Specification: Get and Remove Specification

Property

Functions

afx_msg VARIANT	**OSPropertyUI::DeleteMemberReleaseSpec** (const VARIANT FAR &varnBeamNo, const VARIANT FAR &varLocation) Delete MEMBER RELEASE specification.
afx_msg long	**OSPropertyUI::DeleteAllControlDependentRelations** () Delete all control/dependent joint specifications from model.
afx_msg VARIANT	**OSPropertyUI::GetElementOffsetSpecCount** () Returns the total number of element offset specifications in the current model.
afx_msg VARIANT	**OSPropertyUI::RemoveAllElementOffsetSpec** () Remove all element node offset specification from the model.
afx_msg VARIANT	**OSPropertyUI::GetMemberSpecCode** (const VARIANT FAR &varMembNo, VARIANT FAR &SpecCode) Get the type of specification attached to specified member.
afx_msg VARIANT	**OSPropertyUI::RemovePropertyFromBeam** (const VARIANT FAR &nBeamNo) Remove property from beam.
BOOL	**OSPropertyUI::RemoveBeamPropertyHelper** (long &beamNo) Remove beam property.
afx_msg VARIANT	**OSPropertyUI::DeleteProperty** (const VARIANT FAR &nProperty) Delete property based the on the property ID passed.
afx_msg VARIANT	**OSPropertyUI::GetPropertyUniqueID** (const VARIANT FAR &nPropNo) Get Property Unique ID.
afx_msg void	**OSPropertyUI::SetPropertyUniqueID** (const VARIANT FAR &nPropNo, const VARIANT FAR &szName) Set Property Unique ID to specification property number.
afx_msg VARIANT	**OSPropertyUI::DeleteMemberSpec** (const VARIANT FAR &varnSpecNo) Delete specification based on the specification number passed.
afx_msg VARIANT	**OSPropertyUI::RemoveMemberReleaseSpecFromBeam** (const VARIANT FAR &varnBeamNo, const VARIANT FAR &varLocation) Removes the member specification from a particular member at the provided location (Start or End)..
afx_msg VARIANT	**OSPropertyUI::RemoveMemberOffsetSpecFromBeam** (const VARIANT FAR &varnBeamNo, const VARIANT FAR &varLocation) Removes the member offset specification from a particular member at the provided location

afx_msg VARIANT	**OSPropertyUI::RemoveMemberTrussSpecFromBeam** (const VARIANT FAR &varnBeamNo) Remove member truss specification from beam.
afx_msg VARIANT	**OSPropertyUI::RemoveMemberInactiveSpecFromBeam** (const VARIANT FAR &varnBeamNo) Remove member inactive specification from beam.
afx_msg VARIANT	**OSPropertyUI::RemoveMemberTensionSpecFromBeam** (const VARIANT FAR &varnBeamNo) Remove member tension specification from beam.
afx_msg VARIANT	**OSPropertyUI::RemoveMemberCompressionSpecFromBeam** (const VARIANT FAR &varnBeamNo) Remove member compression specification from beam.
afx_msg VARIANT	**OSPropertyUI::RemoveMemberIgnoreStiffSpecFromBeam** (const VARIANT FAR &varnBeamNo) Remove member ignore stiff specification from beam.
afx_msg VARIANT	**OSPropertyUI::RemoveMemberCableSpecFromBeam** (const VARIANT FAR &varnBeamNo, const VARIANT FAR &varnCableTension) Removes the member cable specification from a particular member at the provided location type (Tension or Length).
afx_msg VARIANT	**OSPropertyUI::RemoveElementPlaneStressSpecFromPlate** (const VARIANT FAR &varnPlateNo) Remove element plane stress specification from plate.
afx_msg VARIANT	**OSPropertyUI::RemoveElementIgnoreInplaneRotnSpecFromPlate** (const VARIANT FAR &varnPlateNo) Remove element ignore in plane rotation specification from plate.
afx_msg VARIANT	**OSPropertyUI::RemoveElementNodeReleaseSpecFromPlate** (const VARIANT FAR &varnPlateNo, const VARIANT FAR &varNodeNo) Remove element node release specification from plate.
afx_msg VARIANT	**OSPropertyUI::RemoveAllElementNodeReleaseSpec** () Remove all element node release specification from the model.
afx_msg VARIANT	**OSPropertyUI::GetMemberReleaseSpecEx** (const VARIANT FAR &varnBeamNo, const VARIANT FAR &varnEnd, VARIANT FAR &varnReleaseArray, VARIANT FAR &varfSpringConstArray, VARIANT FAR &varfMPFactor, VARIANT FAR &varfMPFactorArray) Get releases for the specified member at the specified end.
afx_msg long	**OSPropertyUI::GetInactiveMemberCount** () Returns the total number of inactive members in the current model.
afx_msg void	**OSPropertyUI::GetInactiveMemberList** (VARIANT FAR &nInactiveMemList) Populates a list of the member ID(s) of all the inactive members in the current model.
	:GetAlphaAngleForSection (const VARIANT FAR &nPropNo, VARIANT FAR &dAlpha)

	Returns the alpha angle of the section in radian.
afx_msg VARIANT	**OSPropertyUI::GetCentroidLocationForSection** (const VARIANT FAR &nPropNo, VARIANT FAR &dCey, VARIANT FAR &dCez) Returns the location of the Centroid of the section.
afx_msg VARIANT	**OSPropertyUI::GetFireProofedBeamCount** () Returns count of beams which are fire proofed.
afx_msg VARIANT	**OSPropertyUI::GetFireProofedBeamList** (VARIANT FAR &nFireProofedBeamList) Populates a list of the member ID(s) of all the fire proofed members in the current model.
afx_msg VARIANT	**OSPropertyUI::GetFireProofDataForBeam** (const VARIANT FAR &varnBeamNo, VARIANT FAR &varnFireProofType, VARIANT FAR &varfThickness, VARIANT FAR &varfDensity) Get fire proofing data for beam.
afx_msg VARIANT	**OSPropertyUI::GetFireProofingSpecCount** () Returns the count of different fire proofing specification in the model.
afx_msg VARIANT	**OSPropertyUI::GetFireProofingSpecDetails** (const VARIANT FAR &varnIndex, VARIANT FAR &varnFireProofType, VARIANT FAR &varfThickness, VARIANT FAR &varfDensity, VARIANT FAR &varnAssignCount) Get the details for the specified fire proofing specification number.
afx_msg VARIANT	**OSPropertyUI::GetFireProofingSpecAssignedBeamCount** (const VARIANT FAR &varnIndex) Get the count of beams assigned with a particular fire proofing specification.
afx_msg VARIANT	**OSPropertyUI::GetFireProofingSpecAssignedBeamList** (const VARIANT FAR &varnIndex, VARIANT FAR &nFireProofedBeamList) Populates a list of the member ID(s) of all the members assigned to a particular fire proofing specification.
afx_msg VARIANT	**OSPropertyUI::RemoveMemberFireProofingSpecFromBeam** (const VARIANT FAR &varnBeamNo) Remove member fire proofing specification from beam.

Detailed Description

These functions are related to get or remove specification.

Function Documentation

◆ DeleteAllControlDependentRelations()

long OSPropertyUI::DeleteAllControlDependentRelations ( )

Delete all control/dependent joint specifications from model.

Return values

0 OK Successfully deleted.

-1 ERROR Delete unsuccessful..

C++ Syntax

```
// Delete all control/dependent joint specifications from model.
long RetVal = OSPropertyUI::DeleteAllControlDependentRelations();
```

VBA Syntax

```
' Delete all control/dependent joint specifications from model.
Dim RetVal As Long
RetVal = OSPropertyUI.DeleteAllControlDependentRelations();
```

DeleteMemberReleaseSpec()

VARIANT OSPropertyUI::DeleteMemberReleaseSpec (const VARIANT FAR & varnBeamNo,
const VARIANT FAR & varLocation)

Delete MEMBER RELEASE specification.

Parameters

[in] **varnBeamNo** The beam number ID(Type: Long/Integer).

[in] **varLocation** The Release location at START (= 0) or END (= 1) of the member.

Return values

FALSE Delete Member Release specification failed

TRUE Delete Member Release specification Successful.

VBA Syntax

```
Option Explicit
    Sub Main
        Dim objOpenStaad As Object
        Dim stdFile As String
        Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
        objOpenStaad.GetSTAADFile stdFile, "TRUE"
        If stdFile="" Then
            MsgBox"Bad"
            Set objOpenStaad = Nothing
            Exit Sub
        End If
        Dim beamNo As Integer
        Dim location As Integer
        beamNo = 70
        location = 0
        Dim RetVal As Boolean
        RetVal = objOpenStaad.Property.DeleteMemberReleaseSpec(beamNo,location)
        Set objOpenStaad = Nothing
    End Sub
```

See also

OSPropertyUI::CreateMemberReleaseSpec

◆ DeleteMemberSpec()

VARIANT OSPropertyUI::DeleteMemberSpec (const VARIANT FAR & varnSpecNo)

Delete specification based on the specification number passed.

Parameters

[in] **varnSpecNo** The specification number.

Return values

FALSE Delete specification Failed.

TRUE Delete specification Successful.

VBA Syntax

```
Option Explicit
    Sub Main
        Dim objOpenStaad As Object
        Dim stdFile As String
        Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
        objOpenStaad.GetSTAADFile stdFile, "TRUE"
        If stdFile="" Then
            MsgBox"Bad"
            Set objOpenStaad = Nothing
            Exit Sub
        End If
        Dim specNo As Long
        specNo = objOpenStaad.Property.CreateMemberTrussSpec
        Dim res As Boolean
        res =  objOpenStaad.Property.DeleteMemberSpec (specNo)
        Set objOpenStaad = Nothing
    End Sub
```

◆ DeleteProperty()

VARIANT OSPropertyUI::DeleteProperty (const VARIANT FAR & nProperty)

Delete property based the on the property ID passed.

Parameters

[in] **nProperty** Property ID(Type:Long).

Return values

FALSE Delete Property Generate Error.

TRUE Delete Property Successful.

VBA Syntax

```
Option Explicit
    Sub Main
        Dim objOpenStaad As Object
        Dim stdFile As String
        Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
        objOpenStaad.GetSTAADFile stdFile, "TRUE"
        If stdFile="" Then
            MsgBox"Bad"
            Set objOpenStaad = Nothing
            Exit Sub
        End If
        Dim nProperty As Long
        nProperty = 2
        Dim res As Boolean
        res =  objOpenStaad.Property.DeleteProperty (nProperty)
        Set objOpenStaad = Nothing
    End Sub
```

◆ GetAlphaAngleForSection()

VARIANT OSPropertyUI::GetAlphaAngleForSection (const VARIANT FAR & nPropNo,
VARIANT FAR & dAlpha)

Returns the alpha angle of the section in radian.

@ALPHA = Gets the angle between the principal axis and geometric axis of the section

Parameters

[in] **nPropNo** The specified property ID.

[out] **dAlpha** alpha angle returned (in Radian).

C++ Syntax

```
// dAlpha
// Get Alpha angle for Section Property 4
OSPropertyUI::GetAlphaAngleForSection(4, &dAlpha);
```

VBA Syntax

```
// dAlpha
// Get Alpha angle for Section Property 4
OSPropertyUI.GetAlphaAngleForSection(4, &dAlpha)
```

See also

OSPropertyUI::GetAlphaAngleForSection

◆ GetCentroidLocationForSection()

VARIANT OSPropertyUI::GetCentroidLocationForSection (const VARIANT FAR & nPropNo,
VARIANT FAR & dCey,
VARIANT FAR & dCez)

Returns the location of the Centroid of the section.

Gets the location of the Centroid of the specified section.

@The Cez and Cey are distances to centroid from top left outer edge to the centroid in terms of Y axis and Z axis respectively.

Parameters

[in] **nPropNo** The specified property ID.

[out] **dCey** returns offset value of Centroid along Y axis.

[out] **dCez** returns offset value of Centroid along Z axis.

C++ Syntax

```
// dCey, dCez
// Get Cey and Cez values for Section Property 4
OSPropertyUI::GetCentroidLocationForSection(4, &dCey, &dCez);
```

VBA Syntax

```
// dAlpha
// Get Cey and Cez values for Section Property 4
OSPropertyUI.GetCentroidLocationForSection(4, &dCey, &dCez)
```

See also

OSPropertyUI::GetCentroidLocationForSection

◆ GetElementOffsetSpecCount()

VARIANT OSPropertyUI::GetElementOffsetSpecCount ( )

Returns the total number of element offset specifications in the current model.

Return values

<Val> The total number of element offset specifications.

C++ Syntax

```
// Count of element offset specificationst
long nCount = OSPropertyUI::GetElementOffsetSpecCount();
```

VBA Syntax

```
' Count of element offset specifications.
Dim RetVal As long = OSPropertyUI.GetElementOffsetSpecCount();
```

*

See also

OSPropertyUI::GetInactiveMemberList

◆ GetFireProofDataForBeam()

VARIANT OSPropertyUI::GetFireProofDataForBeam (const VARIANT FAR & varnBeamNo,
VARIANT FAR & varnFireProofType,
VARIANT FAR & varfThickness,
VARIANT FAR & varfDensity)

Get fire proofing data for beam.

Parameters

[in]	**varnBeamNo**	The beam number (type - Long/Integer).
[out]	**varnFireProofType**	Type of fire proof [1 for BFP, 2 for CFP] (type - Long/Integer).
[out]	**varfThickness**	Thickness of fire proof (type - Double).
[out]	**varfDensity**	Density of fire proof (type - Double).

Return values

1 if the method is successful.

0 if the method is unsuccessful.

VBA Syntax

```
Option Explicit

   Sub Main
       Dim objOpenStaad As Object
       Dim stdFile As String

       Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
       objOpenStaad.GetSTAADFile stdFile, "TRUE"
       If stdFile="" Then
           MsgBox"Bad"
           Set objOpenStaad = Nothing
           Exit Sub
       End If

       Dim varReturnVal As Long
       Dim varFireProofType As Long
       Dim varThickness As Double
       Dim varDensity As Double

       varReturnVal = objOpenStaad.Property.GetFireProofDataForBeam(10, varFireProofType,
        varThickness, varDensity)

       If varReturnVal = 1 Then
          MsgBox"GetFireProofDataForBeam method is successful."
       Else
          MsgBox"GetFireProofDataForBeam method is unsuccessful."
       End If

       Set objOpenStaad = Nothing
   End Sub
```

GetFireProofedBeamCount()

VARIANT OSPropertyUI::GetFireProofedBeamCount ( )

Returns count of beams which are fire proofed.

Return values

<Val> The total number of fire proofed beams in the current model.

VBA Syntax

```
Option Explicit

   Sub Main
       Dim objOpenStaad As Object
       Dim stdFile As String

       Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
       objOpenStaad.GetSTAADFile stdFile, "TRUE"
       If stdFile="" Then
           MsgBox"Bad"
           Set objOpenStaad = Nothing
           Exit Sub
       End If

       Dim varCountReturnVal As Long
       varCountReturnVal = objOpenStaad.Property.GetFireProofedBeamCount()

       If varCountReturnVal > 0 Then
          MsgBox"Fire proofed beams found."
       Else
          MsgBox"Fire proofed beams not found."
       End If

       Set objOpenStaad = Nothing
   End Sub
```

See also

OSPropertyUI::GetFireProofedBeamList

GetFireProofedBeamList()

VARIANT OSPropertyUI::GetFireProofedBeamList (VARIANT FAR & nFireProofedBeamList)

Populates a list of the member ID(s) of all the fire proofed members in the current model.

Parameters

[out] **nFireProofedBeamList** VARIANT array of LONG type, for storing returned member number ID(s) of all the members that are fire proofed.

Return values

1 if the method is successful.

0 if the method is unsuccessful.

VBA Syntax

```
Option Explicit

   Sub Main
       Dim objOpenStaad As Object
       Dim stdFile As String

       Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
       objOpenStaad.GetSTAADFile stdFile, "TRUE"
       If stdFile="" Then
           MsgBox"Bad"
           Set objOpenStaad = Nothing
           Exit Sub
       End If

       Dim varCountReturnVal As Long
       Dim varBeamReturnVal As Long
       Dim varBeamIds() As Long

       varCountReturnVal = objOpenStaad.Property.GetFireProofedBeamCount()

       If varCountReturnVal > 0 Then
          ReDim varBeamIds(varCountReturnVal-1)

          varBeamReturnVal = objOpenStaad.Property.GetFireProofedBeamList(varBeamIds)
          If varBeamReturnVal = 1 Then
              MsgBox"GetFireProofedBeamList method is successful."
          Else
              MsgBox"GetFireProofedBeamList method is unsuccessful."
          End If
       Else
          MsgBox"Fire proofed beams not found."
       End If

       Set objOpenStaad = Nothing
   End Sub
```

See also

OSPropertyUI::GetFireProofedBeamCount

◆ GetFireProofingSpecAssignedBeamCount()

VARIANT OSPropertyUI::GetFireProofingSpecAssignedBeamCount (const VARIANT FAR & varnIndex)

Get the count of beams assigned with a particular fire proofing specification.

Parameters

[in] **varnIndex** Non-zero based index of the fire proofing specification (type - Long/Integer).

Return values

Returns the number of beams assigned with a particular fire proofing specification.

VBA Syntax

```
Option Explicit

   Sub Main
       Dim objOpenStaad As Object
       Dim stdFile As String

       Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
       objOpenStaad.GetSTAADFile stdFile, "TRUE"
       If stdFile="" Then
           MsgBox"Bad"
           Set objOpenStaad = Nothing
           Exit Sub
       End If

       Dim varCountReturnVal As Long
       varCountReturnVal = objOpenStaad.Property.GetFireProofingSpecCount()

       If varCountReturnVal > 0 Then
          Dim varAssignCount As Long
          varAssignCount = objOpenStaad.Property.GetFireProofingSpecAssignedBeamCount(1)
          If varAssignCount > 0 Then
             MsgBox"Fire proofing assigned beams found."
          Else
             MsgBox"Fire proofing assigned beams not found."
          End If
       Else
          MsgBox"Fire proofing specifications not found."
       End If

       Set objOpenStaad = Nothing
   End Sub
```

See also

OSPropertyUI::GetFireProofingSpecCount()

◆ ssignedBeamList()

VARIANT
OSPropertyUI::GetFireProofingSpecAssignedBeamList (const VARIANT FAR & varnIndex,
VARIANT FAR & nFireProofedBeamList)

Populates a list of the member ID(s) of all the members assigned to a particular fire proofing specification.

Parameters

[in]	**varnIndex**	Non-zero based index of the fire proofing specification (type - Long/Integer).
[out]	**nFireProofedBeamList**	VARIANT array of LONG type, for storing returned member numbers of all the members that are fire proofed with a particular fire proofing specification.

Return values

1 if the method is successful.

0 if the method is unsuccessful.

VBA Syntax

```
Option Explicit

   Sub Main
       Dim objOpenStaad As Object
       Dim stdFile As String

       Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
       objOpenStaad.GetSTAADFile stdFile, "TRUE"
       If stdFile="" Then
           MsgBox"Bad"
           Set objOpenStaad = Nothing
           Exit Sub
       End If

       Dim varCountReturnVal As Long
       Dim varBeamReturnVal As Long
       Dim varBeamIds() As Long

       varCountReturnVal = objOpenStaad.Property.GetFireProofingSpecAssignedBeamCount(1)

       If varCountReturnVal > 0 Then
          ReDim varBeamIds(varCountReturnVal-1)

          varBeamReturnVal = objOpenStaad.Property.GetFireProofingSpecAssignedBeamList(1,
        varBeamIds)
          If varBeamReturnVal = 1 Then
              MsgBox"Method is successful."
          Else
              MsgBox"Method is unsuccessful."
          End If
       Else
          MsgBox"Fire proofed beams not found."
```



```
       Set objOpenStaad = Nothing
```

```
    End Sub
```

See also

OSPropertyUI::GetFireProofingSpecAssignedBeamCount()

◆ GetFireProofingSpecCount()

VARIANT OSPropertyUI::GetFireProofingSpecCount ( )

Returns the count of different fire proofing specification in the model.

Return values

<Val> The total number of fire proofing specification.

VBA Syntax

```
Option Explicit

   Sub Main
       Dim objOpenStaad As Object
       Dim stdFile As String

       Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
       objOpenStaad.GetSTAADFile stdFile, "TRUE"
       If stdFile="" Then
           MsgBox"Bad"
           Set objOpenStaad = Nothing
           Exit Sub
       End If

       Dim varCountReturnVal As Long
       varCountReturnVal = objOpenStaad.Property.GetFireProofingSpecCount()

       If varCountReturnVal > 0 Then
          MsgBox"Fire proofing specifications found."
       Else
          MsgBox"Fire proofing specifications not found."
       End If

       Set objOpenStaad = Nothing
   End Sub
```

See also

OSPropertyUI::GetFireProofingSpecDetails

◆ GetFireProofingSpecDetails()

```
VARIANT OSPropertyUI::GetFireProofingSpecDetails ( const VARIANT FAR & varnIndex,
                                                   VARIANT FAR &       varnFireProofType,
                                                   VARIANT FAR &       varfThickness,
                                                   VARIANT FAR &       varfDensity,
                                                   VARIANT FAR &       varnAssignCount )
```

Get the details for the specified fire proofing specification number.

Parameters

[in]	**varnIndex**	Non-zero based index of the fire proofing specification (type - Long/Integer).
[out]	**varnFireProofType**	Type of fire proof [1 for BFP, 2 for CFP] (type - Long/Interger).
[out]	**varfThickness**	Thickness of fire proof (type - Double).
[out]	**varfDensity**	Density of fire proof (type - Double).
[out]	**varnAssignCount**	Number of beams assigned (type - array of Long)

Return values

1 if the method is successful.

0 if the method is unsuccessful.

VBA Syntax

```
Option Explicit

   Sub Main
       Dim objOpenStaad As Object
       Dim stdFile As String

       Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
       objOpenStaad.GetSTAADFile stdFile, "TRUE"
       If stdFile="" Then
           MsgBox"Bad"
           Set objOpenStaad = Nothing
           Exit Sub
       End If

       Dim varCountReturnVal As Long
       varCountReturnVal = objOpenStaad.Property.GetFireProofingSpecCount()

       If varCountReturnVal > 0 Then
          Dim varFireProofType As Long
          Dim varThickness As Double
          Dim varDensity As Double
          Dim varAssignCount As Long
          Dim varReturn As Long
          varReturn = objOpenStaad.Property.GetFireProofingSpecDetails(1,
        varFireProofType, varThickness, varDensity, varAssignCount)
          If varReturn = 1 Then
             MsgBox"Fire proofing details found."
          Else
             MsgBox"Fire proofing details not found."
```

```
        Else
           MsgBox"Fire proofing specifications not found."
        End If

        Set objOpenStaad = Nothing
    End Sub
```

See also

OSPropertyUI::GetFireProofingSpecCount()

GetInactiveMemberCount()

long OSPropertyUI::GetInactiveMemberCount ( )

Returns the total number of inactive members in the current model.

Return values

<Val> The total number of inactive member(s).

C++ Syntax

```
// Get inactive members count
long inactiveMemCount = OSPropertyUI::GetInactiveMemberCount();
```

VBA Syntax

```
' Get inactive members list.
Dim RetVal As long = OSPropertyUI.GetInactiveMemberCount();
```

*

See also

OSPropertyUI::GetInactiveMemberList

GetInactiveMemberList()

void OSPropertyUI::GetInactiveMemberList (VARIANT FAR & nInactiveMemList)

Populates a list of the member ID(s) of all the inactive members in the current model.

Parameters

[out] **nInactiveMemList** VARIANT array of LONG type, for storing returned member number ID(s) of all the members that are inactive.

C++ Syntax

```
// Get inactive member list
OSPropertyUI::GetInactiveMemberList(&nInactiveMemList);
```

VBA Syntax

```
' Get inactive member list.
OSPropertyUI.GetInactiveMemberList(&nInactiveMemList)
```

See also

OSPropertyUI::GetInactiveMemberCount

◆ GetMemberReleaseSpecEx()

```
VARIANT OSPropertyUI::GetMemberReleaseSpecEx ( const VARIANT FAR & varnBeamNo,
                                               const VARIANT FAR & varnEnd,
                                               VARIANT FAR &       varnReleaseArray,
                                               VARIANT FAR &       varfSpringConstArray,
                                               VARIANT FAR &       varfMPFactor,
                                               VARIANT FAR &       varfMPFactorArray )
```

Get releases for the specified member at the specified end.

Parameters

[in]	**varnBeamNo**	The beam number ID.
[in]	**varnEnd**	Member Start end (= 0); member End end (= 1).
[out]	**varnReleaseArray**	Translational release VARIANT array with 6 elements for 6 DOFs. @ Element value: No release or spring = 0, release = 1, spring = -1 , Only MP defined = -3 , MPX, MPY or MPZ defined = -2 .
[out]	**varfSpringConstArray**	Rotational releases VARIANT array with 6 elements for 6 DOFs @ Element values Spring value or partial moment factor in floating point number
[out]	**varfMPFactor**	Partial moment release factor (same for MX, MY and MZ)
[out]	**varfMPFactorArray**	Rotational releases VARIANT array with 3 elements for 3 rotational DOFs @ Element values Spring value or partial moment factor in floating point number

Return values

1 OK.

0 General error.

C++ Syntax

```
// Get the release specification of member #2 of End end.
VARIANT RetVal = OSPropertyUI::GetMemberReleaseSpecEx(2, 1, &varnReleaseArray,
       &varfSpringConstArray, &varfMPFactor, &varfMPFactorArray);
```

VBA Syntax

```
' Get the release specification of member #2 of End end.
Option Explicit

    Sub Main
        Dim objOpenStaad As Object
        Dim stdFile As String
        Dim varnReleaseArray(5) As Long
        Dim varfSpringConstArray(5) As Double
        Dim varfMPFactor As Double
        Dim varfMPFactorArray(2) As Double
        Dim RetVal As Long
```

```
    Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
    objOpenStaad.GetSTAADFile stdFile, "TRUE"
   result = objOpenStaad.Property.GetMemberReleaseSpecEx (2, 0, varnReleaseArray,
   varfSpringConstArray, varfMPFactor, varfMPfactorArray)
End Sub
```

GetMemberSpecCode()

VARIANT OSPropertyUI::GetMemberSpecCode (const VARIANT FAR & varMembNo,
VARIANT FAR & SpecCode)

Get the type of specification attached to specified member.

Parameters

[in] **varMembNo** The member number ID.

[out] **SpecCode** Value referring to type of member specification;

Value	Type of Member Specification
0	Truss Member
1	Tension-only Member
2	Compression-only Member
3	Cable-only Member
4	Joist Member
-1	Other

For additional information, please refer to Section "TR.23 Axial Member Specifications" of the Technical Reference Manual.

Return values

TRUE/1 OK.

FALSE/0 Error.

C++ Syntax

```
// Get the type of specification of member #3
VARIANT RetVal = OSPropertyUI::GetMemberSpecCode(3, &SpecCode);
```

VBA Syntax

```
' Get the type of specification of member #3
Dim RetVal As VARIANT = OSPropertyUI.GetMemberSpecCode(3, &SpecCode);
```

See also

OSPropertyUI::CreateMemberTrussSpec

OSPropertyUI::CreateMemberTensionSpec

OSPropertyUI::CreateMemberCompressionSpec

OSPropertyUI::CreateMemberCableSpec

OSPropertyUI::AssignMemberSpecToBeam

VARIANT OSPropertyUI::GetPropertyUniqueID (const VARIANT FAR & nPropNo)

Get Property Unique ID.

Parameters

[in] **nPropNo** Property number(Type:Long).

Returns

Property Unique ID(Type: String).

VBA Syntax

```
Option Explicit
    Sub Main
        Dim objOpenStaad As Object
        Dim stdFile As String
        Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
        objOpenStaad.GetSTAADFile stdFile, "TRUE"
        If stdFile="" Then
            MsgBox"Bad"
            Set objOpenStaad = Nothing
            Exit Sub
        End If
        Dim nProperty As Long
        nProperty = 2
        Dim uniqueID As String
        uniqueID =  objOpenStaad.Property.GetPropertyUniqueID (nProperty)
        Set objOpenStaad = Nothing
    End Sub
```

See also

OSPropertyUI::SetPropertyUniqueID

◆ RemoveAllElementNodeReleaseSpec()

VARIANT OSPropertyUI::RemoveAllElementNodeReleaseSpec ( )

Remove all element node release specification from the model.

Return values

1 OK.

0 No element release specification present

Example (C++ Syntax)

```
// Remove all element node release specifications.
VARIANT RetVal = OSPropertyUI::RemoveAllElementNodeReleaseSpec();
```

Example (C# Syntax)

```
// Remove all element node release specifications.
VARIANT RetVal = OSPropertyUI::RemoveAllElementNodeReleaseSpec();
```

Example (VBA Syntax)

```
' Remove all element node release specifications.
Dim RetVal as VARIANT
RetVal = OSPropertyUI.RemoveAllElementNodeReleaseSpec();
```

◆ RemoveAllElementOffsetSpec()

VARIANT OSPropertyUI::RemoveAllElementOffsetSpec ( )

Remove all element node offset specification from the model.

Return values

1 OK.

0 No element offset specification present

Example (C++ Syntax)

```
// Remove all element node offset specifications.
VARIANT RetVal = OSPropertyUI::RemoveAllElementOffsetSpec();
```

Example (C# Syntax)

```
// Remove all element node offset specifications.
VARIANT RetVal = OSPropertyUI::RemoveAllElementOffsetSpec();
```

Example (VBA Syntax)

```
' Remove all element node offset specifications.
Dim RetVal as VARIANT
RetVal = OSPropertyUI.RemoveAllElementOffsetSpec();
```

◆ RemoveBeamPropertyHelper()

BOOL OSPropertyUI::RemoveBeamPropertyHelper (long & beamNo) private

Remove beam property.

Parameters

[in] **beamNo** The beam number ID.

Return values

0 OK.

-1 General error.

C++ Syntax

```
// Remove beam property from beam #3.
VARIANT RetVal = OSPropertyUI::RemoveBeamPropertyHelper(3);
```

VBA Syntax

```
' Remove beam property from beam #3.
Dim RetVal As BOOL = OSPropertyUI.RemoveBeamPropertyHelper(3);
```

◆ RemoveElementIgnoreInplaneRotnSpecFromPlate()

VARIANT
OSPropertyUI::RemoveElementIgnoreInplaneRotnSpecFromPlate (const VARIANT FAR & varnPlateNo)

Remove element ignore in plane rotation specification from plate.

Parameters

[in] **varnPlateNo** The plate number ID.

Return values

0 OK.

-1 General error.

C++ Syntax

```
// Remove element ignore in plane rotation specification from plate string name of plate
       #3.
VARIANT RetVal = OSPropertyUI::RemoveElementIgnoreInplaneRotnSpecFromPlate(3);
```

VBA Syntax

```
' Remove element ignore in plane rotation specification from plate string name of plate
       #3.
Dim RetVal As VARIANT = OSPropertyUI.RemoveElementIgnoreInplaneRotnSpecFromPlate(3);
```

◆ RemoveElementNodeReleaseSpecFromPlate()

VARIANT OSPropertyUI::RemoveElementNodeReleaseSpecFromPlate (const VARIANT FAR & varnPlateNo, const VARIANT FAR & varNodeNo)

Remove element node release specification from plate.

Parameters

[in] **varnPlateNo** The plate number ID.

[in] **varNodeNo** The node number ID to be released.

Return values

0 OK.

-1 General error.

C++ Syntax

```
// Remove element node #2 release specification from plate string name of plate #3.
VARIANT RetVal = OSPropertyUI::RemoveElementNodeReleaseSpecFromPlate(3, 2);
```

VBA Syntax

```
' Remove element node #2 release specification from plate string name of plate #3.
Dim RetVal As VARIANT = OSPropertyUI.RemoveElementNodeReleaseSpecFromPlate(3, 2);
```

◆ RemoveElementPlaneStressSpecFromPlate()

VARIANT OSPropertyUI::RemoveElementPlaneStressSpecFromPlate (const VARIANT FAR & varnPlateNo)

Remove element plane stress specification from plate.

Parameters

[in] **varnPlateNo** The plate number ID.

Return values

0 OK.

-1 General error.

C++ Syntax

```
// Remove element plane stress specification from plate string name of plate #3.
VARIANT RetVal = OSPropertyUI::RemoveElementPlaneStressSpecFromPlate(3);
```

VBA Syntax

```
' Remove element plane stress specification from plate string name of plate #3.
Dim RetVal As VARIANT = OSPropertyUI.RemoveElementPlaneStressSpecFromPlate(3);
```

◆ RemoveMemberCableSpecFromBeam()

VARIANT OSPropertyUI::RemoveMemberCableSpecFromBeam (const VARIANT FAR & varnBeamNo, const VARIANT FAR & varLocation)

Removes the member cable specification from a particular member at the provided location type (Tension or Length).

Parameters

[in] **varnBeamNo** The beam number ID(Type: Long/Integer).

[in] **varLocation** The Cable location at Tension (= 0) or Length (= 1) of the member.

Return values

FALSE Remove Member Cable specification failed

TRUE Remove Member Cable specification Successful.

VBA Syntax

```
Option Explicit
    Sub Main
        Dim objOpenStaad As Object
        Dim stdFile As String
        Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
        objOpenStaad.GetSTAADFile stdFile, "TRUE"
        If stdFile="" Then
            MsgBox"Bad"
            Set objOpenStaad = Nothing
            Exit Sub
        End If
        Dim beamNo As Integer
        Dim location As Integer
        beamNo = 70
        location = 0
        Dim RetVal As Boolean
       RetVal = objOpenStaad.Property.RemoveMemberCableSpecFromBeam(beamNo,location)
        Set objOpenStaad = Nothing
    End Sub
```

See also

OSPropertyUI::CreateMemberOffsetSpec

◆ RemoveMemberCompressionSpecFromBeam()

VARIANT OSPropertyUI::RemoveMemberCompressionSpecFromBeam (const VARIANT FAR & varnBeamNo)

Remove member compression specification from beam.

Parameters

[in] **varnBeamNo** The beam number ID.

Return values

0 OK.

-1 General error.

C++ Syntax

```
// Remove member compression specification from beam #3.
VARIANT RetVal = OSPropertyUI::RemoveMemberCompressionSpecFromBeam(3);
```

VBA Syntax

```
' Remove member compression specification from beam #3.
Dim RetVal As VARIANT = OSPropertyUI.RemoveMemberCompressionSpecFromBeam(3);
```

◆ RemoveMemberFireProofingSpecFromBeam()

VARIANT OSPropertyUI::RemoveMemberFireProofingSpecFromBeam (const VARIANT FAR & varnBeamNo)

Remove member fire proofing specification from beam.

Parameters

[in] **varnBeamNo** The beam number ID (type - Long/Integer).

Return values

1 Fire proofing specification removed from beam.

0 Unable to remove fire proofing specification from beam.

VBA Syntax

```
Option Explicit

   Sub Main
       Dim objOpenStaad As Object
       Dim stdFile As String

       Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
       objOpenStaad.GetSTAADFile stdFile, "TRUE"
       If stdFile="" Then
           MsgBox"Bad"
           Set objOpenStaad = Nothing
           Exit Sub
       End If

       Dim varReturnVal As Long
       varReturnVal = objOpenStaad.Property.RemoveMemberFireProofingSpecFromBeam(1)

       If varReturnVal = 1 Then
          MsgBox"Fire proofing specification removed from beam."
       Else
          MsgBox"Unable to remove fire proofing specification from beam."
       End If

       Set objOpenStaad = Nothing
   End Sub
```

◆ RemoveMemberIgnoreStiffSpecFromBeam()

VARIANT OSPropertyUI::RemoveMemberIgnoreStiffSpecFromBeam (const VARIANT FAR & varnBeamNo)

Remove member ignore stiff specification from beam.

Parameters

[in] **varnBeamNo** The beam number ID.

Return values

0 OK.

-1 General error.

C++ Syntax

```
// Remove member ignore stiff specification from beam #3.
VARIANT RetVal = OSPropertyUI::RemoveMemberIgnoreStiffSpecFromBeam(3);
```

VBA Syntax

```
' Remove member ignore stiff specification from beam #3.
Dim RetVal As VARIANT = OSPropertyUI.RemoveMemberIgnoreStiffSpecFromBeam(3);
```

◆ RemoveMemberInactiveSpecFromBeam()

VARIANT OSPropertyUI::RemoveMemberInactiveSpecFromBeam (const VARIANT FAR & varnBeamNo)

Remove member inactive specification from beam.

Parameters

[in] **varnBeamNo** The beam number ID.

Return values

0 OK.

-1 General error.

C++ Syntax

```
// Remove member inactive specification from beam #3.
VARIANT RetVal = OSPropertyUI::RemoveMemberInactiveSpecFromBeam(3);
```

VBA Syntax

```
' Remove member truss specification from beam #3.
OSPropertyUI.RemoveMemberInactiveSpecFromBeam(3);
```

◆ RemoveMemberOffsetSpecFromBeam()

VARIANT OSPropertyUI::RemoveMemberOffsetSpecFromBeam (const VARIANT FAR & varnBeamNo,
const VARIANT FAR & varLocation)

Removes the member offset specification from a particular member at the provided location (Start or End).

Parameters

[in] **varnBeamNo** The beam number ID(Type: Long/Integer).

[in] **varLocation** The offset location at START (= 0) or END (= 1) of the member.

Return values

FALSE Remove Member offset specification failed

TRUE Remove Member offset specification Successful.

VBA Syntax

```
Option Explicit
    Sub Main
        Dim objOpenStaad As Object
        Dim stdFile As String
        Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
        objOpenStaad.GetSTAADFile stdFile, "TRUE"
        If stdFile="" Then
            MsgBox"Bad"
            Set objOpenStaad = Nothing
            Exit Sub
        End If
        Dim beamNo As Integer
        Dim location As Integer
        beamNo = 70
        location = 0
        Dim RetVal As Boolean
       RetVal = objOpenStaad.Property.RemoveMemberOffsetSpecFromBeam(beamNo,location)
        Set objOpenStaad = Nothing
    End Sub
```

See also

OSPropertyUI::CreateMemberOffsetSpec

◆ RemoveMemberReleaseSpecFromBeam()

VARIANT OSPropertyUI::RemoveMemberReleaseSpecFromBeam (const VARIANT FAR & varnBeamNo,
const VARIANT FAR & varLocation)

Removes the member specification from a particular member at the provided location (Start or End)..

Parameters

[in] **varnBeamNo** The beam number ID(Type: Long/Integer).

[in] **varLocation** The Release location at START (= 0) or END (= 1) of the member.

Return values

FALSE Remove Member Release specification failed

TRUE Remove Member Release specification Successful.

VBA Syntax

```
Option Explicit
    Sub Main
        Dim objOpenStaad As Object
        Dim stdFile As String
        Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
        objOpenStaad.GetSTAADFile stdFile, "TRUE"
        If stdFile="" Then
            MsgBox"Bad"
            Set objOpenStaad = Nothing
            Exit Sub
        End If
        Dim beamNo As Integer
        Dim location As Integer
        beamNo = 70
        location = 0
        Dim RetVal As Boolean
       RetVal = objOpenStaad.Property.RemoveMemberReleaseSpecFromBeam(beamNo,location)
        Set objOpenStaad = Nothing
    End Sub
```

See also

OSPropertyUI::CreateMemberReleaseSpec

◆ RemoveMemberTensionSpecFromBeam()

VARIANT OSPropertyUI::RemoveMemberTensionSpecFromBeam (const VARIANT FAR & varnBeamNo)

Remove member tension specification from beam.

Parameters

[in] **varnBeamNo** The beam number ID.

Return values

0 OK.

-1 General error.

C++ Syntax

```
// Remove member tension specification from beam #3.
VARIANT RetVal = OSPropertyUI::RemoveMemberTensionSpecFromBeam(3);
```

VBA Syntax

```
' Remove member tension specification from beam #3.
Dim RetVal As VARIANT = OSPropertyUI.RemoveMemberTensionSpecFromBeam(3);
```

◆ RemoveMemberTrussSpecFromBeam()

VARIANT OSPropertyUI::RemoveMemberTrussSpecFromBeam (const VARIANT FAR & varnBeamNo)

Remove member truss specification from beam.

Parameters

[in] **varnBeamNo** The beam number ID.

Return values

0 OK.

-1 General error.

C++ Syntax

```
// Remove member truss specification from beam #3.
VARIANT RetVal = OSPropertyUI::RemoveMemberTrussSpecFromBeam(3);
```

VBA Syntax

```
' Remove member truss specification from beam #3.
OSPropertyUI.RemoveMemberTrussSpecFromBeam(3);
```

RemovePropertyFromBeam()

VARIANT OSPropertyUI::RemovePropertyFromBeam (const VARIANT FAR & nBeamNo)

Remove property from beam.

Parameters

[in] **nBeamNo** The beam number ID.

Return values

0 OK.

-1 General error.

C++ Syntax

```
// Remove property from beam #3.
VARIANT RetVal = OSPropertyUI::RemovePropertyFromBeam(3);
```

VBA Syntax

```
' Remove property from beam #3.
Dim RetVal As VARIANT = OSPropertyUI.RemovePropertyFromBeam(3);
```

SetPropertyUniqueID()

void OSPropertyUI::SetPropertyUniqueID (const VARIANT FAR & nPropNo,
const VARIANT FAR & szID)

Set Property Unique ID to specification property number.

Parameters

[in] **nPropNo** Property number(Type:Long).

[in] **szID** Property Unique ID(Type:String).

VBA Syntax

```
Option Explicit
    Sub Main
        Dim objOpenStaad As Object
        Dim stdFile As String
        Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
        objOpenStaad.GetSTAADFile stdFile, "TRUE"
        If stdFile="" Then
            MsgBox"Bad"
            Set objOpenStaad = Nothing
            Exit Sub
        End If
        Dim nProperty As Long
        nProperty = 2
        Dim uniqueID As String
        uniqueID =  objOpenStaad.Property.GetPropertyUniqueID (nProperty)
        Dim newUniqueID As String
        newUniqueID = "EA8A58A7-FF56-4F25-A9A9-C6D0797FCC47"
        objOpenStaad.Property.SetPropertyUniqueID (nProperty, newUniqueID)
        uniqueID =  objOpenStaad.Property.GetPropertyUniqueID (nProperty)
        Set objOpenStaad = Nothing
    End Sub
```

See also

OSPropertyUI::GetPropertyUniqueID