# Definitions

(Terms that may be unfamiliar to the reader.)

|  |  |
| --- | --- |
| API | Application program interface |
| IN | Input parameter. |
| OUT | Output parameter |

# Design

How to use the sample code.

SETUP INSTRUCTIONS // FRAMEWORK DEVELOPMENT

1. Create a new Visual Basic application of the type “Windows Forms Application”

2.   It is assumed that the action trigger is a button on the basic form. Else, the framework

            Code may be placed appropriately in the action handler of the custom control.

            In the space between the “Private Sub Command1\_Click ()” and “End Sub”,

            Copy / paste, the FRAMEWORK CODE provided below.

**FRAMEWORK CODE:**

On Error Resume Next

Dim objStructureEditorApp As Object

Dim objStructureEditorSEEC As Object

Dim Cleanup As Boolean

objStructureEditorApp = Nothing

objStructureEditorSEEC = Nothing

Cleanup = False

' If the application is not nothing, we are assuming StructureEditor is already invoked

objStructureEditorApp = GetObject(, "StructureEditor.Application")

If objStructureEditorApp Is Nothing Then

MsgBox("Could not locate an already running instance of Structure editor application, Creating a new instance")

If Err.Number <> 0 Then

objStructureEditorApp = CreateObject("StructureEditor.Application")

objStructureEditorApp.Visible = True

Cleanup = True

Err.Clear()

End If

End If

objStructureEditorSEEC = objStructureEditorApp.SEECStructureEditor

**'###############################################**

**'Add API Specific Code Here**

**'###############################################**

If Cleanup = True Then

'kill the instance of structure editor created

objStructureEditorApp.Quit()

Cleanup = False

End If

'Reset the object definition

objStructureEditorSEEC = Nothing

objStructureEditorApp = Nothing

1. Now copy the sample code provided for the API of your interest from the help document.

Paste it in the section “**Add API Specific Code Here**” provided in the framework.

# Sample Code

Since each API has sample code associated with it, the code samples are placed alongside the API details.

## Structure editor Automation APIs

### ValidateLogin

**API Name: ValidateLogin**

Purpose: This API is used to login to teamcenter server.

Syntax: Call objStructureEditorSEEC.ValidateLogin( UserName, Password, Group, Role, DBURL)

Parameters:

* UserName:-

Data Type: String

Type: IN

Purpose: Specifies the UserName of the user to authenticate.

* Password:-

Data Type: String

Type: IN

Purpose: Specifies the password of the user to authenticate

* Group:-

Data Type: String

Type: IN

Purpose: Specifies the group to which the user belongs. The default group is “dba”.

* Role:-.

Data Type: String

Type: IN

Purpose: Specifies the Role of the User in teamcenter.

* DBURL:-

Data Type: String

Type: IN

Purpose: Specifies the server name to which the user wants to connect

***Sample Code:***

|  |
| --- |
| Dim UserName As String  Dim Password As String  Dim Group As String  Dim Role As String  Dim URL As String  UserName = "Username"  Password = ""  Group = ""  Role = ""  URL = "http://Servername:8085/tc"  Call objStructureEditorSEEC.ValidateLogin(UserName,Password,Group,Role,URL) |

### Open

**API Name: Open**

Purpose: This API is used to open a solid edge file from the teamcenter server into structure editor.

Syntax: objStructureEditorSEEC.Open(ItemId, ItemRev, FileName, RevisionRule, FolderName)

Parameters:

* ItemId:-.

Data Type: String

Type: IN

Purpose: Specifies the item id of the document to be opened

* ItemRev:-

Data Type: String

Type: IN

Purpose: Specifies the revision of the document to be opened

* FileName:-

Data Type: String

Type: IN

Purpose: Specifies the name of the document to be opened.

* RevisionRule:-

Data Type: String

Type: IN

Purpose: Specifies the revision rule to be applied on the document.

* FolderName(Input): This parameter should be empty or NULL

Data Type: String

Type: IN

Purpose: This parameter should be NULL.

***Sample Code:***

|  |
| --- |
| Dim ItemId As String  Dim ItemRev As String  Dim FileName As String  Dim RevisionRule As String  Dim FolderName As String  ItemId = "081691"  ItemRev = "A"  FileName = "a.asm"  RevisionRule = "Latest Working"  FolderName = ""    Call objStructureEditorSEEC.Open(ItemId,ItemRev,FileName,RevisionRule,FolderName) |

### SetSaveAsAll

**API Name: SetSaveAsAll**

Purpose: This API is used to set save as action on all the open files in the structure editor.

Syntax: objStructureEditorSEEC.SetSaveAsAll()

Parameters: None.

***Note:***

*Call this API after calling open () API. Because for calling this API a solid edge file needs to be opened in the structure editor application. If the action is set without opening a solid edge file in the application then nothing will happen.*

***Sample Code:***

|  |
| --- |
| If (objStructureEditorSEEC.Open(ItemId, ItemRev, FileName, RevisionRule, FolderName) = 0) Then  Call objStructureEditorSEEC.SetSaveAsAll()  End If |

### SetSaveAsSelected

**API Name: SetSaveAsSelected**

Purpose: This API is used to set save as action on a selected solid edge file in the structure editor.

Syntax: objStructureEditorSEEC.SetSaveAsSelected (ItemId, ItemRev, FileName)

Parameters:

* ItemId:-

Data Type: String

Type: IN

Purpose: Specifies the item id of the document on which the save as action is to be set.

* ItemRev:-

Data Type: String

Type: IN

Purpose: Specifies the revision of the document on which the save as action is to be set.

* FileName:-

Data Type: String

Type: IN

Purpose: Specifies the Name of the file on which the save as action is to be set.

**Note:**

*Call this API after calling open () API. Because for calling this API a solid edge file needs to be opened in the structure editor application*. *If the action is set without opening a solid edge file in the application then nothing will happen.*

*If user wants to select multiple files for save as then the above code needs to be called multiple times.*

***Sample Code:***

|  |
| --- |
| ItemId = "081691"  ItemRev = "A"  FileName = "a.asm"  If (objStructureEditorSEEC.Open(ItemId, ItemRev, FileName, RevisionRule, FolderName) = 0) Then  Call objStructureEditorSEEC.SetSaveAsSelected(ItemId, ItemRev, FileName)  End If |

### SetReviseAll

**API Name: SetReviseAll**

Purpose: This API is used to set revise action on all open solid edge files in structure editor.

Syntax: objStructureEditorSEEC.SetSaveAsAll()

Parameters: None.

**Note:**

*Call this API after calling open() API. Because for calling this API a solid edge file needs to be opened in the structure editor application. If the action is set without opening a solid edge file in the application then nothing will happen.*

*.*

***Sample Code:***

|  |
| --- |
| If (objStructureEditorSEEC.Open(ItemId, ItemRev, FileName, RevisionRule, FolderName) = 0) Then  Call objStructureEditorSEEC.SetReviseAll()  End If |

### SetReviseSelected

**API Name: SetReviseSelected**

Purpose: This API is used to set revise action on a selected file in the structure editor.

Syntax: objStructureEditorSEEC.SetReviseSelected(ItemId, ItemRev, FileName)

Parametes:

* ItemId:-

Data Type: String

Type: IN

Purpose: Specifies the item id of the document on which the save as action is to be set.

* ItemRev:-

Data Type: String

Type: IN

Purpose: Specifies the revision of the document on which the save as action is to be set.

* FileName:-

Data Type: String

Type: IN

Purpose: Specifies the Name of the file on which the save as action is to be set.

**Note:**

*Call this API after calling open() API. Because for calling this API a solid edge file needs to be opened in the structure editor application. If the action is set without opening a solid edge file in the application then nothing will happen.*

*If user wants to select multiple files for revise then the above code needs to be called multiple times.*

***Sample Code:***

|  |
| --- |
| ItemId = "081691"  ItemRev = "A"  FileName = "a.asm"  If (objStructureEditorSEEC.Open(ItemId, ItemRev, FileName, RevisionRule, FolderName) = 0) Then  Call objStructureEditorSEEC.SetReviseSelected(ItemId, ItemRev, FileName)  End If |

### AssignAll

**API Name: AssignAll**

Purpose: This API is used to automatically assign system generated values to the mandatory properties of a document like item id,item revision,item name and dataset name.

Syntax: Call objStructureEditorSEEC.AssignAll()

Parameters: None.

**Note:**

*Call this API after setting either save as or revise action on the file. Because before calling this API action needs to be set on the file. If no action is set on the solid edge file and AssignAll is called then values will not be assigned as there is no file selected and no action is set on the file.*

***There are two cases when this API can be used.***

***Case1 Save as****: If this API is called after save as action is selected then values will be assigned to Item id, item revision, item name and dataset name.*

***Case2 Revise****: If this API is called after revise action is selected then value will be assigned to Revision and dataset name.*

***Sample Code:***

This sample code uses SetReviseSelected API for illustration purpose; user can use any of the SetReviseAll, SetSaveAsAll or SetSaveAsSelected APIs instead.

|  |
| --- |
| If (objStructureEditorSEEC.SetReviseSelected(ItemId, ItemRev, FileName) = 0) Then  Call objStructureEditorSEEC.AssignAll()  End If |

### PerformActions

**API Name: PerformActions**

Purpose: This API is used to perform actions set on the solid edge file.

Syntax: Call objStructureEditorSEEC.PerformActions()

Parameters: None.

**Note:**

*Call this API after setting action on the solid edge file. Do not forget to assign values after setting the action. If the values are not set on the mandatory parameters of the solid edge file then the action will not be performed.*

***Sample Code:***

|  |
| --- |
| If (objStructureEditorSEEC.AssignAll() = 0) Then  Call objStructureEditorSEEC.PerformActions()  End If |

### ClearAllActions

**API Name: ClearAllActions**

Purpose: This API is used to clear the action, if any set on the solid edge file.

Syntax: Call objStructureEditorSEEC.PerformActions()

Parameters: None.

***Sample Code:***

|  |
| --- |
| Call objStructureEditorSEEC.ClearAllActions() |

### Close

**API Name: Close**

Purpose: This API is used to close the solid edge file.

Syntax: Call objStructureEditorSEEC.close()

Parameters: None.

***Sample Code:***

|  |
| --- |
| Call objStructureEditorSEEC.close() |

### SetDisplayAlerts

**API Name: SetDisplayAlerts**

Purpose: This API is used to set the display alerts property ON or OFF.

If this property is ON then certain alerts and messages will be displayed while the application is running.

If this property is OFF then the operation will be performed silently without any alerts and messages.

Syntax: Call objStructureEditorApp.SetDisplayAlerts(True)

Parameters:

* bDisplayAlerts :-

Data Type:- Boolean

Type: IN

Purpose: To set display alerts property ON or OFF.

If TRUE: Set the display alerts ON

If FALSE: Set the display alerts OFF.

***Sample Code:***

|  |
| --- |
| objStructureEditorApp.SetDisplayAlerts(True) |

### GetDisplayAlerts

**API Name: GetDisplayAlerts**

Purpose: This API is used to get the status of display alerts property whether it is ON or OFF.

Syntax: Call objStructureEditorApp.GetDisplayAlerts(bDisplayAlerts)

Parameters:

* bDisplayAlerts :-

Data Type:- Boolean

Type: OUT

Purpose: bDisplayAlerts gets the status of the display alerts property.

If TRUE: Display alerts property is ON.

If FALSE: Display alerts property is OFF.

***Sample Code:***

|  |
| --- |
| Dim bDisplayAlerts As Boolean = False  objStructureEditorApp.GetDisplayAlerts(bDisplayAlerts) |

### Quit

**API Name: Quit**

Purpose: This API is used to quit the structure editor application.

Syntax: objStructureEditorApp.Quit ()

Parameters: None.

***Sample Code:***

|  |
| --- |
| objStructureEditorApp.Quit() |

### GetProjectsForLoggedInUSer

**API Name: GetProjectsForLoggedInUser**

Purpose: This API is used to get the projects for logged in User.

Syntax: objStructureEditorSEEC.GetProjectsForLoggedInUser (LisOfProject)

Parameters:

* LisOfProject :-
  + Data Type – Variant Array.
  + Type – OUT.
  + Purpose – To get the projects for logged in user.

***Sample Code:***

|  |
| --- |
| Dim LisOfProject As Object = Nothing  objStructureEditorSEEC.GetProjectsForLoggedInUser(LisOfProject) |

### GetProjectsForGivenItemID

**API Name: GetProjectsForLoggedInUser**

Purpose: This API is used to get the projects for GivenItemID.

Syntax: objStructureEditorSEEC.GetProjectsForGivenItemID (ItemId, ListOfProjects)

Parameters:

* Item Id :-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Item Id of which the projects to be returned.
* LisOfProject :-
  + Data Type – Variant Array.
  + Type – OUT.
  + Purpose – To get the projects for given item ID.

***Sample Code:***

|  |
| --- |
| Dim ItemId As String  ItemId = "002262"  Dim LisOfProject As Object = Nothing  objStructureEditorSEEC.GetProjectsForGivenItemID(ItemId,LisOfProject) |

### AddItemToGivenProjects

**API Name: AddItemToGivenProjects**

Purpose: This API is used to add the Item to given projects.

Syntax: objStructureEditorSEEC.AddItemToGivenProjects(ItemId, ItemRev, FileName, Projects)

Parameters:

* Item Id :-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Item Id which is to add in given projects.
* Item Rev:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the revision of Item of which is to add in given projects
* File Name:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the File name of the item of which is to add in given projects
* Projects:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Projects in which the given item to be added.

.

***Sample Code:***

|  |
| --- |
| Dim ItemId As String  Dim ItemRev As String  Dim FileName As  Dim Projects As String  ItemId = "002262"  ItemRev = "A"  FileName = "002262.par"  Projects = "1987;1996;2000"  objStructureEditorSEEC.AddItemToGivenProjects(ItemId, ItemRev, FileName, Projects) |

Note – This API will take the multiple projects as Input, which are separated by Semi comma. To Add the Item to the projects with the help of this API, user has to first open the given Item Id, Set save As selected, Assign All and then this API should called. And finally perform action should gets called.

When we do Save As selected on given Item by default projects of given item are assigned to new item. With the help of this API user will get privileged to add the new item in the projects he wants and not the default ones of parent item.

### SetReplaceSelected

**API Name: SetReplaceSelected**

Purpose: This API is used to replace the existing item with new item.

Syntax: objStructureEditorSEEC. SetReplaceSelected (srcItemId, srcItemRev, srcFileName,

TgtItemId, TgtItemRev, TgtFileName, RevisionRule, PrntItemId,

PrntItemRev, PrntFileName, FOAMember, TgtFOAMember)

Parameters:

* srcItemId:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Item Id of the item which is to be replaced.
* srcItemRev:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Revision of the item which is to be replaced.
* srcFileName:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the filename of the item which is to be replaced.
* TgtItemId:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Item Id of the item by which item will be replaced.
* TgtItemRev:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Revision f the item by which item will be replaced.
* TgtFileName:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the filename of the item by which item will be replaced.
* RevisionRule:-

Data Type: String

Type: IN

Purpose: Specifies the revision rule to be applied on the document.

* PrntItemId:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Item ID of the parent from which item will be replaced
* PrntItemRev:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Revision of the parent from which item will be replaced
* PrntFileName:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the filename of the parent from which item will be replaced
* FOAMember:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the FOA member name of the file which is to be replaced
* TgtFOAMember:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the FOA member name of the file by which item will be replaced
* ***Sample Code:***

|  |
| --- |
| Dim srcItemId As String  Dim srcItemRev As String  Dim srcFileName As String  Dim TgtItemId As String  Dim TgtItemRev As String  Dim TgtFileName As String  Dim RevisionRule As String  Dim PrntItemId As String  Dim PrntItemRev As String  Dim PrntFileName As String  Dim FOAMember As String  Dim TgtFOAMember As String  srcItemId = "219350"  srcItemRev = "A"  srcFileName = "219350.par"  PrntItemId = "219351"  PrntItemRev = "A"  PrntFileName = "219351.asm"  FOAMember = Nothing  TgtFOAMember = Nothing  TgtFileName = "222777.par"  TgtItemId = "222777"  TgtItemRev = "A"  objStructureEditorSEEC.SetReplaceSelected(srcItemId, srcItemRev, srcFileName,  TgtItemId, TgtItemRev, TgtFileName,  RevisionRule, PrntItemId, PrntItemRev,  PrntFileName, FOAMember, TgtFOAMember) |

Note – Call this API after calling open () API. Because for calling this API a solid edge file needs to be opended in the structure editor application. If the action is set without opening a solid edge file in the application then nothing will happen.

If user wants to replace multiple files then the above code needs to be called multiple times.

### Add existing item into an Assembly or empty item

**API Name: AddExistingItem**

Purpose: This API is used to add the given Item into specified item.

Syntax: objStructureEditorSEEC.AddExistingItem(ItemId, ItemRev, FileName,

ItemIdToBeAdded, ItemRevToBeAdded, FileNameToBeAdded, RevisionRule)

Parameters:

* ItemId :-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Item Id into which the item will be added.
* ItemRev :-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the revision of Item into which the item will be added.
* File Name:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the File name of the item into which the item will be added.
* ItemIdToBeAdded :-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Item Id of the item to be added.
* ItemRevToBeAdded:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Item Revision of the item to be added.
* FileNameToBeAdded:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the File name of the item to be added.
* RevisionRule:-

Data Type: String

Type: IN

* + Purpose: Specifies the revision rule used for opening the document.

***Sample Code:***

|  |
| --- |
| Dim ItemId As String  Dim ItemRev As String  Dim FileName As String  Dim ItemIdToBeAdded As String  Dim ItemRevToBeAdded As String  Dim FileNameToBeAdded As String  Dim RevisionRule As String  ItemId = " 012186"  ItemRev = "A"  FileName = " 012186.asm"  FileNameToBeAdded = "011854.par"  ItemIdToBeAdded = "011854" ItemRevToBeAdded = "A"  RevisionRule = "Latest Working"  objStructureEditorSEEC.AddExistingItem(ItemId, ItemRev, FileName,  ItemIdToBeAdded, ItemRevToBeAdded,  FileNameToBeAdded, RevisionRule) |

Note – Call this API after calling open () API. Because for calling this API a solid edge file or an empty item into which the item will be added needs to be opened in the structure editor application. If the action is set without opening a solid edge file in the application then nothing will happen.

If user wants to add multiple files then the above code needs to be called multiple times.

### Add new item into an Assembly or empty item

**API Name: AddNewItem**

Purpose: This API is used to add the new empty item into specified item.

Syntax: objStructureEditorSEEC.AddNewItem(ItemId, ItemRev, FileName)

Parameters:

* ItemId :-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Item Id into which the new item will be created.
* ItemRev :-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the revision of Item into which the item will be created.
* File Name:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the File name of the item into which the item will be created.

***Sample Code:***

|  |
| --- |
| Dim ItemId As String  Dim ItemRev As String  Dim FileName As String  ItemId = “ 012186”  ItemRev = "A"  FileName = " 012186.asm"  objStructureEditorSEEC.AddNewItem(ItemId, ItemRev, FileName) |

Note – Call this API after calling open () API. Because for calling this API a solid edge file or an empty item into which the new empty item will be created needs to be opened in the structure editor application. If the action is set without opening a solid edge file in the application then nothing will happen.

If user wants to create multiple new items then the above code needs to be called multiple times.

### Create root new item

#### API Name: CreateRootNewItem

Purpose: This API is used to create root new empty item.

Syntax: objStructureEditorSEEC.CreateRootNewItem()

Parameters: None

***Sample Code:***

|  |
| --- |
| Call objStructureEditorSEEC. CreateRootNewItem() |

Note – In order to create new root item there should not be any file open in the Structure Editor application.

### Remove item from an Assembly or empty item

**API Name: RemoveItem**

Purpose: This API is used to remove the given Item from specified item.

Syntax: objStructureEditorSEEC.RemoveItem (PrntItemId, PrntItemRev, PrntFileName,

ItemIdToBeRemoved, ItemRevToBeRemoved, FileNameToBeRemoved)

Parameters:

* PrntItemId :-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the parent Item Id from which the item will be removed.
* PrntItemRev :-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the parent Item revision from which the item will be removed.
* PrntFile Name:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the parent File name of the item from which the item will be removed.
* ItemIdToBeRemoved:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Item Id of the item to be removed.
* ItemRevToBeRemoved:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the Item Revision of the item to be removed.
* FileNameToBeRemoved:-
  + Data Type – String.
  + Type – In
  + Purpose – Specifies the File name of the item to be removed.

***Sample Code:***

|  |
| --- |
| Dim PrntItemId As String  Dim PrntItemRev As String  Dim PrntFileName As String  Dim ItemIdToBeRemoved As String  Dim ItemRevToBeRemoved As String  Dim FileNameToBeRemoved As String  PrntItemId = " 012186"  PrntItemRev = "A"  PrntFileName = " 012186.asm"  FileNameToBeRemoved = "012213.par"  ItemIdToBeRemoved = "012213"  ItemRevToBeRemoved = "A"  objStructureEditorSEEC.RemoveItem(PrntItemId, PrntItemRev, PrntFileName,  ItemIdToBeRemoved, ItemRevToBeRemoved,  FileNameToBeRemoved) |

Note – Call this API after calling open () API. Because for calling this API a solid edge file or an empty item from which the item will be removed needs to be opened in the structure editor application. If the action is set without opening the parent solid edge file in the application then nothing will happen.

If user wants to remove different items from the parent item then the above code needs to be called multiple times.

If parent contains multiple occurrences of the same item then setting remove action on single item will set remove action on all the occurrences within parent.