
NSF	9260	NSF	9260	LZ	9260
+	+	+	+	+	+
PROFILE	9520	RATIO	9360	MAIN	9520
+	+	+	+	+	+
RATIO	9360	TRACK	9350	NSF	9360
+	+	+	+	+	+
SSY	9290	TSA	9782	NHL	9290

SSZ	9300	UNL	9270	TRAC
TMAIN	9335			
TRACK	9350			
UNF	9270			
UNL	9150			

- Table 7/10:

Design Parameter	NCommandNo	Design Parameter	NCommandNo	Desi Para
**Japanese AIJ 2002**		**Mexian**		**Ne
CODE	9010	CODE	9010	CODE
BEAM	9380	BEAM	9380	ALM
CAN	9883	CB	9280	BEAM
CB	9280	CMB	9878	DFP
DFP	9210	DFP	9210	DJ1
DJ1	9390	DJ1	9390	DJ2
DJ2	9400	DJ2	9400	DMAX
DMAX	9160	DMAX	9160	DMIN
DMIN	9170	DMIN	9170	DUCT
FYLD	9100	DSD	9879	FU
KY	9240	FU	9705	FYLD
KZ	9250	FYLD	9100	GLD
LY	9130	IMM	9877	IST
LZ	9140	INO	9876	KT
MAIN	9336	IRR	9875	KY
MBG	9823	KX	9235	KZ
MISES	9819	KY	9240	LHT



NSF	9260	KZ	9250	LX
RATIO	9360	LDR	9874	LY
SLF	9922	LX	9125	LZ
TMAIN	9335	LY	9130	MAIN
TRACK	9358	LZ	9140	NSC
UNF	9270	NSF	9260	NSF
UNL	9150	RATIO	9360	PBRA
YNG	9824	STIFF	9200	PBCF
		TRACK	9350	RATI
		UNB	9660	SGR
		UNT	9650	SKL
				SKR
				SKT
				TMA
				TRAC
				TSP
				UNB
				UNT

- Table 8/10:

Design Parameter	NCommandNo	Design Parameter	NCommandNo	Desi Para
**Norwegian NORSOK**		**Russian SNip 81**		**Sc
**N-004**				
CODE	9010	CODE	9010	CODE
BEAM	9380	BEAM	9380	BEAM
CMY	9310	CB	9281	CB
CMZ	9320	CMM	9494	CMY
DFF	9210	CMN	9494	CMZ

DJ1	9390	DFF	9210	DFF
DJ2	9400	DMAX	9160	DJ1
DMAX	9160	DMIN	9170	DJ2
DMIN	9170	GAMC1	9809	DMAX
FYLD	9100	GAMC2	9810	DMIN
FU	9705	GMF	9977	FU
HYD	9940	KY	9240	FYLD
KY	9240	KZ	9250	KT
KZ	9250	LEG	9502	KY
LX	9125	LY	9130	KZ
LY	9130	LZ	9140	LT
LZ	9140	MAIN	9333	LY
MAIN	9330	NSF	9260	LZ
PSD	9941	PY	9121	MAIN
RATIO	9360	RATIO	9360	NSF
SGR	9466	SBLT	9471	RATIO
TMAIN	9335	SGR	9463	SSY
TRACK	9351	TB	9952	SSZ
		TRACK	9356	TRAC
		UNL	9151	UNB
				UNT

- Table 9/10:

Design Parameter	NCommandNo	Design Parameter	NCommandNo	Desi Para
**EC3**		**Australian**		**BS
CODE	9010	CODE	9010	CODE

ALH	9674	ALB	9804	BEAM
+-----+	+-----+	+-----+	+-----+	+-----+
ALPHA	9926	ALM	9803	CMY
+-----+	+-----+	+-----+	+-----+	+-----+
BEAM	9382	BEAM	9380	CMZ
+-----+	+-----+	+-----+	+-----+	+-----+
BETA	9382	DFF	9210	CWY
+-----+	+-----+	+-----+	+-----+	+-----+
C1	9550	DJ1	9390	FLX
+-----+	+-----+	+-----+	+-----+	+-----+
C2	9560	DJ2	9400	FU
+-----+	+-----+	+-----+	+-----+	+-----+
C3	9565	DMAX	9160	FYLD
+-----+	+-----+	+-----+	+-----+	+-----+
CAN	9883	DMIN	9170	KX
+-----+	+-----+	+-----+	+-----+	+-----+
CMM	9482	FU	9705	KY
+-----+	+-----+	+-----+	+-----+	+-----+
CMN	9492	FYLD	9100	KZ
+-----+	+-----+	+-----+	+-----+	+-----+
CMT	9671	IST	9800	LX
+-----+	+-----+	+-----+	+-----+	+-----+
DFF	9210	KT	9805	LY
+-----+	+-----+	+-----+	+-----+	+-----+
DJ1	9390	KY	9240	LZ
+-----+	+-----+	+-----+	+-----+	+-----+
DJ2	9400	KZ	9250	MAIN
+-----+	+-----+	+-----+	+-----+	+-----+
DMAX	9160	LHT	9938	NSF
+-----+	+-----+	+-----+	+-----+	+-----+
DMIN	9170	LX	9125	DMAX
+-----+	+-----+	+-----+	+-----+	+-----+
EFT	9673	LY	9130	RATI
+-----+	+-----+	+-----+	+-----+	+-----+
ELB	9697	LZ	9140	TRAC
+-----+	+-----+	+-----+	+-----+	+-----+
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			
+-----+	+-----+	+-----+	+-----+	+-----+
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	NComr
+=====+	+=====+	+=====+	+=====+
**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
+-----+	+-----+	+-----+	+-----+
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

**Returns**

-----

None

**Examples**

-----

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.CreateSteelDesignCommand(1, 9010, [9380, 9310, 9320])
"""
```

```
int_safe_list = make_safe_array_long_input(IntValues)
float_safe_list = make_safe_array_double_input(FloatValues)
string_safe_list = make_safe_array_string_input(StringValues)
nAssign_safe_list = make_safe_array_long_input(NAssignList)
self._command.CreateSteelDesignCommand(NDesignCode, NCommandNo, int_safe_list,
```

[\[docs\]](#)

```
def PerformCableAnalysis(self, NoOfIterations:int, PrintOption:int):
    """
```

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

**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

#### Returns

-----

None

#### Examples

-----

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.PerformCableAnalysis(25, 6)
"""
self._command.PerformCableAnalysis(NoOfIterations, PrintOption)
```

[\[docs\]](#)

```
def PerformBucklingAnalysis(self, MaxNoOfIterations:int, PrintOption:int):
    """
```

Creates the commands required to perform BUCKLING ANALYSIS.

#### 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

#### Returns

-----

None

Examples

-----

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.PerformBucklingAnalysis(5, 6)
"""
self._command.PerformBucklingAnalysis(MaxNoOfIterations, PrintOption)
```

[\[docs\]](#)

```
def SetFloorDiaphragmBaseCommand(self, baseElevationValue:float):
    """
    Set a floor diaphragm base command.

    Parameters
    -----
    baseElevationValue : int
        The diaphragm base command string.

    Returns
    -----
    int
        Status code '1' for OK and '0' if failed to add or update

    Examples
    -----
    >>> from openstaadpy import os_analytical
    >>> staad_obj = os_analytical.connect()
    >>> status = staad_obj.Command.SetFloorDiaphragmBaseCommand(100.0)
    >>> print(status)
    """
    return self._command.SetFloorDiaphragmBaseCommand(baseElevationValue)
```

[\[docs\]](#)

```
def DeleteFloorDiaphragmBaseCommand(self):
    """
    Delete the floor diaphragm base command.

    Returns
    -----
    int
        Status code '1' for OK and '0' if failed to delete.

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



```
>>> print(status)
"""
return self._command.DeleteFloorDiaphragmBaseCommand()
```

[\[docs\]](#)

```
def SetCheckSoftStoryCommand(self, DesignCode:int):
    """
    Set a check soft story command.

    Parameters
    -----
    DesignCode : int
        Index number for specifying code

    Returns
    -----
    int
        Status code '1' for OK and '0' if failed to add or update

    Examples
    -----
    >>> from openstaadpy import os_analytical
    >>> staad_obj = os_analytical.connect()
    >>> result = staad_obj.Command.SetCheckSoftStoryCommand(3)
    >>> print(result)
    """
    return self._command.SetCheckSoftStoryCommand(DesignCode)
```

[\[docs\]](#)

```
def DeleteCheckSoftStoryCommand(self):
    """
    Delete the check soft story command.

    Returns
    -----
    int
        Status code '1' for OK and '0' if failed to delete.

    Examples
    -----
    >>> from openstaadpy import os_analytical
    >>> staad_obj = os_analytical.connect()
    >>> result = staad_obj.Command.DeleteCheckSoftStoryCommand()
    >>> print(result)
    """
    return self._command.DeleteCheckSoftStoryCommand()
```

[\[docs\]](#)

```
def SetCheckIrregularitiesCommand(self, DesignCode):
    """
    Set a check irregularities command.

    Parameters
    -----
    DesignCode : int
        Index number for specifying code

    Returns
    -----
    int
        Status code '1' for OK and '0' if failed to add or update.

    Examples
    -----
    >>> from openstaadpy import os_analytical
    >>> staad_obj = os_analytical.connect()
    >>> result = staad_obj.Command.SetCheckIrregularitiesCommand(3)
    >>> print(result)
    """
    return self._command.SetCheckIrregularitiesCommand(DesignCode)
```

[\[docs\]](#)

```
def DeleteCheckIrregularitiesCommand(self):
    """
    Delete the check irregularities command.

    Returns
    -----
    int
        Status code '1' for OK and '0' if failed to delete.

    Examples
    -----
    >>> from openstaadpy import os_analytical
    >>> staad_obj = os_analytical.connect()
    >>> result = staad_obj.Command.DeleteCheckIrregularitiesCommand()
    >>> print(result)
    """
    return self._command.DeleteCheckIrregularitiesCommand()
```

[\[docs\]](#)

```
def PerformBucklingAnalysisEx(self, Method:int, MaxNoOfIterations:int, Print
    """
    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

-----

bool

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

## Examples

-----

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> result = staad_obj.Command.PerformBucklingAnalysisEx(0, 15, 6)
"""
return self._command.PerformBucklingAnalysisEx(Method, MaxNoOfIterations,
```

[\[docs\]](#)

```
def PerformCableAnalysisEx(self, AdvancedCableAnalysis:int, AdvOptions:list,
```

```
"""
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` =

AdvOptions : list of int
    Additional options for Advanced Cable Analysis.
```

Array Index	Default	STAAD Command	Description
0	1	REFORM f11	Use Full Newton-Raphson
1	0	KGEOM f12	Use Geometric Matrix (Kg

- 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 repre
```

For **\*\*Basic Cable Analysis\*\***:

Array Index	Default Value	STAAD Command	Description
0	145	STEPS f1	The number of steps Each step will
1	300	EQITERATIONS f2	Maximum number of iterations Should be in
2	1.0E-4	EQTOLERANCE f3	The convergence tolerance
3	0.0	SAGMINIMUM f4	Sag Minimum
4	1.0	STABILITY f5	A stiffness reduction factor cables and r
5	1	f6	The number of
6	0.0	KSMALL f7	A stiffness reduction factor cables and r

For **\*\*Advanced Cable Analysis\*\***:

Array Index	Default Value	STAAD Command	Description
-------------	---------------	---------------	-------------

0	1	STEPS f1	Number of load
1	300	EQITERATIONS f2	Max iterations
2	1.0E-6	EQTOLERANCE f3	Convergence

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  
-----  
bool  
`True` if the command was successfully added or updated, `False` otherwise

Examples  
-----  
>>> from openstaadpy import os\_analytical  
>>> staad\_obj = os\_analytical.connect()  
>>> result = staad\_obj.Command.PerformCableAnalysisEx(1, [0, 1], [125, 300, 1.0E-6])  
>>> print(result)  
"""

```
adv_options_safe_list = make_safe_array_long_input(AdvOptions)
adv_options_safe_list_vt = make_variant_vt_ref(adv_options_safe_list, automation)
params_safe_list = make_safe_array_double_input(Params)
params_safe_list_vt = make_variant_vt_ref(params_safe_list, automation)
return self._command.PerformCableAnalysisEx(AdvancedCableAnalysis,adv_options_safe_list_vt,params_safe_list_vt)
```

[\[docs\]](#)

```
def PerformDirectAnalysis(self, Option:int, Params:list, AddOptions:list, PrintOption:int)
    """
    Assigns the commands required to perform a DIRECT ANALYSIS for AISC on the selected members.

    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

Array Index	Default Value	STAAD Command	Description
0	0.01	TAUTOL f1	Tau-b tolerance
1	0.01	DISPTOL f2	Displacement tolerance - 0.01 in (displacement) - 0.01 radians (rotation) The value should be positive
2	1	ITERDIRECT i3	Max number of iterations for Direct Analysis
3	15	PDiter i5	Iterations for Plastic Design

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 * EI)
1	0	TBITER	Iterate Tau-b? 0 = False

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

-----

bool

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

### Examples

-----

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

```
add_options_safe_list = make_safe_array_long_input(AddOptions)
add_options_safe_list_vt = make_variant_vt_ref(add_options_safe_list,
params_safe_list = make_safe_array_double_input(Params)
params_safe_list_vt = make_variant_vt_ref(params_safe_list, automation
return self._command.PerformDirectAnalysis(Option ,params_safe_list_vt
```

[\[docs\]](#)

```
def PerformNonlinearAnalysisEx(self, PrintOption:int, Arclength:float, NoOf
```

Assigns the commands required to perform NONLINEAR ANALYSIS of the current

### Parameters

-----

PrintOption : int

    Option index for specifying the print output. Choose from the follow

```
+-----+-----+
| 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 state.

Tolerance : float
    Tolerance value for determining convergence.

Steps : int
    Number of load steps. The load is applied incrementally in these steps.

Rebuild : int
    Frequency of rebuilding the Tangent or Stiffness Matrix (K) per load step.

    - 0 : Once per load step
    - 1 : Every load step and iteration

AddGeometricStiffness : int
    Whether to add the geometric stiffness matrix to the stiffness matrix.
    Set to `1` to include, or `0` to exclude.

varDisplLimitData : list of float
    Displacement limit data to specify the target displacement. The array

```

Array Index	Default	STAAD Command	Description
0	1	JOINT_TARGET i1	Joint being monitored
1	1	i2	Global degree of freedom 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

## Returns

.....

Int:

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

## Examples

.....

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> result = staad_obj.Command.PerformNonlinearAnalysisEx(6, 0, 5, 0.0001)
>>> print(result)
"""
```

```
disp_limit_data_safe_list = make_safe_array_double_input(DispLimitData)
return self.command.PerformNonlinearAnalysisEx(PrintOption, Arclength,
```



[\[docs\]](#)

```
def DeleteAllAnalysisCommands(self):
    """
    Delete all analysis commands from the current STAAD model.

    Returns
    -----
    int
        Status code '1' for OK and '0' if failed to delete.

    Examples
    -----
    >>> from openstaadpy import os_analytical
    >>> staad_obj = os_analytical.connect()
    >>> result = staad_obj.Command.DeleteAllAnalysisCommands()
    >>> print(result)
    """
    return self._command.DeleteAllAnalysisCommands()
```

[\[docs\]](#)

```
def PerformPDeltaAnalysisEx(self, NoOfIterations:int, PrintOption:int, bSmall)
    """
    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 follow
```

```
+-----+-----+
| 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.

Returns

-----

None

Examples

-----

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Command.PerformPDeltaAnalysisEx(20, 5, 0, 1)
"""
return self._command.PerformPDeltaAnalysisEx(NoOfIterations ,PrintOptions)
```