Root

Functions

afx_msg void	**OpenSTAADUI::Quit** () This function closes the STAAD.Pro application environment.
afx_msg void	**OpenSTAADUI::Analyze** () This function analyzes the currently opened .STD file. This method is equivalent to running analysis from user interface. For more options, see AnalyzeEx() method.
afx_msg long	**OpenSTAADUI::AnalyzeEx** (const VARIANT FAR &bSilent, const VARIANT FAR &bHidden, const VARIANT FAR &bWait) This extended method analyzes the currently opened .STD file. This method is equivalent to running analysis from user interface. However, it has additional three arguments to specify whether to run the analysis in silent or hidden mode. The third parameter specifies whether the method should wait for the analysis to finish or return immediately. This method may be used in conjunction with SetSilentMode(), if one wants to suppress all dialog boxes displayed from the application during running of analysis.
afx_msg void	**OpenSTAADUI::AnalyzeModel** (const VARIANT FAR &varEngine) Analyze the model currently opened in staad.pro.
afx_msg long	**OpenSTAADUI::GetAnalysisStatus** (LPCTSTR szPathFileName, long FAR *NoOfWarnings, long FAR *NoOfErrors, double FAR *CPUTime) Get analysis status for any STD model.
afx_msg VARIANT	**OpenSTAADUI::IsAnalyzing** () Returns a value to specify whether the analysis is running or not.
afx_msg void	**OpenSTAADUI::UpdateStructure** () This function updates the current structure.
afx_msg void	**OpenSTAADUI::SaveModel** (const VARIANT FAR &saveOption) This function saves the current structure with optional silent mode.
afx_msg void	**OpenSTAADUI::OpenSTAADFile** (const VARIANT FAR &bstrFileName) This function will open the specified .STD file.
afx_msg void	**OpenSTAADUI::NewSTAADFile** (const VARIANT FAR &bstrFileName, const VARIANT FAR &nLenUnitInput, const VARIANT FAR &nForceUnitInput) This function creates a .STD file with specified length and force units.
afx_msg void	**OpenSTAADUI::CloseSTAADFile** () This function closes the currently open .STD file.
afx_msg void	**OpenSTAADUI::GetSTAADFile** (VARIANT FAR &fileName, const VARIANT FAR &bFullPath) Retrieves the name of the current .STD file.
afx_msg void	**OpenSTAADUI::GetSTAADFileFolder** (VARIANT FAR &fileFolder) Retrieves only the path of the current .STD file.

afx_msg VARIANT	**OpenSTAADUI::GetInputUnitForLength** (VARIANT FAR &strUnit) Retrieves the input unit of length of the currently open .STD file.
afx_msg VARIANT	**OpenSTAADUI::GetInputUnitForForce** (VARIANT FAR &strUnit) Retrieves the input unit of force of the currently open .STD file.
afx_msg void	**OpenSTAADUI::SetInputUnitForLength** (const VARIANT FAR &iUnit) This function sets the input unit of length of the currently open .STD file.
afx_msg void	**OpenSTAADUI::SetInputUnitForForce** (const VARIANT FAR &iUnit) This function sets the input unit of force of the currently open .STD file.
afx_msg void	**OpenSTAADUI::SetInputUnits** (const VARIANT FAR &lUnit, const VARIANT FAR &fUnit) This function sets the input unit of length of the currently open .STD file.
afx_msg VARIANT	**OpenSTAADUI::IsPhysicalModel** () Checks if the loaded model is a physical model.
afx_msg VARIANT	**OpenSTAADUI::GetProcessHandle** () This function retrives the current STAAD.Pro process handle.
afx_msg VARIANT	**OpenSTAADUI::GetProcessId** () This function retrives the current STAAD.Pro process ID.
afx_msg VARIANT	**OpenSTAADUI::GetMainWindowHandle** () This function retrives the main STAAD.Pro window handle.
afx_msg void	**OpenSTAADUI::GetShortJobInfo** (VARIANT FAR &jobName, VARIANT FAR &jobClient, VARIANT FAR &enggName) Retrieves the short job information of the currently open .STD file.
afx_msg void	**OpenSTAADUI::SetShortJobInfo** (const VARIANT FAR &jobName, const VARIANT FAR &jobClient, const VARIANT FAR &enggName) This function sets the short job information of the currently open .STD file.
afx_msg void	**OpenSTAADUI::GetFullJobInfo** (VARIANT FAR &jobName, VARIANT FAR &jobClient, VARIANT FAR &enggName, VARIANT FAR &eDate, VARIANT FAR &jobNumber, VARIANT FAR &revision, VARIANT FAR &part, VARIANT FAR &reference, VARIANT FAR &checkerName, VARIANT FAR &cDate, VARIANT FAR &approverName, VARIANT FAR &aDate, VARIANT FAR &comments) Retrieves the full job information of the currently open .STD file.
afx_msg void	**OpenSTAADUI::SetFullJobInfo** (const VARIANT FAR &jobName, const VARIANT FAR &jobClient, const VARIANT FAR &enggName, const VARIANT FAR &eDate, const VARIANT FAR &jobNumber, const VARIANT FAR &revision, const VARIANT FAR &part, const VARIANT FAR &reference, const VARIANT FAR &checkerName, const VARIANT FAR &cDate, const VARIANT FAR &approverName, const VARIANT FAR &aDate, const VARIANT FAR &comments) This function sets the full job information of the currently open .STD file.
afx_msg void	**OpenSTAADUI::CreateNamedView** (const VARIANT FAR &strName, const VARIANT FAR &nFlag, VARIANT FAR &nError) This function creates a view with the specified name.

afx_msg void	**OpenSTAADUI::RemoveNamedView** (const VARIANT FAR &strName, VARIANT FAR &nError) This function removes the current view with the specified name.
afx_msg void	**OpenSTAADUI::SaveNamedView** (const VARIANT FAR &strName, VARIANT FAR &nError) This function saves the current view with the specified name.
afx_msg void	**OpenSTAADUI::ModifyNamedView** (const VARIANT FAR &strName, const VARIANT FAR &nEntities, const VARIANT FAR &EntityArray, const VARIANT FAR &nArrayQualifier, const VARIANT FAR &nModifyFlag, VARIANT FAR &nError) This function saves the current view with the specified name.
afx_msg VARIANT	**OpenSTAADUI::GetBaseUnit** () Returns the base unit for the currently open .STD file.
afx_msg VARIANT	**OpenSTAADUI::SetSilentMode** (const VARIANT FAR &varFlag) Sets the silent mode of the application. Calling this method with 1, will suppress the display of warning message boxes during analysis and saving of the file For example, warning messages like empty loads, repeat load expansions etc. Note, if any error occurs like license not available, or folder inaccessible, message boxes will be displayed appropriately.
afx_msg VARIANT	**OpenSTAADUI::GetErrorMessage** () Returns error messages thrown by **OpenSTAAD** (e.g. - for unavailability of license, unavailability of required named view)
afx_msg VARIANT	**OpenSTAADUI::GetApplicationVersion** (VARIANT FAR &MajorA, VARIANT FAR &MajorB, VARIANT FAR &Minor, VARIANT FAR &Build) Returns the application version number as text and in individual parts.
afx_msg VARIANT	**OpenSTAADUI::GetCONNECTEDProjectInfo** (VARIANT FAR &szProjID, VARIANT FAR &szName) Get CONNECTED Project ID and name.
afx_msg VARIANT	**OpenSTAADUI::SetCONNECTEDProjectInfo** (const VARIANT FAR &szProjID, const VARIANT FAR &szName) Set CONNECTED Project ID and name.

Detailed Description

These functions are related to Root Functions.

Function Documentation

◆ Analyze()

void OpenSTAADUI::Analyze ( )

This function analyzes the currently opened .STD file. This method is equivalent to running analysis from user interface. For more options, see AnalyzeEx() method.

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Run Analysis
objOpenSTAAD.Analyze
```

See also

OpenSTAADUI::AnalyzeEx

OpenSTAADUI::SetSilentMode

◆ AnalyzeEx()

```
long OpenSTAADUI::AnalyzeEx ( const VARIANT FAR & varSilent,
                              const VARIANT FAR & varHidden,
                              const VARIANT FAR & varWait )
```

This extended method analyzes the currently opened .STD file. This method is equivalent to running analysis from user interface. However, it has additional three arguments to specify whether to run the analysis in silent or hidden mode. The third parameter specifies whether the method should wait for the analysis to finish or return immediately. This method may be used in conjunction with SetSilentMode(), if one wants to suppress all dialog boxes displayed from the application during running of analysis.

Parameters

[in] **varSilent** Integer value to enable silent mode. [1 = Enable, 0 otherwise]. Enabling silent mode will suppress all dialog boxes in the engine which requires user input. The analysis dialog box however will be displayed and close automatically on completion.

[in] **varHidden** Integer value to enable hidden mode. [1 = Enable, 0 = Disable]. Enabling hidden mode will suppress the display of analysis dialog. The analysis dialog box will not be displayed.

[in] **varWait** Integer value to specify whether to wait for the analysis process to finish or return immediately. [1 to wait , 0 otherwise]

Return values

-1	Analysis Terminated
0	General Error
1	Analysis is in progress
2	Analysis completed without errors or warnings
3	Analysis completed with warnings but without errors
4	Analysis completed with errors
5	Analysis has not been performed

VBA Syntax

```
Sub Main
    Dim objOpenStaad As Object
    Dim stdFile As String
    Dim RetVal as long

    ' Get the running instantce of STAAD.Pro
    Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")

    ' Check if file is opened or not
    objOpenStaad.GetSTAADFile stdFile, "TRUE"
    If stdFile="" Then
        MsgBox"Bad"
        Set objOpenStaad = Nothing
        Exit Sub
    End If
    ' Set silent mode for the main application
    objOpenStaad.SetSilentMode 1
    ' run analysis in silent mode and wait for it to finish
    RetVal = objOpenStaad.AnalyzeEx(1, 0, 1)

    ' Check if results are available or not
    If objOpenStaad.Output.AreResultsAvailable <> 0 Then
        MsgBox "Results Available"
    End If

    MsgBox"Macro Ending"
    Set objOpenStaad = Nothing
End Sub
```

See also

OpenSTAADUI::SetSilentMode

OpenSTAADUI::GetAnalysisStatus

◆ AnalyzeModel()

void OpenSTAADUI::AnalyzeModel (const VARIANT FAR & varEngine)

Analyze the model currently opened in staad.pro.

Parameters

`[in]` **varEngine** Put -1 for launching STAAD.Pro engine.

C++ Syntax

```
//Analyze the currently loaded model
OpenSTAADUI::AnalyzeModel (-1);
```

VBA Syntax

```
'Analyze the currently loaded model
objOpenStaad.AnalyzeModel (-1)
```

C# Syntax

```
//Analyze the currently loaded model
long RetVal = m_OStd.AnalyzeModel (-1);
```

See also

OpenSTAADUI::AnalyzeEx

OpenSTAADUI::Analyze

◆ CloseSTAADFile()

void OpenSTAADUI::CloseSTAADFile ( )

This function closes the currently open .STD file.

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Close current STAAD file
objOpenSTAAD.CloseSTAADFile
```

◆ CreateNamedView()

```
void OpenSTAADUI::CreateNamedView ( const VARIANT FAR & strName,
                                    const VARIANT FAR & nFlag,
                                    VARIANT FAR &       nError )
```

This function creates a view with the specified name.

Parameters

[in] **strName**	A string variable that will hold the name of the view to be created.
[in] **nFlag**	A long variable that will hold the flag value depending upon which the view will be created.
[in] **nError**	A long variable that will hold the error number if the view cannot be created.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim strName As String
Dim nFlag As Long
Dim nError As Long

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Create Named View
objOpenSTAAD.CreateNamedView strName, nFlag, nError
```

◆ GetAnalysisStatus()

```
long OpenSTAADUI::GetAnalysisStatus ( LPCTSTR       szModelNameWithPath,
                                      long FAR *    NoOfWarnings,
                                      long FAR *    NoOfErrors,
                                      double FAR *  CPUTime )
```

Get analysis status for any STD model.

Parameters

`[in]`	**szModelNameWithPath**	Name of the model for which analysis status is required.
`[out]`	**NoOfWarnings**	No. of warnings produced by the STAAD engine while analyzing.
`[out]`	**NoOfErrors**	No. of errors produced by the STAAD engine while analyzing.
`[out]`	**CPUTime**	Time required by the STAAD engine to analyze the model in seconds.

Return values

-2	Invalid model path
-1	Analysis Terminated
0	General Error
1	Analysis is in progress
2	Analysis completed without errors or warnings
3	Analysis completed with warnings but without errors
4	Analysis completed with errors
5	Analysis has not been performed

C++ Syntax

```
//Get Analysis Status for a STD model
long retVal = OpenSTAADUI::GetAnalysisStatus (FileNameWithPath, &noOfWarnings,
       &NoofErrors, &CPUTime);
```

VBA Syntax

```
'Get analysis status for a STD model
Dim NoOfWarnings As Long
Dim NoOfErrors As Long
Dim CPUTime As Double
Dim RetVal As Long
Dim szModelNameWithPath As String
szModelNameWithPath = "C:\Users\Public\Documents\STAAD.Pro CONNECT Edition\Samples\Sample
       Models\US\US-1 Plane Frame with Steel Design.STD"
RetVal = objOpenStaad.GetAnalysisStatus (szModelNameWithPath, NoOfWarnings, NoOfErrors,
       CPUTime)
```

C# Syntax

```
//Get Analysis Status for a STD model
```

```
long RetVal = m_OStd.GetAnalysisStatus (FileNameWithPath, ref NoOfWarnings, ref
       NoofErrors, ref CPUTime);
```

◆ GetApplicationVersion()

VARIANT OpenSTAADUI::GetApplicationVersion (VARIANT FAR & MajorA,
VARIANT FAR & MajorB,
VARIANT FAR & Minor,
VARIANT FAR & Build)

Returns the application version number as text and in individual parts.

Parameters

[out] **MajorA** An integer value representing the version number

[out] **MajorB** An integer value representing the update number

[out] **Minor** An integer value representing the revision number

[out] **Build** An integer value representing the build number

Return values

The current application version as text

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")

Dim major As Long
Dim update As Long
Dim revision As Long
Dim build As Long
Dim version As String

' Get the version
version = objOpenStaad.GetApplicationVersion(major, update, revision, build)
```

◆ GetBaseUnit()

VARIANT OpenSTAADUI::GetBaseUnit ( )

Returns the base unit for the currently open .STD file.

Return values

1 (Long/Integer) Value will return 1 for English system of units (The values that are derived from a length unit, e.g. dimensions, areas, stresses, will be based on inches, 'in'. All values derived from a force unit, e.g. Axial force, moments, stresses, etc, will be based on kilopounds, 'KIP').

2 (Long/Integer) Value will return 2 for Metric system of units (The values that are derived from a length unit, will be based on Meters, 'm'. All values derived from a force unit, will be based on kilo newtons, 'kNs').

C++ Syntax

```
//Get Base Unit
long retVal = OpenSTAADUI::GetBaseUnit();
```

C# Syntax

```
OpenSTAADUI.OpenSTAAD os = Marshal.GetActiveObject("StaadPro.OpenSTAAD") as
       OpenSTAADUI.OpenSTAAD;
//Get Base Unit
long retVal = os.GetBaseUnit();
```

VBA Syntax

```
Option Explicit

   Sub Main
       Dim objOpenStaad As Object
       Dim stdFile As String
       Dim nReturn As Long

       Set objOpenStaad = GetObject(,"StaadPro.OpenSTAAD")
       objOpenStaad.GetSTAADFile stdFile, "TRUE"
       nReturn = objOpenStaad.GetBaseUnit()
   End Sub
```

◆ GetCONNECTEDProjectInfo()

VARIANT OpenSTAADUI::GetCONNECTEDProjectInfo (VARIANT FAR & szProjID,
VARIANT FAR & szName)

Get CONNECTED Project ID and name.

Parameters

[out]	**szProjID**	ID of the CONNECTED Project
[out]	**szName**	Name of the CONNECTED Project

Return values

1	True.
0	False.

VBA Syntax

```
'Get Connected project ID
Dim RetVal as Variant
Dim szProjID as String
Dim szName as String
RetVal = objOpenSTAAD.GetCONNECTEDProjectInfo(szProjID, szName)
```

See also

OpenSTAADUI::SetCONNECTEDProjectInfo

◆ GetErrorMessage()

VARIANT OpenSTAADUI::GetErrorMessage ( )

Returns error messages thrown by **OpenSTAAD** (e.g. - for unavailability of license, unavailability of required named view)

Return values

errmsg	Error message thrown by **OpenSTAAD**

VBA Syntax

```
'Get the error message
Dim errmsg As String
errmsg = objOpenStaad.GetErrorMessage()
```

◆ GetFullJobInfo()

```
void OpenSTAADUI::GetFullJobInfo ( VARIANT FAR & jobName,
                                   VARIANT FAR & jobClient,
                                   VARIANT FAR & enggName,
                                   VARIANT FAR & eDate,
                                   VARIANT FAR & jobNumber,
                                   VARIANT FAR & revision,
                                   VARIANT FAR & part,
                                   VARIANT FAR & reference,
                                   VARIANT FAR & checkerName,
                                   VARIANT FAR & cDate,
                                   VARIANT FAR & approverName,
                                   VARIANT FAR & aDate,
                                   VARIANT FAR & comments )
```

Retrieves the full job information of the currently open .STD file.

Parameters

`[in]` **jobName** A string variable that will hold the Job Name for the current .STD file.

`[in]` **jobClient** A string variable that will hold the Job Client for the current .STD file.

`[in]` **enggName** A string variable that will hold the Engineer’s Name for the current .STD file.

`[in]` **eDate** A string variable that will hold the Engineer Date for the current .STD file.

`[in]` **jobNumber** A string variable that will hold the Job Number for the current .STD file.

`[in]` **revision** A string variable that will hold the Job Revision for the current .STD file.

`[in]` **part** A string variable that will hold the Job Part Name for the current .STD file.

`[in]` **reference** A string variable that will hold the Job Reference for the current .STD file.

`[in]` **checkerName** A string variable that will hold the Checker Name for the current .STD file.

`[in]` **cDate** A string variable that will hold the Checker Date for the current .STD file.

`[in]` **approverName** A string variable that will hold the Approver Name for the current .STD file.

`[in]` **aDate** A string variable that will hold the Approved Date for the current .STD file.

`[in]` **comments** A string variable that will hold the Job Comments for the current .STD file.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim jobName As String
Dim jobClient As String
Dim enggName As String
Dim eDate As String
Dim jobNumber As String
Dim revision As String
Dim part As String
Dim reference As String
Dim checkerName As String
Dim chDate As String
Dim approverName As String
Dim aDate As String
Dim comments As String

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Get Job Info
objOpenSTAAD.GetFullJobInfo jobName, jobClient, enggName, eDate, jobNumber, revision,
       part, reference, checkerName, chDate, approverName, aDate, comments
```

◆ GetInputUnitForForce()

VARIANT OpenSTAADUI::GetInputUnitForForce (VARIANT FAR & strUnit)

Retrieves the input unit of force of the currently open .STD file.

Parameters

`[in]` **strUnit** A string variable that holds the input unit for force of the currently open .STD file. Later the value is internally converted to an integer ranging from 0 to 7 0- Kilopound 1- Pound 2- Kilogram 3- Metric Ton 4- Newton 5- KiloNewton 6- MegaNewton 7- DecaNewton

Return values

1 (Boolean) True (1) if the function is successful.

0 (Boolean) False (0) if the function is not successful.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim strUnit As String

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Get Force  Unit
objOpenSTAAD.GetInputUnitForForce strUnit
```

◆ GetInputUnitForLength()

VARIANT OpenSTAADUI::GetInputUnitForLength (VARIANT FAR & strUnit)

Retrieves the input unit of length of the currently open .STD file.

Parameters

`[in]` **strUnit** A string variable that will hold the input unit for length of the currently open .STD file. Later the value will be internally converted to an integer ranging from 0 to 7 (0- Inch, 1- Feet, 2- Feet, 3- CentiMeter, 4- Meter, 5- MilliMeter, 6 - DeciMeter, 7 – KiloMeter).

Return values

1 (Boolean) True (1) if the function is successful.

0 (Boolean) False (0) if the function is not successful.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim strUnit As String

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Get Length Unit
objOpenSTAAD.GetInputUnitForLength strUnit
```

◆ GetMainWindowHandle()

VARIANT OpenSTAADUI::GetMainWindowHandle ( )

This function retrives the main STAAD.Pro window handle.

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Get Main Window Handle
objOpenSTAAD.GetMainWindowHandle
```

◆ GetProcessHandle()

VARIANT OpenSTAADUI::GetProcessHandle ( )

This function retrives the current STAAD.Pro process handle.

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Get Process Handle
objOpenSTAAD.GetProcessHandle
```

◆ GetProcessId()

VARIANT OpenSTAADUI::GetProcessId ( )

This function retrives the current STAAD.Pro process ID.

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Get Process ID
objOpenSTAAD.GetProcessId
```

◆ GetShortJobInfo()

```
void OpenSTAADUI::GetShortJobInfo ( VARIANT FAR & jobName,
                                    VARIANT FAR & jobClient,
                                    VARIANT FAR & enggName )
```

Retrieves the short job information of the currently open .STD file.

Parameters

`[in]` **jobName** A string variable that will hold the Job Name for the current .STD file.

`[in]` **jobClient** A string variable that will hold the Job Client for the current .STD file.

`[in]` **enggName** A string variable that will hold the Engineer's Name for the current .STD file.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim jobName As String
Dim jobClient As String
Dim enggName As String

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Get Job Info
objOpenSTAAD.GetShortJobInfo jobName, jobClient, enggName
```

◆ GetSTAADFile()

void OpenSTAADUI::GetSTAADFile (VARIANT FAR & fileName,
const VARIANT FAR & bFullPath)

Retrieves the name of the current .STD file.

Parameters

[out] **fileName** A string variable that will hold the name or path of the currently open .STD file.

[in] **bFullPath** A Boolean variable which if true, will write the entire path of the .STD file in the variable FileName or else only the file name.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim fileName As String
Dim bFullPath As Boolean

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Retrieve the entire path
bFullPath = true
objOpenSTAAD.GetSTAADFile fileName, bFullPath
```

◆ GetSTAADFileFolder()

void OpenSTAADUI::GetSTAADFileFolder (VARIANT FAR & fileFolder)

Retrieves only the path of the current .STD file.

Parameters

[in] **fileFolder** A string variable that will hold the path name of folder where the currently open .STD file resides. It will not write the name of the .STD file into the variable.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim fileFolder As String

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Get the file folder
objOpenSTAAD.GetSTAADFileFolder fileFolder
```

◆ IsAnalyzing()

VARIANT OpenSTAADUI::IsAnalyzing ( )

Returns a value to specify whether the analysis is running or not.

Return values

Returns 1 if analysis is still running, 0 otherwise.

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Run Analysis

If objOpenStaad.IsAnalyzing <> 0 Then
    MsgBox "Analysis is running"
End If
```

◆ IsPhysicalModel()

VARIANT OpenSTAADUI::IsPhysicalModel ( )

Checks if the loaded model is a physical model.

Return values

1 True

0 False

C++ Syntax

```
bool RetVal = OpenSTAADUI::IsPhysicalModel();
```

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Checks whether the loaded model is a physical model
Dim retVal As Variant
retVal = objOpenSTAAD.IsPhysicalModel()
```

◆ ModifyNamedView()

void OpenSTAADUI::ModifyNamedView (const VARIANT FAR & strName,
const VARIANT FAR & nEntities,
const VARIANT FAR & EntityArray,
const VARIANT FAR & nArrayQualifier,
const VARIANT FAR & nModifyFlag,
VARIANT FAR & nError)

This function saves the current view with the specified name.

Parameters

- `[in]` **strName** A string variable that will hold the name of the view to be modified.
- `[in]` **nEntities** An long variable that will hold number of entities.
- `[in]` **EntityArray** An long that will hold entity number.
- `[in]` **nArrayQualifier** A integer variable that will hold entity qualifier value. Value may vary from 0 to 4(0 - Node, 1 - Beam, 2 - Plate, 3 - Solid, 4 – Surface)
- `[in]` **nModifyFlag** A long variable that will hold the flag value depending upon which the view will be modified.
- `[in]` **nError** A long variable that will hold the error number if the view cannot be modified.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim strName As String
Dim nEntities As Integer
Dim EntityArray as Long
Dim nArrayQualifier as Long
Dim nModifyFlag as Long
Dim nError as Long

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Modify Named View
objOpenSTAAD.ModifyNamedView strName, nEntities, EntityArray, nArrayQualifier,
       nModifyFlag, nError
```

◆ NewSTAADFile()

```
void OpenSTAADUI::NewSTAADFile ( const VARIANT FAR & bstrFileName,
                                 const VARIANT FAR & nLenUnitInput,
                                 const VARIANT FAR & nForceUnitInput )
```

This function creates a .STD file with specified length and force units.

Parameters

`[in]` **bstrFileName** A string variable that will hold the name of the .STD file, which needs to be created.

`[in]` **nLenUnitInput** An integer variable that will hold the input unit to be assigned for length of the new .STD file. Value may vary from 0 to 7 (0- Inch, 1- Feet, 2- Feet, 3- CentiMeter, 4- Meter, 5- MilliMeter, 6 - DeciMeter, 7 – KiloMeter).

`[in]` **nForceUnitInput** An integer variable that will hold the input unit to be assigned for force of the new .STD file. Value may vary from 0 to 7 (0- Kilopound, 1- Pound, 2- Kilogram, 3- Metric Ton, 4- Newton, 5-Kilo Newton, 6- Mega Newton, 7- DecaNewton).

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim bstrFileName As String
Dim nLenUnitInput As Integer
Dim nForceUnitInput As Integer

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Create New File
objOpenSTAAD.NewSTAADFile bstrFileName, nLenUnitInput, nForceUnitInput
```

◆ OpenSTAADFile()

void OpenSTAADUI::OpenSTAADFile (const VARIANT FAR & bstrFileName)

This function will open the specified .STD file.

Parameters

[in] **bstrFileName** A string variable that will hold the name of the .STD file, which needs to be open.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim bstrFileName As String

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Open the file
objOpenSTAAD.OpenSTAADFile bstrFileName
```

◆ Quit()

void OpenSTAADUI::Quit ( )

This function closes the STAAD.Pro application environment.

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Quit Application environment
objOpenSTAAD.Quit
```

◆ RemoveNamedView()

void OpenSTAADUI::RemoveNamedView (const VARIANT FAR & strName,
VARIANT FAR & nError)

This function removes the current view with the specified name.

Parameters

- [in] **strName** A string variable that will hold the name of the view to be removed.
- [in] **nError** A long variable that will hold the error number if the view cannot be removed.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim strName As String
Dim nError As Long

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Remove Named View
objOpenSTAAD.RemoveNamedView strName, nError
```

◆ SaveModel()

void OpenSTAADUI::SaveModel (const VARIANT FAR & bSilentOpt)

This function saves the current structure with optional silent mode.

Parameters

- [in] **varSilent** Integer value to enable silent mode. [1 = Enable, 0 = Disable, default = 0]

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Save structure silently
objOpenSTAAD.SaveModel 1
```

◆ SaveNamedView()

void OpenSTAADUI::SaveNamedView (const VARIANT FAR & strName,
VARIANT FAR & nError)

This function saves the current view with the specified name.

Parameters

[in] **strName** A string variable that will hold the name of the view to be saved.

[in] **nError** A long variable that will hold the error number if the view cannot be saved.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim strName As String
Dim nError As Long

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Save Named View
objOpenSTAAD.SaveNamedView strName, nError
```

◆ SetCONNECTEDProjectInfo()

VARIANT OpenSTAADUI::SetCONNECTEDProjectInfo (const VARIANT FAR & szProjID,
const VARIANT FAR & szName)

Set CONNECTED Project ID and name.

Parameters

[in] **szProjID** ID of the CONNECTED Project

[in] **szName** Name of the CONNECTED Project

Return values

1 True.

0 False.

VBA Syntax

```
'Set Connected project ID
Dim RetVal as Variant
Dim szProjID as String
Dim szName as String
szProjID = "1A2267AB-04D5-4385-93A3-6A3B872A5859"
szName = "TestProject"
RetVal = objOpenSTAAD.SetCONNECTEDProjectInfo(szProjID, szName)
```

See also

OpenSTAADUI::GetCONNECTEDProjectInfo

◆ SetFullJobInfo()

```
void OpenSTAADUI::SetFullJobInfo ( const VARIANT FAR & jobName,
                                   const VARIANT FAR & jobClient,
                                   const VARIANT FAR & enggName,
                                   const VARIANT FAR & eDate,
                                   const VARIANT FAR & jobNumber,
                                   const VARIANT FAR & revision,
                                   const VARIANT FAR & part,
                                   const VARIANT FAR & reference,
                                   const VARIANT FAR & checkerName,
                                   const VARIANT FAR & cDate,
                                   const VARIANT FAR & approverName,
                                   const VARIANT FAR & aDate,
                                   const VARIANT FAR & comments )
```

This function sets the full job information of the currently open .STD file.

Parameters

[in] **jobName** A string variable that will hold the Job Name for the current .STD file.

[in] **jobClient** A string variable that will hold the Job Client for the current .STD file.

[in] **enggName** A string variable that will hold the Engineer’s Name for the current .STD file.

[in] **eDate** A string variable that will hold the Engineer Date for the current .STD file.

[in] **jobNumber** A string variable that will hold the Job Number for the current .STD file.

[in] **revision** A string variable that will hold the Job Revision for the current .STD file.

[in] **part** A string variable that will hold the Job Part Name for the current .STD file.

[in] **reference** A string variable that will hold the Job Reference for the current .STD file.

[in] **checkerName** A string variable that will hold the Checker Name for the current .STD file.

[in] **cDate** A string variable that will hold the Checker Date for the current .STD file.

[in] **approverName** A string variable that will hold the Approver Name for the current .STD file.

[in] **aDate** A string variable that will hold the Approved Date for the current .STD file.

[in] **comments** A string variable that will hold the Job Comments for the current .STD file.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim jobName As String
Dim jobClient As String
Dim enggName As String
Dim eDate As String
Dim jobNumber As String
Dim revision As String
Dim part As String
Dim reference As String
Dim checkerName As String
Dim chDate As String
Dim approverName As String
Dim aDate As String
Dim comments As String

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Set Job Info
objOpenSTAAD.SetFullJobInfo jobName, jobClient, enggName, eDate, jobNumber, revision,
       part, reference, checkerName, chDate, approverName, aDate, comments
```

SetInputUnitForForce()

void OpenSTAADUI::SetInputUnitForForce (const VARIANT FAR & iUnit)

This function sets the input unit of force of the currently open .STD file.

Parameters

[in] **iUnit** An integer variable that will hold the input unit to be assigned for force of the currently open .STD file. Value may vary from 0 to 7 (0- Kilopound, 1- Pound, 2- Kilogram, 3-Metric Ton, 4- Newton, 5-Kilo Newton, 6- Mega Newton, 7- DecaNewton).

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim iUnit As Integer

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Set Force  Unit
objOpenSTAAD.SetInputUnitForForce iUnit
```

SetInputUnitForLength()

void OpenSTAADUI::SetInputUnitForLength (const VARIANT FAR & iUnit)

This function sets the input unit of length of the currently open .STD file.

Parameters

[in] **iUnit** An integer variable that will hold the input unit to be assigned for length of the currently open .STD file. Value may vary from 0 to 7 (0- Inch, 1- Feet, 2- Feet, 3- CentiMeter, 4- Meter, 5- MilliMeter, 6 - DeciMeter, 7 – KiloMeter).

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim iUnit As Integer

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Set Length Unit
objOpenSTAAD.SetInputUnitForLength iUnit
```

◆ SetInputUnits()

void OpenSTAADUI::SetInputUnits (const VARIANT FAR & lUnit,
const VARIANT FAR & fUnit)

This function sets the input unit of length of the currently open .STD file.

Parameters

[in] **lUnit** An integer variable that will hold the input unit to be assigned for length of the currently open .STD file. Value may vary from 0 to 7 (0- Inch, 1- Feet, 2- Feet, 3- CentiMeter, 4- Meter, 5- MilliMeter, 6 - DeciMeter, 7 – KiloMeter).

[in] **fUnit** An integer variable that will hold the input unit to be assigned for force of the currently open .STD file. Value may vary from 0 to 7 (0- Kilopound, 1- Pound, 2- Kilogram, 3-Metric Ton, 4- Newton, 5-Kilo Newton, 6- Mega Newton, 7- DecaNewton).

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim lUnit As Integer
Dim fUnit As Integer

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Set Input Units
objOpenSTAAD.SetInputUnits lUnit, fUnit
```

◆ SetShortJobInfo()

void OpenSTAADUI::SetShortJobInfo (const VARIANT FAR & jobName,
const VARIANT FAR & jobClient,
const VARIANT FAR & enggName)

This function sets the short job information of the currently open .STD file.

Parameters

[in] **jobName** A string variable that will hold the Job Name for the current .STD file.

[in] **jobClient** A string variable that will hold the Job Client for the current .STD file.

[in] **enggName** A string variable that will hold the Engineer’s Name for the current .STD file.

VBA Syntax

```
Dim objOpenSTAAD As Object
Dim jobName As String
Dim jobClient As String
Dim enggName As String

'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Set Job Info
objOpenSTAAD.SetShortJobInfo jobName, jobClient, enggName
```

◆ SetSilentMode()

VARIANT OpenSTAADUI::SetSilentMode (const VARIANT FAR & varFlag)

Sets the silent mode of the application. Calling this method with 1, will suppress the display of warning message boxes during analysis and saving of the file For example, warning messages like empty loads, repeat load expansions etc. Note, if any error occurs like license not available, or folder inaccessible, message boxes will be displayed appropriately.

Parameters

`[in]` **varFlag** An integer value 1 to set silent mode on and 0 to set it off

Return values

The existing value of the silent mode

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Set silent mode ON
objOpenSTAAD.SetSilentMode 1
'Run analysis engine silently
objopenstaad.AnalyzeEx 1, 0
```

See also

OpenSTAADUI::AnalyzeEx

OpenSTAADUI::Analyze

◆ UpdateStructure()

void OpenSTAADUI::UpdateStructure ( )

This function updates the current structure.

VBA Syntax

```
Dim objOpenSTAAD As Object
'Get the application object
Set objOpenSTAAD = GetObject( , "StaadPro.OpenSTAAD")
'Update structure
objOpenSTAAD.UpdateStructure
```