Load Case Details: Load Combination

Load » **Load: Load Case Details**

Functions

afx_msg VARIANT	**OSLoadUI::GetLoadCombinationCaseCount** () Gets total number of combination load case(s) present in the current structure.
afx_msg VARIANT	**OSLoadUI::GetLoadCombinationCaseNumbers** (VARIANT FAR &lCases) Gets all load combination case number(s).
afx_msg VARIANT	**OSLoadUI::CreateNewLoadCombination** (const VARIANT FAR &varLoadCombTitle, const VARIANT FAR &varLoadCombNo) Creates new load combination with the number and title defined.
afx_msg VARIANT	**OSLoadUI::AddLoadAndFactorToCombination** (const VARIANT FAR &varLoadCombNo, const VARIANT FAR &varLoadNo, const VARIANT FAR &varFactor) Adds a primary load case with specified multiplication factor to an existing load combination.
afx_msg VARIANT	**OSLoadUI::GetNoOfLoadAndFactorPairsForCombination** (const VARIANT FAR &varLoadCombNo) Gets the number of load case(s) applied with multiplication factor in specified load combination.
afx_msg VARIANT	**OSLoadUI::GetLoadAndFactorForCombination** (const VARIANT FAR &varLoadCombNo, VARIANT FAR &varLoadNos, VARIANT FAR &varFactors) Gets load case reference number ID(s) and corresponding multiplication factor(s) for specified load combination.
afx_msg VARIANT	**OSLoadUI::IsCombinationCase** (long nLoadCase) Checks if specified load case is combination load case.
afx_msg VARIANT	**OSLoadUI::AddAutoLoadCombinations** (const VARIANT FAR &varCode, const VARIANT FAR &varCategory, const VARIANT FAR &varLoadList, VARIANT FAR &varStartLoadCaseNo) Automatically adds load combination based on assigned design code and Category.
VARIANT	**OSLoadUI::AddAutoCombinationRepeat** (const VARIANT FAR &varCode, const VARIANT FAR &varCategory, const VARIANT FAR &varLoadList, VARIANT FAR &varStartLoadCaseNo, VARIANT FAR &VarGeneratedLCS, const VARIANT FAR &bVarReference, const VARIANT FAR &bVarNotional, const VARIANT FAR &dVarFactor, const VARIANT FAR &bVarGB50017, const VARIANT FAR &nVarFloor, const VARIANT FAR &bVarX, const VARIANT FAR &bVarNegtiveX, const VARIANT FAR &bVarZ, const VARIANT FAR &bVarNegtiveZ) Automatically adds repeat load based on assigned design code and Category.

Detailed Description

These functions are related to Load Combination.

Function Documentation

◆ AddAutoCombinationRepeat()

```
VARIANT OSLoadUI::AddAutoCombinationRepeat ( const VARIANT FAR & varCode,
                                             const VARIANT FAR & varCategory,
                                             const VARIANT FAR & varLoadList,
                                             VARIANT FAR &       varStartLoadCaseNo,
                                             VARIANT FAR &       VarGeneratedLCS,
                                             const VARIANT FAR & bVarReference,
                                             const VARIANT FAR & bVarNotional,
                                             const VARIANT FAR & dVarFactor,
                                             const VARIANT FAR & bVarGB50017,
                                             const VARIANT FAR & nVarFloor,
                                             const VARIANT FAR & bVarX,
                                             const VARIANT FAR & bVarNegtiveX,
                                             const VARIANT FAR & bVarZ,
                                             const VARIANT FAR & bVarNegtiveZ )
```

Automatically adds repeat load based on assigned design code and Category.

Parameters

[in]	**varCode**	Load Combination Code string name (refer to "Codes.ini") (type - String).
[in]	**varCategory**	Load Combination Category string name (refer to corresponding rule ini file defined in "Codes.ini") (type - String).
[in]	**varLoadList**	Load case reference ID(s), Array of Load case numbers. If the array is either null or empty then all load cases in current model will be considered (type - Long/Integer).
[out]	**varStartLoadCaseNo**	(Repeat Load) load case reference ID with which automatically generation starts (type - Long/Integer). \t If nStartLoadCaseNo is valid, auto repeat load will be created from the provided ID. \t If nStartLoadCaseNo is invalid Load Case ID/already present Load Case ID, the repeat load would automatically generated from next available Load Case ID and nStartLoadCaseNo will be returned/updated with this ID.
[out]	**varGeneratedLCS**	(Repeat Load) The counts of automatically generated repeat loads.
[in]	**bVarReference**	Whether include Reference load
[in]	**bVarNotional**	Whether include Notional load. If it's True but all Directions are Flase, return -1.
[in]	**dVarNotionalLoadFactor**	If bVarNotional is valid, the value of Notional load factor
[in]	**bVarGB50017**	Consider Notional load factor per GB 50017 Design code
[in]	**nVarFloor**	The count of floor, it is valid when bVarGB50017 is True only
[in]	**bVarX**	Consider X Direction of Notional Load
[in]	**bVarNegtiveX**	Consider -X Direction of Notional Load
[in]	**bVarZ**	Consider Z Direction of Notional Load
[in]	**bVarNegtiveZ**	Consider -Z Direction of Notional Load

Return values

0 if successful.

-1 if unsuccessful

Remarks

The default path of Codes.ini under "%localappdata%\Bentley\Engineering\STAAD.Pro 2023\Default\Language\en".

VBA Syntax

```
' This code will automatically add repeat load using 'ACI:318-2002 'Table1' category with Start
       Load Case ID 10.

  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 rValue As Integer
      Dim loadList(1) As Integer
      Dim startLoadCaseNumber As Long
      Dim nCountGeneratedLCS As Long
      loadList(0) = 1
      loadList(1) = 2
      startLoadCaseNumber = 10
      Dim bRef As Boolean
      Dim bNotional As Boolean
      Dim bGB As Boolean

      bRef = True
      bNotional = True
      bGB = False
      Dim dNotionalLoadFactor As Double
      dNotionalLoadFactor = 0.002

      rValue = objOpenStaad.Load.AddAutoCombinationRepeat("ACI:318-2002","Table1", loadList,
        startLoadCaseNumber,  nCountGeneratedLCS, bRef, bNotional, dNotionalLoadFactor, bGB, 0,
        True, True, True, False)
      'If last primary/combination load case ID in the model is 5, then after successful
        execution of the API, the Repeat load IDs will start from 10 (startLoadCaseID = 10)
      'Else if last primary load case ID in the model is 15, then after successful execution of
        the API, the Repeat load IDs will start from 16 (startLoadCaseID = 16).

      MsgBox"Generated "& nCountGeneratedLCS & " Repeat Loads"
      Set objOpenStaad = Nothing
  End Sub
```

If nStartLoadCaseNo is available (i.e. id greater than max. ID of primary/combination load cases), Automatic Load Cases would be generated starting from user-provided nStartLoadCaseNo. But if nStartLoadCaseNo is invalid/ID already present, then Automatic Load Cases would be created starting from the next available ID.

◆ AddAutoLoadCombinations()

```
VARIANT OSLoadUI::AddAutoLoadCombinations ( const VARIANT FAR & varCode,
                                            const VARIANT FAR & varCategory,
                                            const VARIANT FAR & varLoadList,
                                            VARIANT FAR &       varStartLoadCaseNo )
```

Automatically adds load combination based on assigned design code and Category.

Parameters

`[in]`	**varCode**	Load Combination Code string name (refer to "Codes.ini") (type - String).
`[in]`	**varCategory**	Load Combination Category string name (refer to corresponding rule ini file defined in "Codes.ini") (type - String).
`[in]`	**varLoadList**	Load case reference ID(s), Array of Load case numbers. If the array is either null or empty then all load cases in current model will be considered (type - Long/Integer).
`[out]`	**varStartLoadCaseNo**	(Combination) load case reference ID with which automatically load combination generation starts (type - Long/Integer). \t If nStartLoadCaseNo is valid, auto load combinations will be created from the provided ID. \t If nStartLoadCaseNo is invalid Load Case ID/already present Load Case ID, load combinations would automatically generated from next available Load Case ID and nStartLoadCaseNo will be returned/updated with this ID.

Return values

0	if successful.
-8002	load case not found
-8040	invalid load combination code name
-8041	invalid load combination category name

> **Remarks**
> The default path of Codes.ini under "%localappdata%\Bentley\Engineering\STAAD.Pro CONNECT Edition\Default\Language\en".

C# Syntax

```
// This code will automatically add load combination using 'AISC 9th Ed' code '2.3 LRFD General'
       category with Start Load Case ID 10.
string strCode = "AISC 9th Ed";
string strCategory = "2.3 LRFD General";
int[] nPrimaryLoadCaseArray = { 1, 2 };
object objPrimaryLoadCaseArray = nPrimaryLoadCaseArray as object;
int nStartLoadCaseNo = 10;
int rValue = OSLoadUI.AddAutoLoadCombinations(strCode, strCategory, objPrimaryLoadCaseArray, ref
       nStartLoadCaseNo);
//If last primary/combination load case ID in the model is 5, then after successful execution of
       the API, the load combination IDs will start from 10 (startLoadCaseID = 10)
//Else if last primary/combination load case ID in the model is 15, then after successful
       execution of the API, the load combination IDs will start from 16 (startLoadCaseID = 16).
```

VBA Syntax

```
' This code will automatically add load combination using 'AISC 9th Ed' code '2.3 LRFD General' category with Start Load Case ID 10.
Dim rValue As Integer
Dim loadList(1) As Integer
Dim startLoadCaseNumber As Long
loadList(0) = 1
loadList(1) = 2
startLoadCaseNumber = 10
rValue = objOpenStaad.Load.AddAutoLoadCombinations("AISC 9th Ed","2.3 LRFD General", loadList, startLoadCaseNumber)
'If last primary/combination load case ID in the model is 5, then after successful execution of the API, the load combination IDs will start from 10 (startLoadCaseID = 10)
'Else if last primary/combination load case ID in the model is 15, then after successful execution of the API, the load combination IDs will start from 16 (startLoadCaseID = 16).
```

◆ AddLoadAndFactorToCombination()

```
VARIANT OSLoadUI::AddLoadAndFactorToCombination ( const VARIANT FAR & varLoadCombNo,
                                                  const VARIANT FAR & varLoadNo,
                                                  const VARIANT FAR & varFactor )
```

Adds a primary load case with specified multiplication factor to an existing load combination.

Parameters

`[in]` **varLoadCombNo**	(Combination) Load case reference number ID.
`[in]` **varLoadNo**	(Primary) Load case reference number ID.
`[in]` **varFactor**	Multiplication factor for the specified primary load case.

Return values

0	OK.
-1	General error.

C++ Syntax

```
// Sets primary load case #4 to load combination #5 with multiplication factor of 1.0.
VARIANT RetVal = OSLoadUI::AddLoadAndFactorToCombination(5, 4, 1.0);
```

.

VBA Syntax

```
' Sets primary load case #4 to load combination #5 with multiplication factor of 1.0.
Dim RetVal As VARIANT = OSLoadUI.AddLoadAndFactorToCombination(5, 4, 1.0)
```

See also

OSLoadUI::CreateNewLoadCombination

OSLoadUI::GetLoadCombinationType

OSLoadUI::SetLoadCombinationType

OSLoadUI::GetNoLCandFactorPairsForComb

OSLoadUI::GetLCandFactorForCombination

◆ CreateNewLoadCombination()

VARIANT OSLoadUI::CreateNewLoadCombination (const VARIANT FAR & varLoadCombTitle,
const VARIANT FAR & varLoadCombNo)

Creates new load combination with the number and title defined.

Parameters

[in]	**varLoadCombTitle**	Load case string title.
[in]	**varLoadCombNo**	Load case reference number ID.

Return values

<Val>	Number ID, assigned to new load combination.
-1	General error.

C++ Syntax

```
// Create a load combination, assigned with #4 named "COMBINATION LOAD CASE 4"
VARIANT nLoadComboNo = OSLoadUI::CreateNewLoadCombination((LPCTSTR)"COMBINATION LOAD CASE 4", 4);
```

.

VBA Syntax

```
' Create a load combination, assigned with #4 named "COMBINATION LOAD CASE 4"
Dim nLoadComboNo As VARIANT = OSLoadUI.CreateNewLoadCombination((LPCTSTR)"COMBINATION LOAD CASE
       4", 4)
```

See also

OSLoadUI::GetLoadCombinationType

OSLoadUI::SetLoadCombinationType

OSLoadUI::AddLoadAndFactorToCombination

◆ GetLoadAndFactorForCombination()

```
VARIANT OSLoadUI::GetLoadAndFactorForCombination ( const VARIANT FAR & varLoadCombNo,
                                                   VARIANT FAR &       varLoadNos,
                                                   VARIANT FAR &       varFactors )
```

Gets load case reference number ID(s) and corresponding multiplication factor(s) for specified load combination.

Parameters

`[in]` **varLoadCombNo** (Combination) Load case reference number ID.

`[out]` **varLoadNos** (Primary) Load case reference number ID(s) VARIANT array.

`[out]` **varFactors** Multiplication factor(s) VARIANT array. For SRSS allocate size of this array as no. of primary cases + 1. The last value will return the overall multiplication factor for SRSS combination.

Return values

True if the method is successful.

False if the method is unsuccessful.

C++ Syntax

```
// Gets the load combination information for load combination #4
VARIANT RetVal = OSLoadUI::GetLoadAndFactorForCombination(4, &varLoadNos, &varFactors);
```

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 objLoadCombId As Long
      Dim objPrimaryLCArray(2) As Long
      Dim objFactorsArray(3) As Double
      Dim msgString As String

      Dim rValue As Boolean
      Dim i As Integer

      objLoadCombId = 201

      rValue =
        objOpenStaad.Load.GetLoadAndFactorForCombination(objLoadCombId,objPrimaryLCArray,objFactorsArr

      For i =0 To 2
          MsgBox(objPrimaryLCArray(i)& vbCr)
      Next

      ' The last value will return the overall multiplication factor for SRSS combination
      For i =0 To 3
          MsgBox(objFactorsArray(i)& vbCr)
      Next

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

See also

OSLoadUI::AddLoadAndFactorToCombination

OSLoadUI::GetNoOfLoadAndFactorPairsForCombination

◆ GetLoadCombinationCaseCount()

VARIANT OSLoadUI::GetLoadCombinationCaseCount ( )

Gets total number of combination load case(s) present in the current structure.

Returns

The total number of combination load cases(s).

C++ Syntax

```
// Count the combination load case(s).
VARIANT nLoadCombinationCase = OSLoadUI::GetLoadCombinationCaseCount();
```

VBA Syntax

```
' Count the combination load case(s).
Dim nLoadCombinationCase As VARIANT = OSLoadUI.GetLoadCombinationCaseCount()
```

See also

OSLoadUI::GetLoadCombinationCaseNumbers

◆ GetLoadCombinationCaseNumbers()

VARIANT OSLoadUI::GetLoadCombinationCaseNumbers (VARIANT FAR & lCases)

Gets all load combination case number(s).

Parameters

`[out]` **lCases** The load case reference number ID(s) VARIANT array.

Return values

<Val>	The number of primary load case(s).
-1	General error.
-106	1 dimensional array of long expected.
-114	OLE Exception Occurred.

C++ Syntax

```
// Get load combination case(s) refs.
VARIANT nLoadCombinationCase = OSLoadUI::GetLoadCombinationCaseNumbers(&lCases);
```

VBA Syntax

```
' Get load combination case(s) refs.
Dim nLoadCombinationCase As VARIANT = OSLoadUI.GetLoadCombinationCaseNumbers(&lCases)
```

See also

OSLoadUI::GetLoadCombinationCaseCount

◆ GetNoOfLoadAndFactorPairsForCombination()

VARIANT OSLoadUI::GetNoOfLoadAndFactorPairsForCombination (const VARIANT FAR & varLoadCombNo)

Gets the number of load case(s) applied with multiplication factor in specified load combination.

Parameters

[in] **varLoadCombNo** (Combination) Load case reference number ID.

Returns

The number of load case(s) in specified load combination.

C++ Syntax

```
// Gets the number of load case(s) in load combination #4
VARIANT nLoadCaseInComb = OSLoadUI::GetNoOfLoadAndFactorPairsForCombination(4);
```

VBA Syntax

```
' Gets the number of load case(s) in load combination #4
Dim nLoadCaseInComb As VARIANT = OSLoadUI.GetNoOfLoadAndFactorPairsForCombination(4)
```

See also

OSLoadUI::AddLoadAndFactorToCombination

OSLoadUI::GetLoadAndFactorForCombination

◆ IsCombinationCase()

VARIANT OSLoadUI::IsCombinationCase (long nLoadCase)

Checks if specified load case is combination load case.

Parameters

`[in]` **nLoadCase** Load case reference ID.

Return values

1 YES.

0 NO.

-1 General error.

C++ Syntax

```
// Check if load case #1 is load combination load case.
VARIANT IsCombCase =  OSLoadUI::IsCombinationCase(1);
```

VBA Syntax

```
' Check if load case #1 is load combination load case.
Dim IsCombCase As VARIANT = OSLoadUI.IsCombinationCase(1)
```