Result: Members

Analysis Results

Functions

afx_msg VARIANT	**OSOutputUI::GetMemberEndDisplacements** (const VARIANT FAR &nMemberNo, const VARIANT FAR &nEnd, const VARIANT FAR &nLC, VARIANT FAR &pdDispls) Returns member end displacements for specified member number, member end and load case.
afx_msg VARIANT	**OSOutputUI::GetIntermediateMemberTransDisplacements** (const VARIANT FAR &nMemberNo, const VARIANT FAR &dDistance, const VARIANT FAR &nLC, VARIANT FAR &pdDisps) Returns section displacement (or relative displacements) of a beam section for specified member number, distance, and load case.
afx_msg VARIANT	**OSOutputUI::GetMemberEndForces** (const VARIANT FAR &nMemberNo, const VARIANT FAR &nEnd, const VARIANT FAR &nLC, VARIANT FAR &pdForces, const VARIANT FAR &LocalOrGlobal) Returns member end forces (in local coordinates) for specified member number, member end and load case.
afx_msg VARIANT	**OSOutputUI::GetPMemberEndForces** (const VARIANT FAR &nMemberNo, const VARIANT FAR &nEnd, const VARIANT FAR &nLC, VARIANT FAR &pdForces, const VARIANT FAR &LocalOrGlobal) Returns member end forces for specified physical member number, member end and load case.
afx_msg VARIANT	**OSOutputUI::GetMinMaxBendingMoment** (const VARIANT FAR &nMemberNo, const VARIANT FAR &strDir, const VARIANT FAR &nLoadNo, VARIANT FAR &dMin, VARIANT FAR &dMinPos, VARIANT FAR &dMax, VARIANT FAR &dMaxPos) Returns maximum and minimum bending moments and their locations for specified member number, load case, and bending direction.
afx_msg VARIANT	**OSOutputUI::GetMinMaxShearForce** (const VARIANT FAR &nMemberNo, const VARIANT FAR &strDir, const VARIANT FAR &nLoadNo, VARIANT FAR &dMin, VARIANT FAR &dMinPos, VARIANT FAR &dMax, VARIANT FAR &dMaxPos) Returns maximum and minimum shear force and their locations for specified member number, load case, and force direction.
afx_msg VARIANT	**OSOutputUI::GetMinMaxAxialForce** (const VARIANT FAR &nMemberNo, const VARIANT FAR &nLoadNo, VARIANT FAR &dMin, VARIANT FAR &dMinPos, VARIANT FAR &dMax, VARIANT FAR &dMaxPos) Returns maximum and minimum axial force and their locations for specified member number, load case.
afx_msg VARIANT	**OSOutputUI::GetIntermediateMemberForcesAtDistance** (const VARIANT FAR &nMemberNo, const VARIANT FAR &dDistance, const VARIANT FAR &nLC, VARIANT FAR &pdForces) Returns sectional forces and moments for specified member number, distance, and load case.
afx_msg VARIANT	**OSOutputUI::GetPMemberIntermediateForcesAtDistance** (const VARIANT FAR &nMemberNo, const VARIANT FAR &dDistance, const VARIANT FAR &nLC, VARIANT FAR

Loading [MathJax]/extensions/MathZoom.js

	&pdForces) Returns sectional forces and moments for specified physical member number, distance, and load case.
afx_msg VARIANT	**OSOutputUI::GetMaxSectionDisplacement** (const VARIANT FAR &nMemberNo, const VARIANT FAR &strDir, const VARIANT FAR &nLoadNo, VARIANT FAR &dMax, VARIANT FAR &dMaxPos) Returns the maximum section displacements for specified member number, direction, and load case.
afx_msg VARIANT	**OSOutputUI::GetIntermediateDeflectionAtDistance** (const VARIANT FAR &nMemberNo, const VARIANT FAR &dDistance, const VARIANT FAR &nLC, VARIANT FAR &dY, VARIANT FAR &dZ) Returns the intermediate section deflections for specified member number and load case.
afx_msg VARIANT	**OSOutputUI::GetIntermediateMemberAbsTransDisplacements** (const VARIANT FAR &nMemberNo, const VARIANT FAR &dDistance, const VARIANT FAR &nLC, VARIANT FAR &pdDisps) Returns section displacement (or relative displacements) of a beam section for specified member number, distance, and load case.

Detailed Description

These functions are related to output analysis member results.

Function Documentation

◆ GetIntermediateDeflectionAtDistance()

```
VARIANT OSOutputUI::GetIntermediateDeflectionAtDistance ( const VARIANT FAR & nMemberNo,
                                                          const VARIANT FAR & dDistance,
                                                          const VARIANT FAR & nLC,
                                                          VARIANT FAR &       dY,
                                                          VARIANT FAR &       dZ )
```

Returns the intermediate section deflections for specified member number and load case.

Parameters

[in]	**nMemberNo**	Member number ID.
[in]	**dDistance**	Distance from the starting end of the member.
[in]	**nLC**	Load Case reference ID.
[out]	**dY**	Displacement in Y direction.
[out]	**dZ**	Displacement in Z direction.

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
'Get intermediate deflection of section 1.5 units from starting end of member, for member
       #1, load case 1.
Sub IntermediateDeflectionAtDistance()
  Dim memb As Long
  Dim Lcase As Long
  Dim RetVal As Variant
  Dim dDistance As Double
  Dim DY As Double
  Dim DZ As Double

'Launch OpenSTAAD Object
On Error GoTo ErrHandler
  Set objOpenSTAAD = GetObject(, "StaadPro.OpenSTAAD")

'Is Analysis Completed
  Cells(1, 2).Value = objOpenSTAAD.Output.AreResultsAvailable()

'Intermediate Section Deflections
  memb = Cells(34, 1).Value 'Member ID = 1
  Lcase = Cells(31, 2).Value 'LoadCase = 1
  dDistance = Cells(32, 2).Value 'Distance = 1.5
  RetVal = objOpenSTAAD.Output.GetIntermediateDeflectionAtDistance(memb, dDistance, Lcase,
       DY, DZ)
  Cells(34, 2).Value = DY
  Cells(34, 3).Value = DZ

Set objOpenSTAAD = Nothing
Exit Sub

ErrHandler:
  MsgBox ("Run StaadPro First" & vbCrLf)
Resume Next
End Sub
```

Loading [MathJax]/extensions/MathZoom.js

◆ GetIntermediateMemberAbsTransDisplacements()

Loading [MathJax]/extensions/MathZoom.js

VARIANT OSOutputUI::GetIntermediateMemberAbsTransDisplacements (const VARIANT FAR & nMemberNo,
const VARIANT FAR & dDistance,
const VARIANT FAR & nLC,
VARIANT FAR & pdDisps)

Returns section displacement (or relative displacements) of a beam section for specified member number, distance, and load case.

Parameters

[in]	**nMemberNo**	Member number ID.
[in]	**dDistance**	Distance from starting end in terms of member length.
[in]	**nLC**	Load Case reference ID.
[out]	**pdDisps**	Relative displacements at specified section in terms of LOCAL X, Y, Z coordinates (in order).

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
'Get Section Displacement at End B for Member #1, Load case #1, 1.5 units distant from
       starting end
Sub IntermediateMemberAbsTransDisplacements()
  Dim memb As Long
  Dim Lcase As Long
  Dim RetVal As Variant
  Dim dDistance As Double
  Dim pdDisps(0 To 5) As Double
  Dim count As Integer

'Launch OpenSTAAD Object
On Error GoTo ErrHandler
  Set objOpenSTAAD = GetObject(, "StaadPro.OpenSTAAD")

'Is Analysis Completed
  Cells(1, 2).Value = objOpenSTAAD.Output.AreResultsAvailable()

'Intermediate Member Trans Displacements
  memb = Cells(29, 1).Value 'Member ID = 1
  Lcase = Cells(26, 2).Value 'LoadCase = 1
  dDistance = Cells(27, 2).Value 'Distance = 1.5
  RetVal = objOpenSTAAD.Output.GetIntermediateMemberAbsTransDisplacements(memb, dDistance,
       Lcase, pdDisps)
  For count = 1 To 6
  Cells(29, count + 1).Value = pdDisps(count - 1)
  Next

Set objOpenSTAAD = Nothing
Exit Sub

ErrHandler:
  MsgBox ("Run StaadPro First" & vbCrLf)
Resume Next
End Sub
```

Loading [MathJax]/extensions/MathZoom.js

◆ GetIntermediateMemberForcesAtDistance()

Loading [MathJax]/extensions/MathZoom.js

VARIANT OSOutputUI::GetIntermediateMemberForcesAtDistance (const VARIANT FAR & nMemberNo,
const VARIANT FAR & dDistance,
const VARIANT FAR & nLC,
VARIANT FAR & pdForces)

Returns sectional forces and moments for specified member number, distance, and load case.

Parameters

[in]	**nMemberNo**	Member number ID.
[in]	**dDistance**	Distance from the starting end of the member.
[in]	**nLC**	Load Case reference ID.
[out]	**pdForces**	Array of Section axial force, Shear force in LOCAL Y & Z direction, Torsion and Bending moment in Local MY & MZ direction

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
'Get section forces at section 3 units from member starting end, for member #2, load case
       1.
Sub IntermediateMemberForcesAtDistance()
  Dim memb As Long
  Dim Lcase As Long
  Dim count As Integer
  Dim RetVal As Variant
  Dim dDistance As Double
  Dim pdForces(0 To 5) As Double

'Launch OpenSTAAD Object
On Error GoTo ErrHandler
  Set objOpenSTAAD = GetObject(, "StaadPro.OpenSTAAD")

'Is Analysis Completed
  Cells(1, 2).Value = objOpenSTAAD.Output.AreResultsAvailable()

'Sectional forces and moments for specified member number, distance, and load case.
  memb = Cells(39, 1).Value 'Member ID = 2
  Lcase = Cells(36, 2).Value 'LoadCase = 1
  dDistance = Cells(37, 2).Value 'Distance = 3
  RetVal = objOpenSTAAD.Output.GetIntermediateMemberForcesAtDistance(memb, dDistance,
       Lcase, pdForces)
  For count = 1 To 6
  Cells(39, count + 1).Value = pdForces(count - 1)
  Next

Set objOpenSTAAD = Nothing
Exit Sub

ErrHandler:
  MsgBox ("Run StaadPro First" & vbCrLf)
Resume Next
End Sub
```

◆ GetIntermediateMemberTransDisplacements()

Loading [MathJax]/extensions/MathZoom.js

VARIANT OSOutputUI::GetIntermediateMemberTransDisplacements (const VARIANT FAR & nMemberNo,
const VARIANT FAR & dDistance,
const VARIANT FAR & nLC,
VARIANT FAR & pdDisps)

Returns section displacement (or relative displacements) of a beam section for specified member number, distance, and load case.

Parameters

[in]	**nMemberNo**	Member number ID.
[in]	**dDistance**	Distance from starting end in terms of member length.
[in]	**nLC**	Load Case reference ID.
[out]	**pdDisps**	Relative displacements at specified section in terms of LOCAL X, Y, Z coordinates (in order).

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
'Get Section Displacement at End B for Member #1, Load case #1, 1.5 units distant from
       starting end
Sub IntermediateMemberTransDisplacements()
  Dim memb As Long
  Dim Lcase As Long
  Dim RetVal As Variant
  Dim dDistance As Double
  Dim pdDisps(0 To 5) As Double
  Dim count As Integer

'Launch OpenSTAAD Object
On Error GoTo ErrHandler
  Set objOpenSTAAD = GetObject(, "StaadPro.OpenSTAAD")

'Is Analysis Completed
  Cells(1, 2).Value = objOpenSTAAD.Output.AreResultsAvailable()

'Intermediate Member Trans Displacements
  memb = Cells(29, 1).Value 'Member ID = 1
  Lcase = Cells(26, 2).Value 'LoadCase = 1
  dDistance = Cells(27, 2).Value 'Distance = 1.5
  RetVal = objOpenSTAAD.Output.GetIntermediateMemberTransDisplacements(memb, dDistance,
       Lcase, pdDisps)
  For count = 1 To 6
  Cells(29, count + 1).Value = pdDisps(count - 1)
  Next

Set objOpenSTAAD = Nothing
Exit Sub

ErrHandler:
  MsgBox ("Run StaadPro First" & vbCrLf)
Resume Next
```

Loading [MathJax]/extensions/MathZoom.js

◆ GetMaxSectionDisplacement()

Loading [MathJax]/extensions/MathZoom.js

```
VARIANT OSOutputUI::GetMaxSectionDisplacement ( const VARIANT FAR & nMemberNo,
                                                const VARIANT FAR & strDir,
                                                const VARIANT FAR & nLoadNo,
                                                VARIANT FAR &       dMax,
                                                VARIANT FAR &       dMaxPos )
```

Returns the maximum section displacements for specified member number, direction, and load case.

Parameters

[in]	**nMemberNo**	Member number ID.
[in]	**strDir**	Direction in GLOBAL: X = 1, Y = 2, Z = 3.
[in]	**nLoadNo**	Load Case reference ID.
[out]	**dMax**	Maximum section displacement in specified direction.
[out]	**dMaxPos**	The location along the length of the member where the maximum section displacement is located.

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
'Get maximum section displacements in GLOBAL X direction for member #3, load case 1.
Sub MaxSectionDisplacement()
  Dim memb As Long
  Dim Lcase As Long
  Dim RetVal As Variant
  Dim strDir As Double
  Dim DMax As Double
  Dim dMaxPos As Double

'Launch OpenSTAAD Object
On Error GoTo ErrHandler
  Set objOpenSTAAD = GetObject(, "StaadPro.OpenSTAAD")

'Is Analysis Completed
  Cells(1, 2).Value = objOpenSTAAD.Output.AreResultsAvailable()

'Get maximum section displacements
  memb = Cells(44, 1).Value 'Member ID = 3
  Lcase = Cells(41, 2).Value 'LoadCase = 1
  strDir = Cells(42, 2).Value 'Direction = 1  'GLOBAL: X = 1, Y = 2, Z = 3.
  RetVal = objOpenSTAAD.Output.GetMaxSectionDisplacement(memb, strDir, Lcase, DMax,
        dMaxPos)
  Cells(44, 2).Value = DMax
  Cells(44, 3).Value = dMaxPos

Set objOpenSTAAD = Nothing
Exit Sub

ErrHandler:
  MsgBox ("Run StaadPro First" & vbCrLf)
Resume Next
End Sub
```

Loading [MathJax]/extensions/MathZoom.js

◆ GetMemberEndDisplacements()

```
VARIANT OSOutputUI::GetMemberEndDisplacements ( const VARIANT FAR & nMemberNo,
                                                const VARIANT FAR & nEnd,
                                                const VARIANT FAR & nLC,
                                                VARIANT FAR &       pdDisps )
```

Returns member end displacements for specified member number, member end and load case.

Parameters

[in]	**nMemberNo**	Member number ID. (Type: Long)
[in]	**nEnd**	Member End (0 for starting and 1 for ending). (Type: Long)
[in]	**nLC**	Load Case reference ID. (Type: Long)
[out]	**pdDisps**	Displacements of Member End in terms of X, Y, Z (in order). (Type: Double Array)

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
'Get Member Displacement at Member #10 for End #1 and Load case #1
Sub MemberEndDisplacements()
 Dim memb As Long
 Dim endDis As Long
 Dim Lcase As Long
 Dim eDisp(0 To 2) As Double
 Dim count As Integer
 Dim RetVal As Variant
 Dim objOpenStaad As Object
'Launch OpenSTAAD Object
 On Error GoTo ErrHandler
 Set objOpenStaad = GetObject(, "StaadPro.OpenSTAAD")
'Run Analysis
 RetVal = objOpenStaad.AnalyzeEx(1, 0, 1)
'Is Analysis Completed
 Cells(1, 2).Value = objOpenStaad.Output.AreResultsAvailable()
'Member End Displacements
 memb = 10
 endDis = 1
 Lcase = 1
 RetVal = objOpenStaad.Output.GetMemberEndDisplacements(memb, endDis, Lcase, eDisp)
 For count = 1 To 3
 Cells(18, count + 1).Value = eDisp(count - 1)
 Next
 Set objOpenStaad = Nothing
 Exit Sub
ErrHandler:
 MsgBox ("Run StaadPro First" & vbCrLf)
Resume Next
End Sub
```

Loading [MathJax]/extensions/MathZoom.js

◆ GetMemberEndForces()

Loading [MathJax]/extensions/MathZoom.js

VARIANT OSOutputUI::GetMemberEndForces (const VARIANT FAR & nMemberNo,
const VARIANT FAR & nEnd,
const VARIANT FAR & nLC,
VARIANT FAR & pdForces,
const VARIANT FAR & LocalOrGlobal)

Returns member end forces (in local coordinates) for specified member number, member end and load case.

Parameters

[in]	**nMemberNo**	Member number ID.
[in]	**nEnd**	Member End (0 for starting and 1 for ending).
[in]	**nLC**	Load Case reference ID.
[out]	**pdForces**	Force of Member End in LOCAL coordinates in terms of FX, FY, FZ, MX, MY and MZ (in order).
[in]	**LocalOrGlobal**	Local Or Global direction.

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
'Get Member Force at Member #2 for End #0, LocalOrGlobal #0 and Load case #1
Sub MemberEndForces()
  Dim memb As Long
  Dim Lcase As Long
  Dim endDis As Long
  Dim LoctoGlb As Long
  Dim eForce(0 To 5) As Double
  Dim RetVal As Variant
  Dim count As Integer

'Launch OpenSTAAD Object
On Error GoTo ErrHandler
  Set objOpenSTAAD = GetObject(, "StaadPro.OpenSTAAD")

'Is Analysis Completed
  Cells(1, 2).Value = objOpenSTAAD.Output.AreResultsAvailable()

'Member End Force
  memb = Cells(24, 1).Value 'Member ID = 2
  endDis = Cells(21, 2).Value 'Member End = 0
  Lcase = Cells(20, 2).Value 'LoadCase = 1
  LoctoGlb = Cells(22, 2).Value 'LocalOrGlobal = 0
  RetVal = objOpenSTAAD.Output.GetMemberEndForces(memb, endDis, Lcase, eForce, LoctoGlb)
  For count = 1 To 6
  Cells(24, count + 1).Value = eForce(count - 1)
  Next

Set objOpenSTAAD = Nothing
Exit Sub

ErrHandler:
  MsgBox ("Run StaadPro First" & vbCrLf)
```

Loading [MathJax]/extensions/MathZoom.js

```
End Sub
```

◆ GetMinMaxAxialForce()

Loading [MathJax]/extensions/MathZoom.js

VARIANT OSOutputUI::GetMinMaxAxialForce (const VARIANT FAR & nMemberNo,
const VARIANT FAR & nLoadNo,
VARIANT FAR & dMin,
VARIANT FAR & dMinPos,
VARIANT FAR & dMax,
VARIANT FAR & dMaxPos)

Returns maximum and minimum axial force and their locations for specified member number, load case.

Parameters

[in]	**nMemberNo**	Member number ID.
[in]	**nLoadNo**	Load Case reference ID.
[out]	**dMin**	Minimum axial force.
[out]	**dMinPos**	The location along the length of the member where the minimum axial force is located.
[out]	**dMax**	Maximum axial force.
[out]	**dMaxPos**	The location along the length of the member where the maximum axial force t is located.

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
'Get maximum and minimum Axial Force for member #1, load case 2.
Sub MinMaxAxialForce()
  Dim memb As Long
  Dim Lcase As Long
  Dim RetVal As Variant
  Dim ddMin As Double
  Dim ddMinPos As Double
  Dim ddMax As Double
  Dim ddMaxPos As Double

'Launch OpenSTAAD Object
On Error GoTo ErrHandler
  Set objOpenSTAAD = GetObject(, "StaadPro.OpenSTAAD")

'Is Analysis Completed
  Cells(1, 2).Value = objOpenSTAAD.Output.AreResultsAvailable()

'Get maximum and minimum Axial Force
  memb = Cells(58, 1).Value 'Member ID = 1
  Lcase = Cells(56, 2).Value 'LoadCase = 2
  RetVal = objOpenSTAAD.Output.GetMinMaxAxialForce(memb, Lcase, ddMin, ddMinPos, ddMax,
       ddMaxPos)
  Cells(58, 2).Value = ddMin
  Cells(58, 3).Value = ddMinPos
  Cells(58, 4).Value = ddMax
  Cells(58, 5).Value = ddMaxPos

  Set objOpenSTAAD = Nothing

ErrHandler:
```

Loading [MathJax]/extensions/MathZoom.js

```
  MsgBox ("Run StaadPro First" & vbCrLf)
Resume Next
End Sub
```

◆ GetMinMaxBendingMoment()

Loading [MathJax]/extensions/MathZoom.js

VARIANT OSOutputUI::GetMinMaxBendingMoment (const VARIANT FAR & nMemberNo,
const VARIANT FAR & strDir,
const VARIANT FAR & nLoadNo,
VARIANT FAR & dMin,
VARIANT FAR & dMinPos,
VARIANT FAR & dMax,
VARIANT FAR & dMaxPos)

Returns maximum and minimum bending moments and their locations for specified member number, load case, and bending direction.

Parameters

[in]	**nMemberNo**	Member number ID.
[in]	**strDir**	Bending direction in LOCAL coordinate: MY = 5, MZ = 6;
[in]	**nLoadNo**	Load Case reference ID.
[out]	**dMin**	Minimum bending moment.
[out]	**dMinPos**	The location along the length of the member where the minimum bending moment is located.
[out]	**dMax**	Maximum bending moment.
[out]	**dMaxPos**	The location along the length of the member where the maximum bending moment is located.

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
'Get maximum and minimum bending moments for member #1, load case 2.
Sub MinMaxBendingMoment()
  Dim memb As Long
  Dim Lcase As Long
  Dim strDir As Double
  Dim RetVal As Variant
  Dim ddMin As Double
  Dim ddMinPos As Double
  Dim ddMax As Double
  Dim ddMaxPos As Double

'Launch OpenSTAAD Object
On Error GoTo ErrHandler
  Set objOpenSTAAD = GetObject(, "StaadPro.OpenSTAAD")

'Is Analysis Completed
  Cells(1, 2).Value = objOpenSTAAD.Output.AreResultsAvailable()

'Get maximum and minimum bending moments
  memb = Cells(49, 1).Value 'Member ID = 1
  Lcase = Cells(46, 2).Value 'LoadCase = 2
  strDir = Cells(47, 2).Value 'Direction = 5
                       Output.GetMinMaxBendingMoment(memb, strDir, Lcase, ddMin, ddMinPos,

  Cells(49, 2).Value = ddMin
```

Loading [MathJax]/extensions/MathZoom.js

```
  Cells(49, 3).Value = ddMinPos
  Cells(49, 4).Value = ddMax
  Cells(49, 5).Value = ddMaxPos

  Set objOpenSTAAD = Nothing
Exit Sub

ErrHandler:
  MsgBox ("Run StaadPro First" & vbCrLf)
Resume Next
End Sub
```

◆ GetMinMaxShearForce()

```
VARIANT OSOutputUI::GetMinMaxShearForce ( const VARIANT FAR & nMemberNo,
                                          const VARIANT FAR & strDir,
                                          const VARIANT FAR & nLoadNo,
                                          VARIANT FAR &       dMin,
                                          VARIANT FAR &       dMinPos,
                                          VARIANT FAR &       dMax,
                                          VARIANT FAR &       dMaxPos )
```

Returns maximum and minimum shear force and their locations for specified member number, load case, and force direction.

Parameters

[in] **nMemberNo** Member number ID.

[in] **strDir** Bending direction in LOCAL coordinate: FY = 2, FZ = 3;

[in] **nLoadNo** Load Case reference ID.

[out] **dMin** Minimum shear force.

[out] **dMinPos** The location along the length of the member where the minimum shear force is located.

[out] **dMax** Maximum shear force.

[out] **dMaxPos** The location along the length of the member where the maximum shear force t is located.

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
'Get maximum and minimum Shear Force for member #3, load case 2.
Sub MinMaxShearForce()
  Dim memb As Long
  Dim Lcase As Long
  Dim strDir As Double
  Dim RetVal As Variant
  Dim ddMin As Double
  Dim ddMinPos As Double
  Dim ddMax As Double
  Dim ddMaxPos As Double

'Launch OpenSTAAD Object
On Error GoTo ErrHandler
  Set objOpenSTAAD = GetObject(, "StaadPro.OpenSTAAD")

'Is Analysis Completed
  Cells(1, 2).Value = objOpenSTAAD.Output.AreResultsAvailable()

'Get maximum and minimum Shear Force
  memb = Cells(54, 1).Value 'Member ID = 3
  Lcase = Cells(51, 2).Value 'LoadCase = 2
  strDir = Cells(52, 2).Value 'Direction = 5
  RetVal = objOpenSTAAD.Output.GetMinMaxShearForce(memb, strDir, Lcase, ddMin, ddMinPos,
        ddMax, ddMaxPos)
```

Loading [MathJax]/extensions/MathZoom.js

```
                       dMin
  Cells(54, 3).Value = ddMinPos
  Cells(54, 4).Value = ddMax
  Cells(54, 5).Value = ddMaxPos

  Set objOpenSTAAD = Nothing
Exit Sub

ErrHandler:
  MsgBox ("Run StaadPro First" & vbCrLf)
Resume Next
End Sub
```

◆ GetPMemberEndForces()

```
VARIANT OSOutputUI::GetPMemberEndForces ( const VARIANT FAR & nMemberNo,
                                          const VARIANT FAR & nEnd,
                                          const VARIANT FAR & nLC,
                                          VARIANT FAR &       pdForces,
                                          const VARIANT FAR & LocalOrGlobal )
```

Returns member end forces for specified physical member number, member end and load case.

Parameters

[in]	**nMemberNo**	Member number ID. (Type : Long/Int)
[in]	**nEnd**	Member End (0 for starting and 1 for ending). (Type : Long/Int)
[in]	**nLC**	Load Case reference ID. (Type : Long/Int)
[out]	**pdForces**	Force of Member End in LOCAL coordinates in terms of FX, FY, FZ, MX, MY and MZ (in order). (Type : Double)
[in]	**LocalOrGlobal**	Results returned in either local or global axes. 0= Local, 1= Global direction. (Type : Long/Int)

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
Option Explicit

   Sub Main
       Dim objOpenStaad As Object
       Dim stdFile As String
       Dim forces(5) As Double
       Dim result As Boolean
       Dim pMember As Long
       Dim nEnd As Long
       Dim LoadCase As Long
       Dim LocalOrGlobal As Long
       Dim retVal As Long

       Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
       objOpenStaad.GetSTAADFile stdFile, "TRUE"
       ' perform analysis and design the model and wait for finish
       retVal = objOpenStaad.AnalyzeEx(1, 0, 1)
       pMember = 2
       nEnd = 0
       LoadCase = 1
       LocalOrGlobal = 0
       result =
        objOpenStaad.Output.GetPMemberEndForces(pMember,nEnd,LoadCase,forces,LocalOrGlobal)
   End Sub
```

@Note: When a physical member is created in the model then reordering of the analytical beams takes place. Initially it looks for the beam in the list which have its start end free and places it at the start of the list, so that the free end becomes the start node of the physical member. If the nodal incidences of the first beam and the last beam f the Fx, Fz, Mx and Mz values changes according to the local axes system of the physical member. The local axes system of a physical member has its origin at the start node with +Local-X pointing towards the last node along the member length. This is similar to the local axes system of an analytical beam. For local axes system of analytical beam refer : "GS. Coordinates in STAAD.Pro" of the STAAD.Pro manual.

◆ GetPMemberIntermediateForcesAtDistance()

VARIANT OSOutputUI::GetPMemberIntermediateForcesAtDistance (const VARIANT FAR & nMemberNo,
const VARIANT FAR & dDistance,
const VARIANT FAR & nLC,
VARIANT FAR & pdForces)

Returns sectional forces and moments for specified physical member number, distance, and load case.

Parameters

[in]	**nMemberNo**	Physical Member number ID. (Type : Long/Int)
[in]	**dDistance**	Distance from the starting end of the member. (Type : Double)
[in]	**nLC**	Load Case reference ID. (Type : Long/Int)
[out]	**pdForces**	6 item array of Section axial force, Shear force in LOCAL Y & Z direction, Torsion and Bendin moment in Local MY & MZ direction. (Type : Double)

Returns

Boolean (TRUE/FALSE) whether succeeded or not.

VBA Syntax

```
Sub Main
    Dim objOpenStaad As Object
    Dim stdFile As String
    Dim forces(5) As Double
    Dim result As Boolean
    Dim pMember As Long
    Dim dDistance As Double
    Dim LoadCase As Long
    Dim retVal As Long

    Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
    objOpenStaad.GetSTAADFile stdFile, "TRUE"
    ' perform analysis and design the model and wait for finish
    retVal = objOpenStaad.AnalyzeEx(1, 0, 1)
    pMember = 2
    dDistance = 35.4
    LoadCase = 1
    result =
       objOpenStaad.Output.GetPMemberIntermediateForcesAtDistance(pMember,dDistance,LoadCase,forc
End Sub
```

Loading [MathJax]/extensions/MathZoom.js