Support

Contents

class openstaadpy.os_analytical.ossupport.**OSSupport** [source]

Bases: object

AssignSupportToEntityList(*supportid*, *entitylist*) [source]

Assign the specified support to an entity list.

Parameters:

- **supportid** (*int*) – Support reference ID.
- **entitylist** (*list of int*) – List of node or plate numbers.

Returns:

True if successful.

Return type:

bool

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> status = staad_obj.Support.AssignSupportToEntityList(3, [41, 42])
>>> print(status)
```

AssignSupportToNode(*NodeIDs: list | int*, *SupportID: int*) [source]

Assign a support to one or more nodes.

Parameters:

- **NodeIDs** (*list of int or int*) – List of node numbers or a single node number to assign the support to.
- **SupportID** (*int*) – Support reference number ID.

Return type:

None

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> fixed_id = staad_obj.Support.CreateSupportFixed()
>>> staad_obj.Support.AssignSupportToNode([1, 2, 3], fixed_id) // Passi
>>> staad_obj.Support.AssignSupportToNode(5, fixed_id) //Passing suppor
```

CreateElasticFooting(*length, width, direction, subgrade*) [source]

Create an elastic footing support.

Parameters:

- **length** (*float*) – Length of footing.
- **width** (*float*) – Width of footing.
- **direction** (*int*) – Direction.
- **subgrade** (*float*) – Subgrade modulus.

Returns:

Support reference number ID.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> support_id = staad_obj.Support.CreateElasticFooting(5, 6, 2, 20)
>>> print(support_id)
```

CreateElasticMat(*direction, subgrade, printFlag, springType*)

Create an elastic mat support. [source]

Parameters:

- **direction** (*int*) – Direction.
- **subgrade** (*float*) – Subgrade modulus.
- **printFlag** (*int*) – Print flag.
- **springType** (*int*) – Spring type.

Returns:

Support reference number ID.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> support_id = staad_obj.Support.CreateElasticMat(5, 20, 0, 1)
>>> print(support_id)
```

CreateInclinedSupport(*inclinedType: int, refType: int, refNode: int, coord, releaseSpec: list, springSpec: list*)

[source]

Create an inclined support.

Parameters:

- **inclinedType** (*int*) –

  Type of the inclined support:

Value	Inclined Type
1	Pinned
2	Fixed
3	FixedBut
4	Enforced
5	EnforcedBut

- **refType** (*int*) –

  Type of the reference point:

Value	Table Type
0	fRefX, fRefY, fRefY global distances from the joint to the reference point.
1	fRefX, fRefY, fRefY global coordinates of the reference point.
2	a joint number (vaRefNode) whose x, y, z global coordinates is the reference point.

- **refNode** (*int*) – Reference node number.
- **coord** (*list of float*) – Reference coordinates. [X, Y, Z]
- **releaseSpec** (*list of float*) – Release specification. Fixed (= 0) or Release (= 1) [FX, FY, FZ, MX, MY, MZ]
- **springSpec** (*list of float*) – Spring specification. [KFX, KFY, KFZ, KMX, KMY, KMZ]

Returns:

Support reference number ID.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> support_id = staad_obj.Support.CreateInclinedSupport(2, 2, 1, [0,0,
>>> print(support_id)
```

CreatePlateMat(*direction: int, subgrades, printFlag: bool, springType: int*) [source]

Create a plate mat support.

Parameters:

- **direction** (*int*) –

Value	Inclined Type
0	X Direction
1	Y Direction
2	Z Direction
3	X Only Direction
4	Y Only Direction
5	Z Only Direction
6	All Direction

- **subgrades** (*list of float or float*) – Subgrade modulus value(s).
- **printFlag** (*bool*) – Print flag.
- **springType** (*int*) – Spring type.

Returns:

Support reference number ID. / 0 if error.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> support_id = staad_obj.Support.CreatePlateMat(2, [20, 30, 40], Fals
>>> print(support_id)
```

CreateSupportFixed()

[source]

Creates a fully fixed support.

Returns:

Support reference number ID. -1 indicates General Error.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> count = staad_obj.Support.CreateSupportFixed()
```

CreateSupportFixedBut(*ReleaseSpec: list, SpringSpec: list*)

[source]

Creates fixed support with releases in specified directions or a spring support with spring constants in specified directions.

Returns:

Support reference number ID. -1 indicates General Error.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> count = staad_obj.Support.CreateSupportFixedBut(ReleaseSpec, Spring
```

CreateSupportPinned() [source]

Creates a pinned support (i.e., free to rotate about local y and z axis, fixed in all other degrees of freedom).

Returns:

Support reference number ID. -1 indicates General Error.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> count = staad_obj.Support.CreateSupportPinned()
```

DeleteSupport(*supportNo: int*) [source]

Remove a support item from the model.

Parameters:

supportNo (*int*) – Support item number.

Returns:

True if successful.

Return type:

bool

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> status = staad_obj.Support.DeleteSupport(2)
>>> print(status)
```

GetCountOfElasticFooting() [source]

Get the total number of elastic footing supports.

Returns:

Number of elastic footing supports.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> count = staad_obj.Support.GetCountOfElasticFooting()
>>> print(count)
```

GetCountOfElasticMat() [source]

Get the total number of ElasticMat supports.

Returns:

Number of ElasticMat supports.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> support_count = staad_obj.Support.GetCountOfElasticMat()
>>> print(support_count)
```

GetCountOfPlateMat() [source]

Get the total number of plate mat supports.

Returns:

Number of plate mat supports.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> support_count = staad_obj.Support.GetCountOfPlateMat()
>>> print(support_count)
```

GetElasticFootingAssignmentList(*supportid*) [source]

Get list of assigned node Ids for a specific elastic footing support Id.

Parameters:

supportid (*int*) – Support reference ID.

Returns:

List of node numbers.

Return type:

list of int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> node_list = staad_obj.Support.GetElasticFootingAssignmentList(3)
>>> print(node_list)
```

GetElasticFootingDetail(*supportid*) [source]

Get elastic footing support information for a specific support Id.

Parameters:

supportid (*int*) – Support reference ID.

Returns:

(length, width, direction, subgrade, nodesCount)

Return type:

tuple

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> details = staad_obj.Support.GetElasticFootingDetail(2)
>>> print(details)
```

GetElasticMatAssignmentList(*supportid*) [source]

Get elastic mat support entity list for a specific support Id.

Parameters:

supportid (*int*) – Support reference ID.

Returns:

List of node numbers.

Return type:

list of int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> node_list = staad_obj.Support.GetElasticMatAssignmentList(4)
>>> print(node_list)
```

GetElasticMatDetail(*supportid*) [source]

Get elastic mat support information for a specific support Id.

Parameters:

supportid (*int*) – Support reference ID.

Returns:

(direction, subgrade, printFlag, springType, nodesCount)

- **direction : *int***

Value	Inclined Type
0	X Direction
1	Y Direction
2	Z Direction
3	X Only Direction
4	Y Only Direction
5	Z Only Direction

- **subgrade : *float***

  Subgrade modulus.

- **printFlag : *bool***

  Print flag. True if checked, False if not.

- **springType : *int***

Value	Spring Type
0	None
1	Compression only
2	Multi-linear

- **nodesCount : *int***

  Number of nodes assigned to this support.

Return type:

tuple

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> details = staad_obj.Support.GetElasticMatDetail(4)
>>> print(details)
```

GetPlateMatAssignmentList(*plateMatNo*) [source]

Get plate mat support entity list for a specific support Id.

Parameters:

plateMatNo (*int*) – Plate mat support ID.

Returns:

List of plate numbers.

Return type:

list of int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> plate_list = staad_obj.Support.GetPlateMatAssignmentList(1)
>>> print(plate_list)
```

GetPlateMatDetail(*plateMatNo*) [source]

Get plate mat support information for a specific support Id.

Parameters:

plateMatNo (*int*) – Plate mat support ID.

Returns:

(direction, subgrade1, subgrade2, subgrade3, printFlag, springType, nAssignedPlateCount)

Return type:

tuple

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> details = staad_obj.Support.GetPlateMatDetail(1)
>>> print(details)
```

GetPlateMatSupportId(*plateMatIndex*) [source]

Get the plate mat support ID.

Parameters:

plateMatIndex (*int*) – PlateMat index (starting from 0).

Returns:

Plate mat support ID.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> support_id = staad_obj.Support.GetPlateMatSupportId(1)
>>> print(support_id)
```

GetSupportCount() [source]

Get the total number of supported nodes in the current structure.

Returns:

Number of supported nodes.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> count = staad_obj.Support.GetSupportCount()
```

GetSupportInformation(*nodeNo: int*) [source]

Get support information for the specified node.

Parameters:

nodeNo (*int*) – Node number.

Returns:

Returns a tuple consisting of support_type, list of release specifications.and list of spring specifications respectively.

Return type:

tuple

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> stype, release, spring = staad_obj.Support.GetSupportInformation(1)
```

GetSupportInformationEx(*nodeNo: int*) [source]

Get extended support information for the specified node.

Parameters:

nodeNo (*int*) – Node number.

Returns:

(supportNo, supportType, releaseSpec, springSpec)

- **supportNo**

  Support item number.

- **supportType**

  Support type code.

- **releaseSpec**

  List of release specifications. (= 1) or Fixed (= 0) or Spring (=-1) for [FX, FY, FZ, MX, MY, MZ]

- **springSpec**

  List of spring specifications. [KFX, KFY, KFZ, KMX, KMY, KMZ]

Return type:

tuple

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Support.GetSupportInformationEx(1)
```

GetSupportName(*supportNo: int*) [source]

Get support string name.

Parameters:

supportNo (*int*) – Support item number.

Returns:

Support name.

Return type:

str

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Support.GetSupportName(2)
```

GetSupportNodes() [source]

Get all supported node numbers.

Returns:

List of supported node numbers.

Return type:

list of int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> nodes = staad_obj.Support.GetSupportNodes()
```

GetSupportType(*nodeNo: int*) [source]

Get the support type for the specified node.

Parameters:

nodeNo (*int*) – Node number.

Returns:

Support type code.

Return type:

int

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Support.GetSupportType(1)
```

GetSupportUniqueID(*supportNo: int*) [source]

Get unique ID GUID string for a support item.

Parameters:

supportNo (*int*) – Support item number.

Returns:

GUID string.

Return type:

str

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Support.GetSupportUniqueID(2)
```

RemoveElasticFooting(*supportid*) [source]

Remove elastic footing support for a specific support Id.

Parameters:

supportid (*int*) – Support reference ID.

Returns:

True if successful.

Return type:

bool

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> status = staad_obj.Support.RemoveElasticFooting(3)
>>> print(status)
```

RemoveElasticFootingFromNode(*nodeid*) [source]

Remove elastic footing support from a specific node.

Parameters:

nodeid (*int*) – Node number.

Returns:

True if successful.

Return type:

bool

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> status = staad_obj.Support.RemoveElasticFootingFromNode(2)
>>> print(status)
```

RemoveElasticMat(*supportid*) [source]

Remove elastic mat support for a specific support Id.

Parameters:

supportid (*int*) – Support reference ID.

Returns:

True if successful.

Return type:

bool

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> status = staad_obj.Support.RemoveElasticMat(4)
>>> print(status)
```

RemoveElasticMatFromNode(*nodeid*) [source]

Remove elastic mat support from a specific node.

Parameters:

nodeid (*int*) – Node number.

Returns:

True if successful.

Return type:

bool

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> status = staad_obj.Support.RemoveElasticMatFromNode(7)
>>> print(status)
```

RemovePlateMat(*supportId*) [source]

Remove plate mat support for a specific support Id.

Parameters:

supportId (*int*) – Plate mat support ID.

Returns:

True if successful.

Return type:

bool

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> status = staad_obj.Support.RemovePlateMat(4)
>>> print(status)
```

RemovePlateMatFromPlate(*plateNo: int*) [source]

Remove plate mat support from a specific plate.

Parameters:

plateNo (*int*) – Plate number.

Returns:

True if successful.

Return type:

bool

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> status = staad_obj.Support.RemovePlateMatFromPlate(56)
>>> print(status)
```

RemoveSupportFromNode(*NodeIDs: list*) [source]

Remove support from one or more nodes.

Parameters:

NodeIDs (*list of int*) – List of node numbers from which to remove the support.

Return type:

None

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Support.RemoveSupportFromNode([1, 2, 3])
```

SetSupportUniqueID(*supportNo: int, guid: str*) [source]

Set unique ID for a support item.

Parameters:

- **supportNo** (*int*) – Support item number.
- **guid** (*str*) – GUID string.

Return type:

None

Examples

```
>>> from openstaadpy import os_analytical
>>> staad_obj = os_analytical.connect()
>>> staad_obj.Support.SetSupportUniqueID(2, "supportId")
```

__init__(*staadObj*) [source]