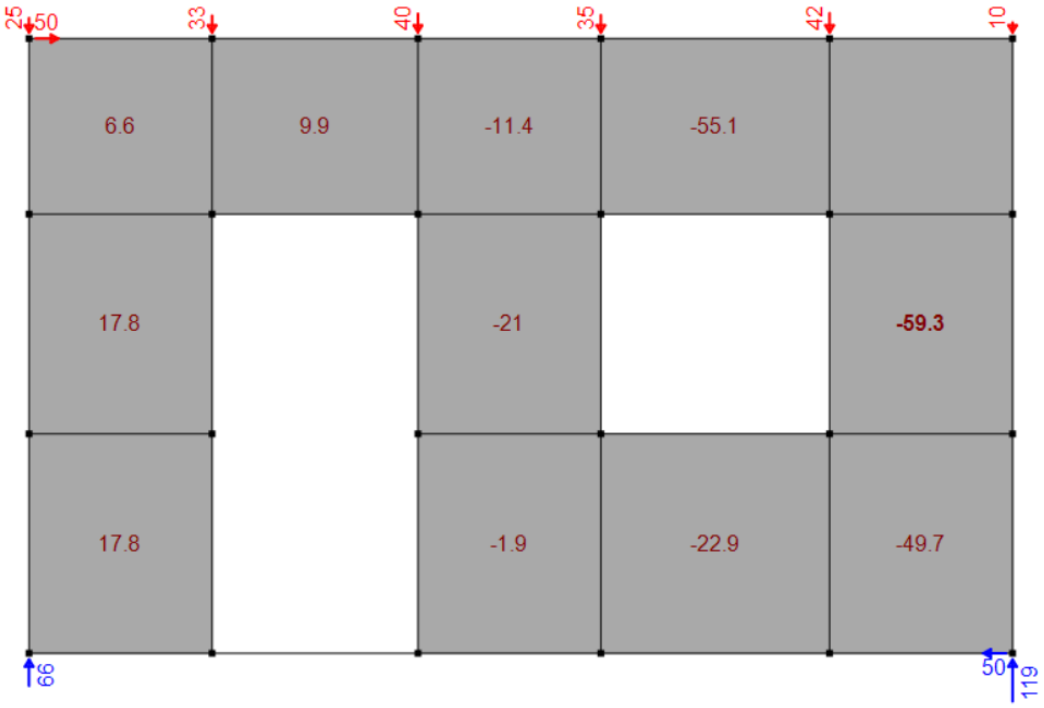
PolyStringer

An Optimal Plastic Solution

PolyStringer API – Documentation



Introduction

The PolyStringer API is an application program interface which allows the user to input geometry, openings, stringers, loads, reactions and shear field limits in PolyStringer programmatically from for example an Excel spreadsheet. Furthermore, it is possible to extract calculation results such as stringer forces, reactions and shear forces. These results can then easily be processed in a custom excel spreadsheet.

The PolyStringer API is an ideal tool to automate custom processes in relation to stringermodel calculations. Both to generate stringermodels and to post process results, such as verify concrete stresses or use reactions for further calculations.

In this short guide the possibilities with PolyStringer API is described and a detailed description on how to get started is provided.

Content

Integrations

It is possible to integrate PolyStringer with your self-developed load path Excel sheet to automatically transfer loads, reactions and geometry to PolyStringer through PolyStringer API. The user can then edit the stringermodel by adding openings, remove stringers etc. in PolyStringer.

An Excel template can be found at www.PolyStringer.com/PolyStringerAPI. With this template it is easy to get started with PolyStringer API. It is possible to open a PolyStringer API file (.pstrAPI) and view both input and calculation results. Furthermore, you can create a complete PolyStringer model with all geometry and loads defined and open the model in PolyStringer to get a graphical view, do the calculation and export the results in a pdf.

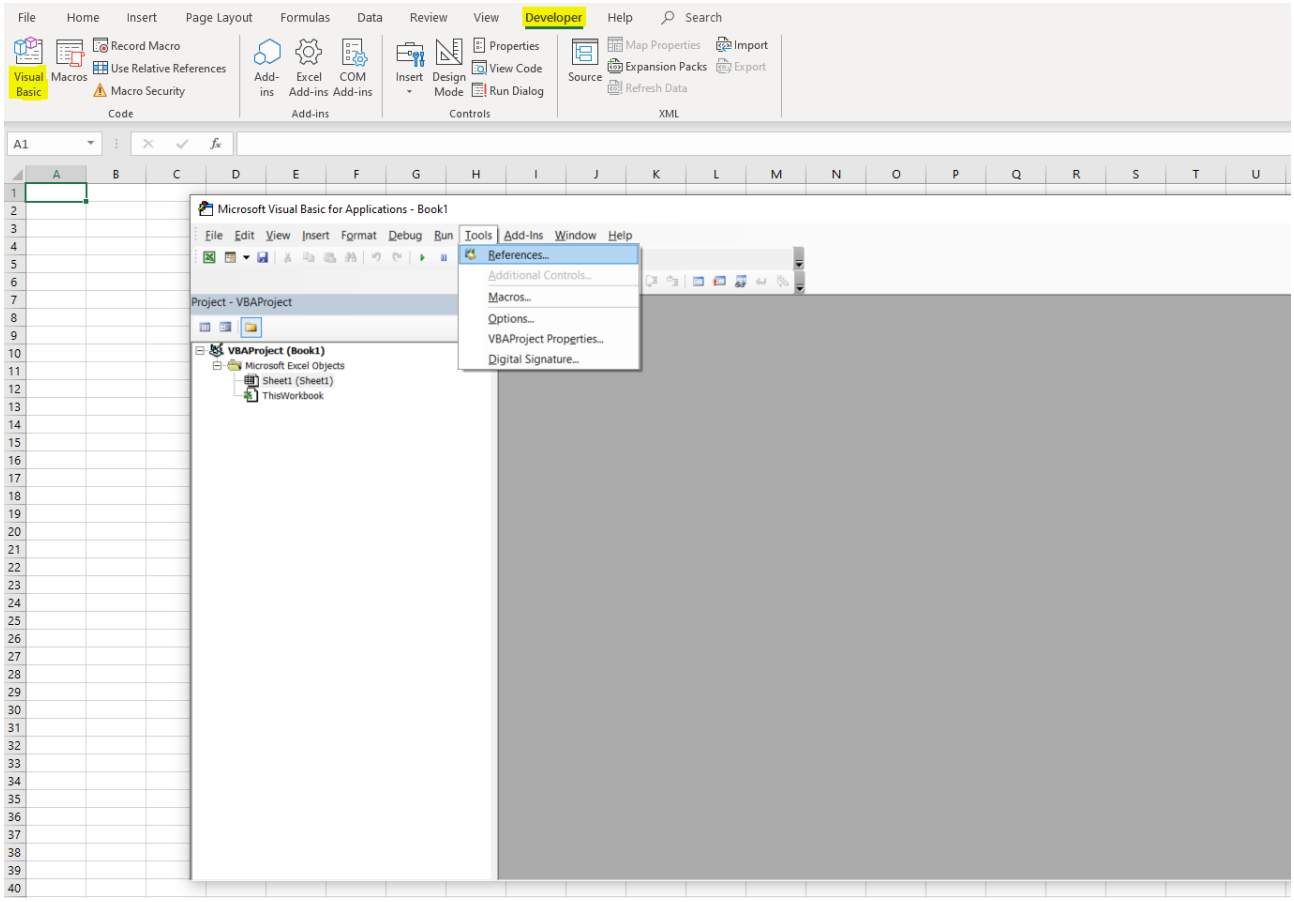
How to Start

1. Download and install PolyStringer from www.PolyStringer.com/Download
2. Make sure you have a PolyStringer API license or contact us at Contact@PolyStringer.com to inquire about license options.

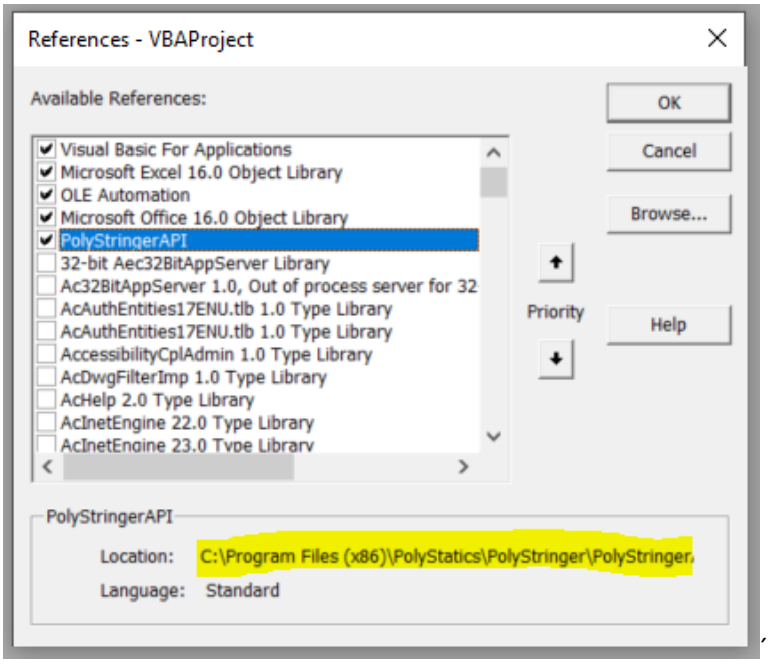
Setup PolyStringer API for Excel VBA

1. Setup Excel developer tab: file -> options -> Customize Ribbon -> Make sure “developer” is checked.
2. Reference PolyStringer API in Excel:

Open an Excel spreadsheet, go to the “developer” tab click on “Visual Basic” in the “Microsoft Visual Basic For Applications” form that appear, go to tools -> references



Find PolyStringerAPI in the list of "Available References" check the tick box and click Ok.



3. Check your Excel installation: file -> Account -> About Excel -> is your Excel 32bit or 64bit.
    a. For 32bit Excel go to step 4
    b. For 64bit Excel: Open the folder containing PolyStringerAPI.dll (path can be found in references from step 2 – see with yellow) -> find register.bat, right click and edit -> change the path to PolyStringerAPI.dll if needed -> save the changes to the file and close it -> run register.bat as administrator.
4. Create PolyStringer model using PolyStringer API

PolyStringer API is now registered for your Excel application and you can get started with your own implementations. A template and multiple examples can be found on www.PolyStringer.com/PolyStringerAPI these documents will both show you the opportunities with PolyStringerAPI and help you get started developing your own custom Excel spreadsheets.

Class: PolyStringerModel

This class is the main class in PolyStringer API, it includes all methods needed to create a PolyStringer model and open PolyStringer. All relevant parameters are also included in this class both related to input and results.

All inputs to PolyStringer API are SI-Units ([m], [N], [N/m] etc.)

Input Parameters

xCoordinatesArr

An array with all added xCoordinates and all automatically generated xCoordinates (Loads, supports, stringers and openings coordinates).

yCoordinatesArr

An array with all added yCoordinates and all automatically generated yCoordinates (Loads, supports, stringers and openings coordinates).

LoadCombinationsArr

An array with all loadcombinations. Includes both load combination name and an array with both node and stringer loads.

StringersArr

An array with all special stringers (Stringer type: Compression Only or Inactive). Includes both stringer type, start- and end coordinates.

OpeningsArr

An array with all openings. Include parameter removeInternalStringers (true/false) and corner coordinates.

NodeSupportsArr

An array with all node supports. Include a position, fixed directions, and a list of reactions if these are defined for some or all loadcombinations.

StringerSupportsArr

An array with all stringer supports. Include start-, end coordinate and a list of reactions if these are defined for some or all loadcombinations.

ShearFieldLimitGroupsArr

An array with all shear field limit groups. Includes a group number, a shearlimit and a list of shearfields.

APIMessages

A string which contains all information generated while using the PolyStringer API. The message might be relevant when creating the PolyStringer model through the PolyStringer API, since it will contain information about which properties are added/removed or changed.

Result Parameters

These parameters will be available when the PolyStringer model is calculated in PolyStringer and saved as a PolyStringer API file (.pstrAPI).

NodeReactionsArr

An array with all Node Reactions. Includes both position, reaction values, loadcombination name, and a string describing calculationResultType (Modified or initial solution)

StringerReactionsArr

An array with all Stringer Reactions. Includes both start and end coordinate, reaction value, loadcombination name, and a string describing calculationResultType (Modified or initial solution)

StringerForcesArr

An array with all stringer forces. Includes both start and end coordinate, start and end forces, loadcombination name, and a string describing calculationResultType (Modified or initial solution)

ShearFieldResultsArr

An array with all shearfield forces. Includes both corner coordinates, shearfield force, loadcombination name, and a string describing calculationResultType (Modified or initial solution)

Methods

AddXCoordinates(double x)

Adds a x-coordinate to the PolyStringer model.

Failure:

Occurs if the x-value is already added.

AddYCoordinates(double y)

Adds a y-coordinate to the PolyStringer model.

Failure:

Occurs if the y-value is already added.

AddLoadCombination(string loadCombinationName)

Adds a load combination to the PolyStringer model.

Failure:

Occurs if a loadcombination with the same name is already added. Or the input string loadcombination-Name is either null, empty or whitespace.

AddNodeLoad (string loadCombinationName, double xCoordinate, double yCoordinate, double fx, double fy, bool replaceLoad = false)

Adds a Node load to a loadcombination.

If replaceLoad is true then for a node load at the same position the load values are overwritten with the new fx and fy-values.

If replaceLoad is false then for a node load at the same position the load values are added together with the new fx and fy-values.

Failure:

Occurs if a loadcombination with loadCombinationName is not previously defined using the command Ad-dLoadCombination(string loadCombinationName).

AddStringerLoad (string loadCombinationName, double x1, double x2, double y1, double y2, double f)

Adds a stringer load to a loadcombination.

If the stringer is not vertical or horizontal (either x1 = x2 or y1 = y2) a NodeLoad is inserted at the center with corresponding fx and fy-values

If a stringer load at the same position is already defined for this loadcombination then the load values are overwritten with the new f-value and x/y-coordinates.

Failure:

Occurs if a loadcombination with loadCombinationName is not previously defined using the command Ad-dLoadCombination(string loadCombinationName).

AddCompressionOnlyStringer(double x1, double x2, double y1, double y2)

Adds a compression only stringer.

If a stringer with special parameters (Inactive or CompressionOnly) is already defined for this position, then the stringer type and x/y-coordinates is overwritten.

AddInactiveStringer(double x1, double x2, double y1, double y2)

Adds an inactive stringer.

If a stringer with special parameters (Inactive or CompressionOnly) is already defined for this position, then the stringer type and x/y-coordinates is overwritten.

AddOpening(bool removeInternalStringers, double x1, double x2, double y1, double y2)

Adds an opening in PolyStringer.

If removeInternalStringers are true, then all stringers within the opening are set inactive.

AddNodeSupport(bool fxFixed, bool fyFixed, double x, double y, string fxReaction = "", string fyReaction = "", string loadCombinationName = "")

Adds a Node support.

A reaction value can be defined by a number for either fxReaction or fyReaction and a valid loadCombinationName.

Failure:

Occurs if fxReaction or fyReaction is not a valid number -> Node support without defined reactions are inserted.

Occurs if fxFixed and fyFixed = false -> atleast one support direction must be fixed for the support to be active.

Occurs if a reaction is inserted in a direction that is not fixed.

Occurs if a reaction is inserted and loadcombination with loadCombinationName is not previously defined using the command AddLoadCombination(string loadCombinationName).

AddStringerSupport(double x1, double x2, double y1, double y2, string reaction = "", string loadCombinationName = "")

Adds a Stringer support.

A reaction value can be defined by a number for reaction and a valid loadCombinationName.

Failure:

Occurs if reaction is not a valid number -> stringer support without defined reaction is inserted.

Occurs if a reaction is inserted and loadcombination with loadCombinationName is not previously defined using the command AddLoadCombination(string loadCombinationName).

AddShearFieldLimitGroup(string shearLimit = "", int groupNr = 1)

Adds a shear field limit group.

If the shear field limit group number is already used then the shearLimit is changed.

Failure:

Occurs if the shearLimit is not passed in as a number. -> The group no. is created without a shear limit.

AddShearFieldToLimitGroup(double x1, double x2, double y1, double y2, int groupNo = 1)

Adds a shearField to a shear field limit group.

Failure:

Occurs if the shear field limit group number could not be found.

Occurs for shearfield added to multiple shear field groups. -> the shear field is added to one of the shear field limit groups.

AddShearReinforcementGroup(int n, string øx, string sx, string øy, string sy, int groupNo = 1)

Adds a shear reinforcement group. n is number of reinforcement layers. Rebar size and spacing can be inserted as either actual value or -1 for auto dimensions.

AddShearFieldToReinforcementGroup(double x1, double x2, double y1, double y2, int groupNo = 1)

Adds a shearfield to a shear reinforcement group.

Failure:

Occurs if the shear reinforcement group number could not be found.

AddStringerReinforcementGroup(string n, string ø, int groupNo = 1)

Adds a stringer reinforcement group. n is number of rebars and ø is size. If rebar dimension is -1 then auto values are implemented.

AddStringerToReinforcementGroup(double x1, double x2, double y1, double y2, int groupNo = 1)

Adds a stringer to a stringer reinforcement group.

Failure:

Occurs if the stringer reinforcement group number could not be found.

LoadAPIModel(string savedFilePath = "")

Open a saved PolyStringer API file (.pstrAPI)

If no savedFilePath is given, then a file can be opened through a dialog box.

OpenModel(string filePath = "")

Saves a PolyStringer API file (.pstrAPI) and opens the file in PolyStringer.

Based on the input provided with the add methods for stringers, loads, openings and shearfields all necessary coordinates (x and y) are added.

filePath: Full path to where the file is saved, if none is provided a dialog box will appear.

Failure:

Occurs if PolyStringer is already opened with a file with the same name.

ClearXCoordinates()

Clears list containing x-coordinates.

ClearYCoordinates()

Clears list containing y-coordinates.

ClearLoadCombinations()
Clears list containing load combinations. Including node- and stringerloads.

ClearCompressionOnlyStringers()
Clears list containing all compression only stringers.

ClearInactiveStringers()
Clears list containing all inactive stringers.

ClearOpenings()
Clears list containing all openings.

ClearNodeSupports()
Clears list containing all node supports.

ClearStringerSupports()
Clears list containing all stringer supports.

ClearShearFieldLimitGroups()
Clears list containing all shear field limit groups.

ClearShearReinforcement()
Clears list containing all shear reinforcement groups.

ClearStringerReinforcement()
Clears list containing all stringer reinforcement groups.