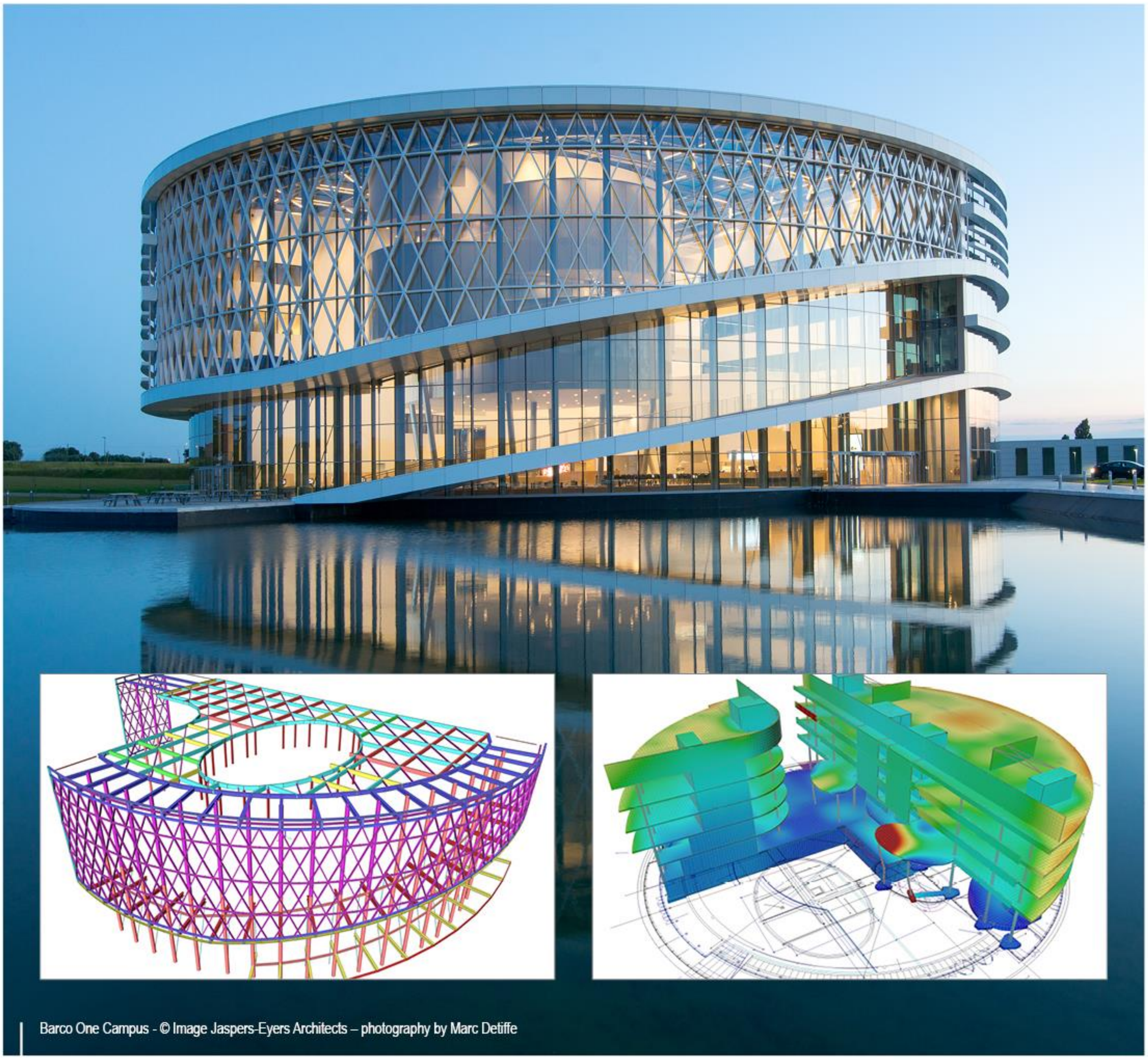
SCIA ENGINEER


Barco One Campus - © Image Jaspers-Eyers Architects – photography by Marc Detiffe

XML data exchange

SCIA Engineer 18

Table of contents

Introduction

In SCIA Engineer there is the possibility to use the XML format to get the data of the Project the users are working on. In the first chapters of this tutorial the different features of the format will be presented in detailed and the XML import/export option explained.

The XML output can be adopted for several uses. The example here presented is to create VBA in Excel which modifies certain parameters of the XML file and allows to update these properties in the model.

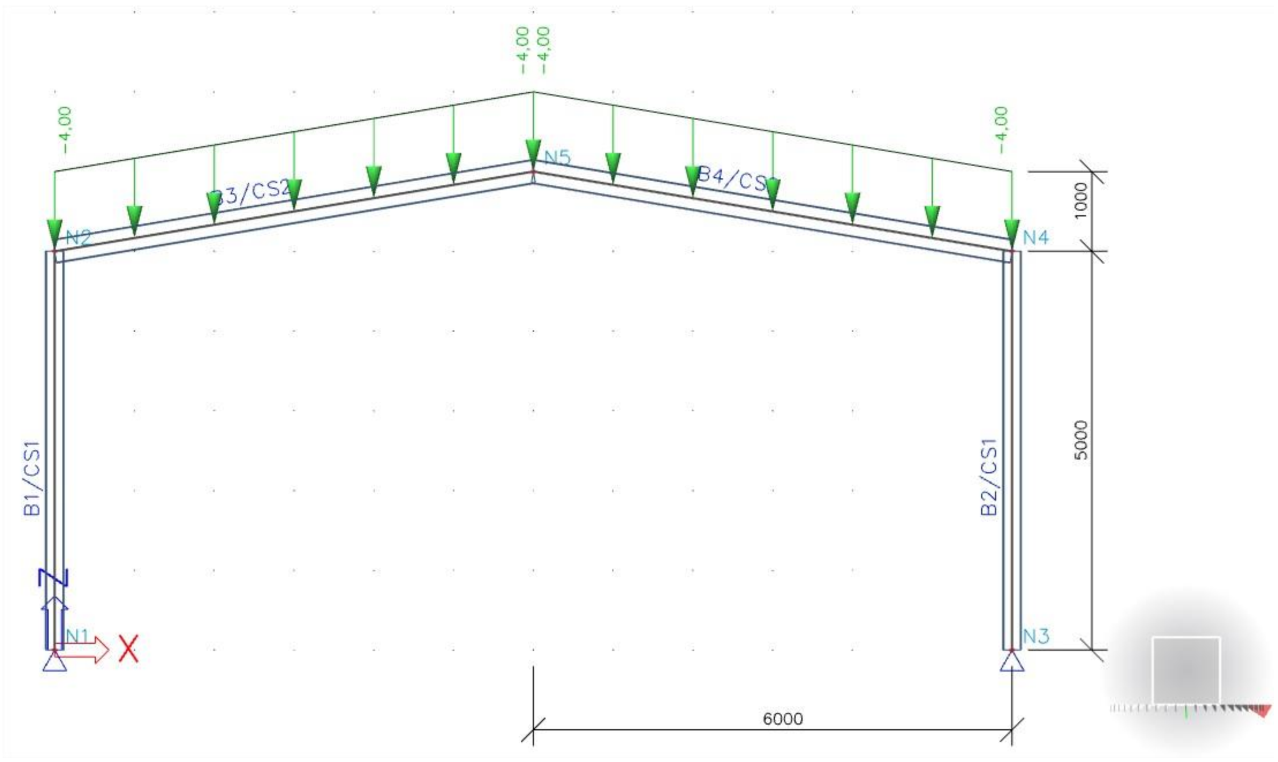
Connected to this format the application ESA_XML.exe has been developed.
It is a powerful technology that allows run SCIA Engineer in the background, and by consequence to run calculations and export results.

Part 1: Import/Export of the XML format

1.1 Creating the model

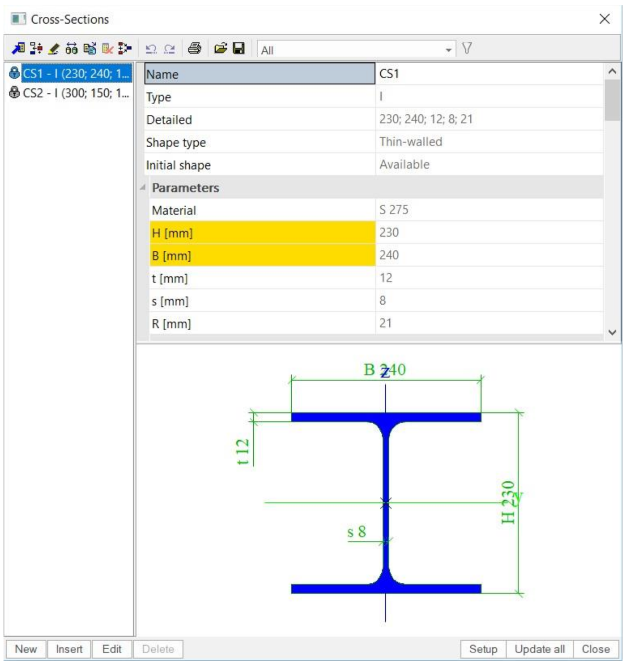
Input of geometry

Create the following structure in SCIA Engineer.



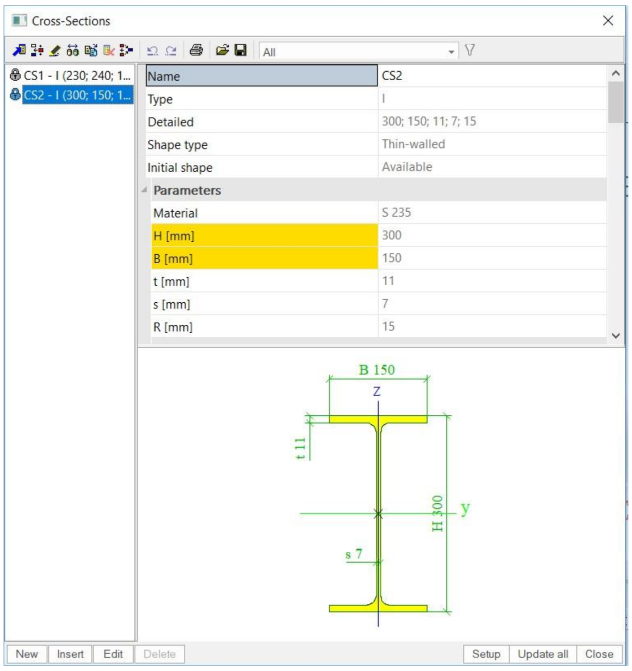
Sections

Columns



Name	CS1
Type	I
Detailed	230; 240; 12; 8; 21
Shape type	Thin-walled
Initial shape	Available
Parameters	
Material	S 275
H [mm]	230
B [mm]	240
t [mm]	12
s [mm]	8
R [mm]	21

Rafters



Name	CS2
Type	I
Detailed	300; 150; 11; 7; 15
Shape type	Thin-walled
Initial shape	Available
Parameters	
Material	S 235
H [mm]	300
B [mm]	150
t [mm]	11
s [mm]	7
R [mm]	15

Note: The use of parametric sections will allow us to influence one of the section parameter to obtain different values.

Loads

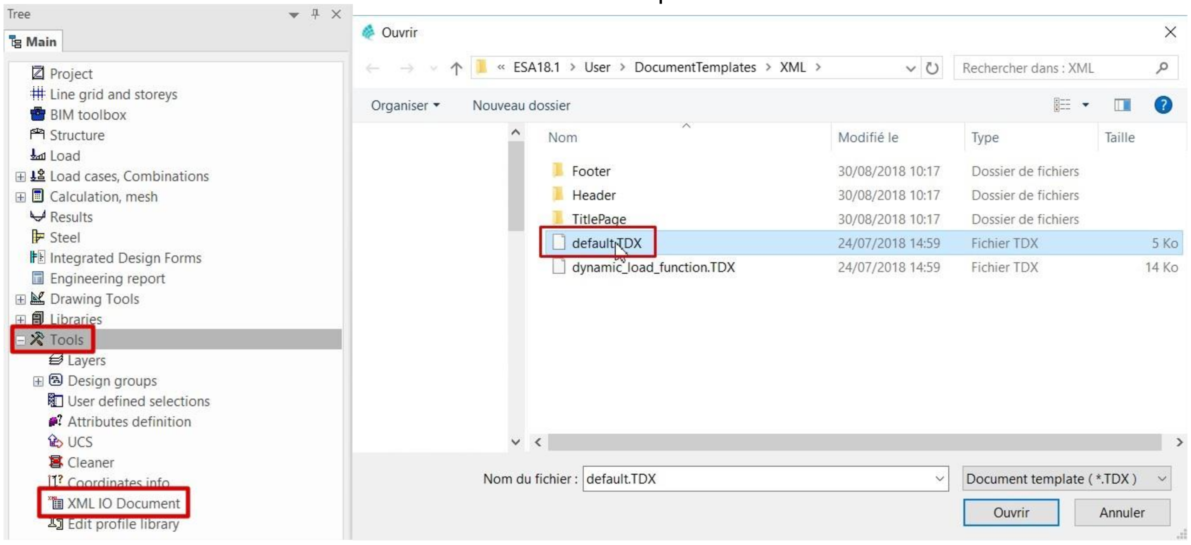
G: Self weight
Q: Variable > line force on roof beams 4 kN/m in Z direction (GCS)

Load Combinations

CO1: EN-ULS Set B
CO2: EN-SLS Characteristic

1.2 Export structure data to XML

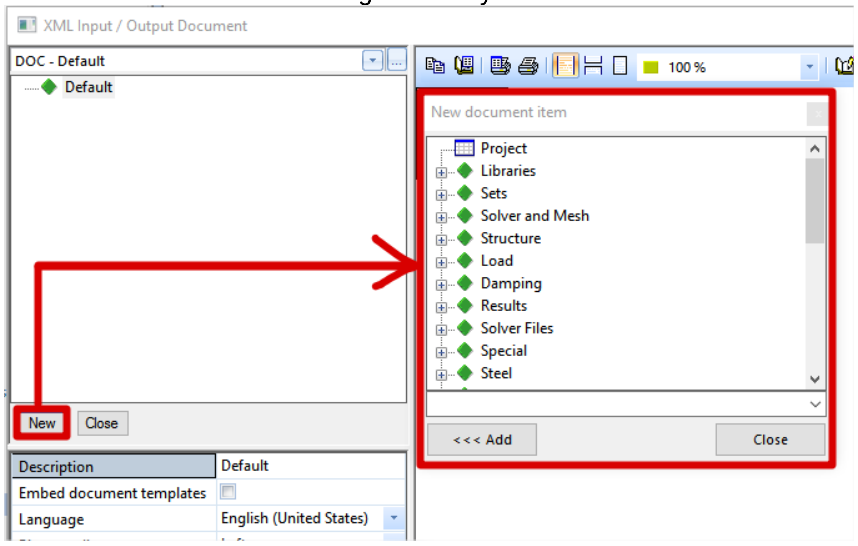
After creating the load cases and load combinations and applying these on the structure we can create the XML-file.
This can be done via Tools > XML IO Document
It will ask you to load a template. SCIA Engineer has some standard templates in which you can choose from. In this case we click on cancel in order to not use a template but to create a new XML-file from scratch.



Nom	Modifié le	Type	Taille
Footer	30/08/2018 10:17	Dossier de fichiers	
Header	30/08/2018 10:17	Dossier de fichiers	
TitlePage	30/08/2018 10:17	Dossier de fichiers	
default.TDX	24/07/2018 14:59	Fichier TDX	5 Ko
dynamic_load_function.TDX	24/07/2018 14:59	Fichier TDX	14 Ko

Nom du fichier : default.TDX
Document template (*.TDX)

We click on new to start adding items in your XML-file

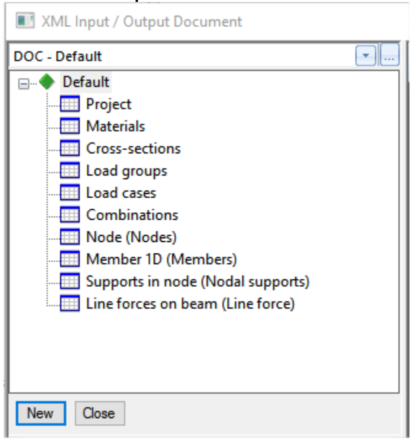


Description	Default
Embed document templates	☐
Language	English (United States)

The order of the added items is very important in an XML-file because the XML-file will be read from top to bottom. For example, you cannot define a load case before defining the load groups because the load group is a property of the load case. This can result in errors when using the XML-file. Another example is that you cannot input a beam without prior inputting the cross-section and the nodes because these are properties of that beam. Always keep the logical order in mind so that you obtain a working XML-file which can be used in SCIA Engineer.

We add to the Document all the data of the structure we have created. Every element added will be shown on the right side of the Screen in the preview. The document should then contain the structural elements (supports, columns, beams) as well as the load cases, load combinations, applied loads.

The below picture should be the end result.

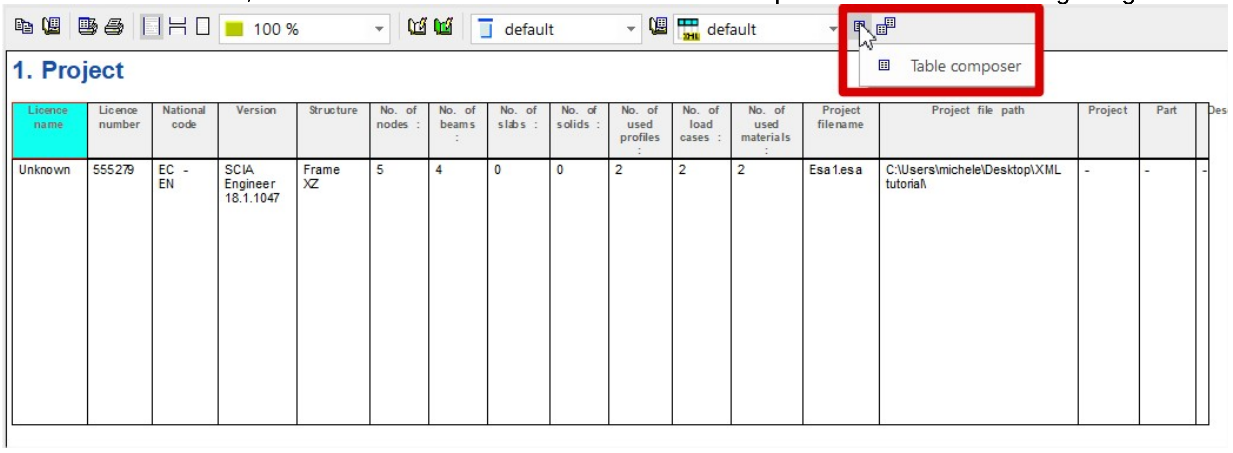


XML Input / Output Document

DOC - Default

- Default
  - Project
  - Materials
  - Cross-sections
  - Load groups
  - Load cases
  - Combinations
  - Node (Nodes)
  - Member 1D (Members)
  - Supports in node (Nodal supports)
  - Line forces on beam (Line force)

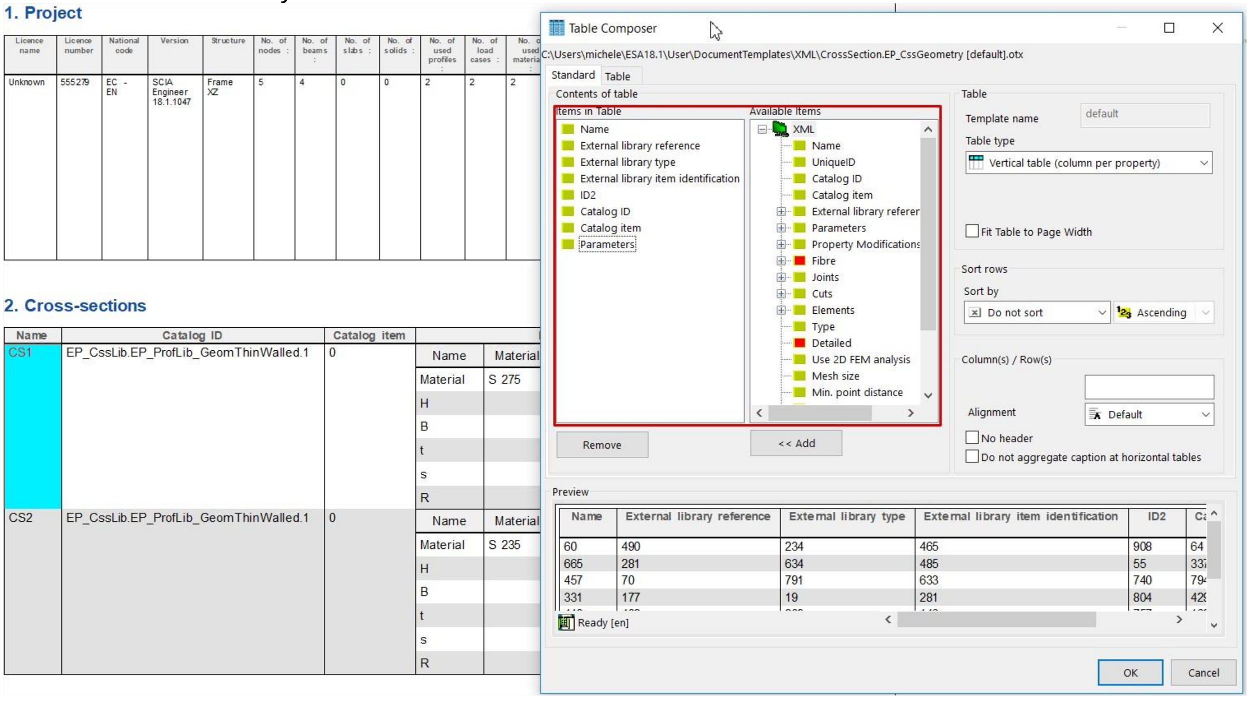
In the properties shown in the bottom part we can change only the text shown in the Preview.
To amend the tables, we can select them and click on Table Composer as for the following image.



1. Project

Licence name	Licence number	National code	Version	Structure	No. of nodes :	No. of beams :	No. of slabs :	No. of solids :	No. of used profiles :	No. of load cases :	No. of used materials :	Project filename	Project file path	Project	Part	Des
Unknown	555279	EC - EN	SCIA Engineer 18.1.1047	Frame XZ	5	4	0	0	2	2	2	Esa1.esa	C:\Users\michele\Desktop\XML tutorial\	-	-	-

All the data shown in the Table Composer can be added to our Document and exported in the XML document and then used externally.



1. Project

Licence name	Licence number	National code	Version	Structure	No. of nodes :	No. of beams :	No. of slabs :	No. of solids :	No. of used profiles :	No. of load cases :	No. of used materia
Unknown	555279	EC - EN	SCIA Engineer 18.1.1047	Frame XZ	5	4	0	0	2	2	2

2. Cross-sections

Name	Catalog ID	Catalog item		
CS1	EP_CssLib.EP_ProfLib_GeomThinWalled.1	0	Name	Material
			Material	S 275
			H	
			B	
			t	
			s	
			R	
CS2	EP_CssLib.EP_ProfLib_GeomThinWalled.1	0	Name	Material
			Material	S 235
			H	
			B	
			t	
			s	
			R	

Table Composer

C:\Users\michele\ESA18.1\User\DocumentTemplates\XML\CrossSection.EP_CssGeometry [default].otx

Standard | Table

Contents of table

Items in Table:
- Name
- External library reference
- External library type
- External library item identification
- ID2
- Catalog ID
- Catalog item
- Parameters

Available Items:
- XML
  - Name
  - UniqueID
  - Catalog ID
  - Catalog item
  - External library referer
  - Parameters
  - Property Modifications
  - Fibre
  - Joints
  - Cuts
  - Elements
  - Type
  - Detailed
  - Use 2D FEM analysis
  - Mesh size
  - Min. point distance

Table

Template name: default

Table type: Vertical table (column per property)

☐ Fit Table to Page Width

Sort rows

Sort by: Do not sort | Ascending

Column(s) / Row(s)

Alignment: Default

☐ No header

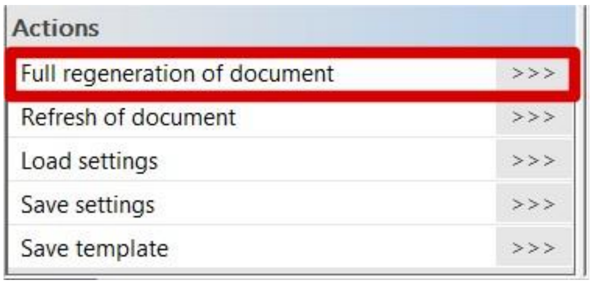
☐ Do not aggregate caption at horizontal tables

Preview

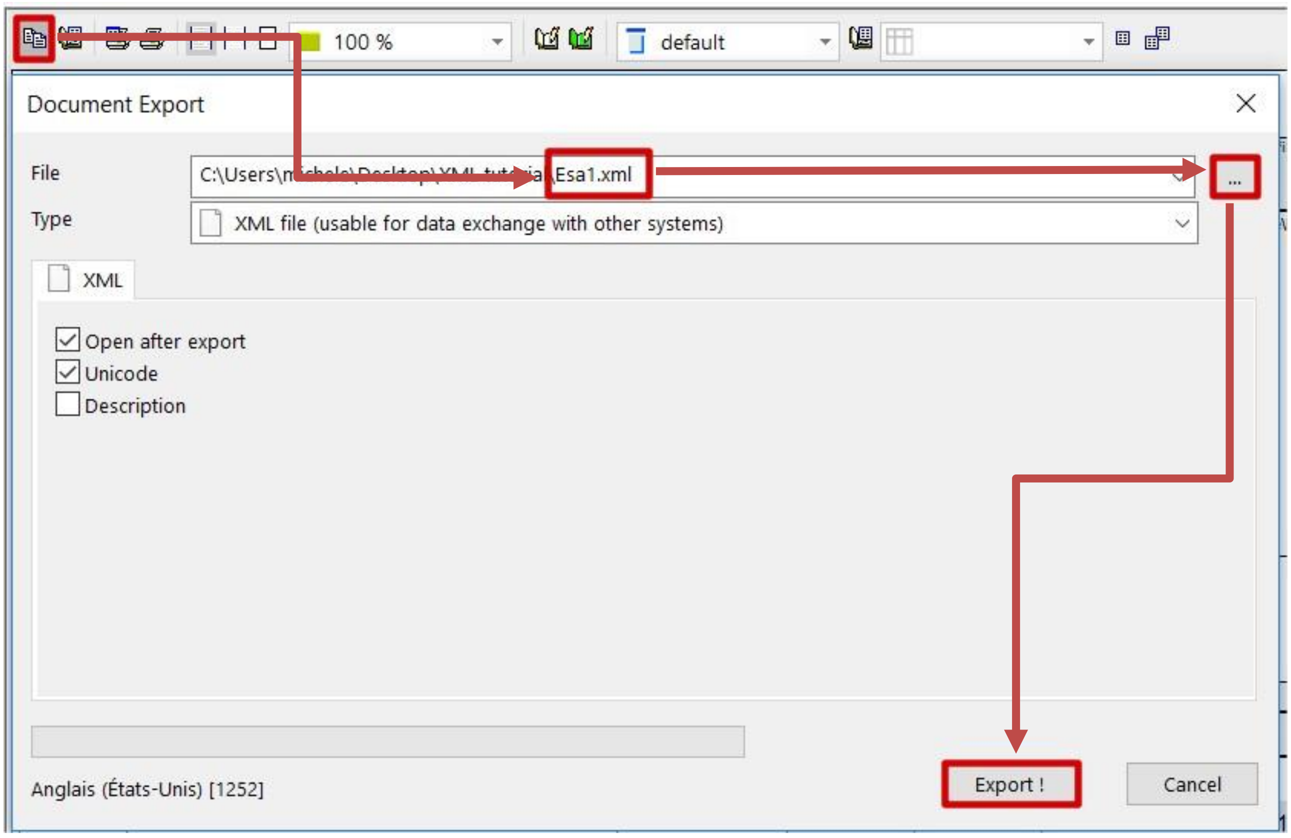
Name	External library reference	External library type	External library item identification	ID2	Ca
60	490	234	465	908	64
665	281	634	485	55	337
457	70	791	633	740	79
331	177	19	281	804	42

Ready [en]

We export then the XML document. Before exporting this data to XML make sure that the content is fully regenerated.



We click on export and navigate to the folder in which you want to export this data to XML.
For the XML document the Unicode Standard needs to be used. It is selected by default.



File: C:\Users\m...\Desktop\XML tut...ia\Esa1.xml

Type: XML file (usable for data exchange with other systems)

XML

- [x] Open after export
- [x] Unicode
- [ ] Description

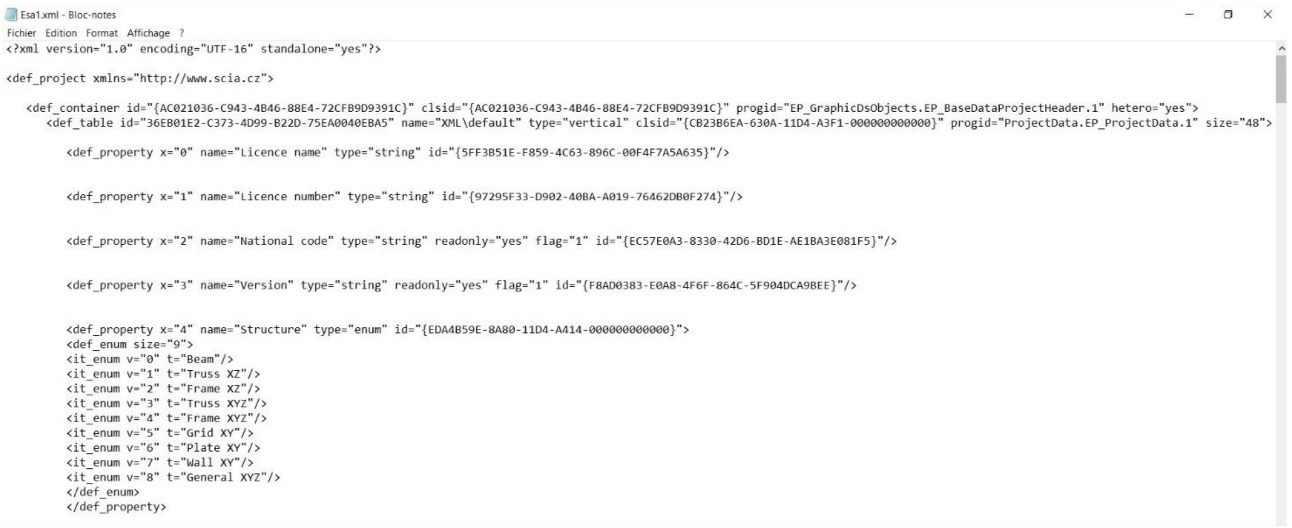
Anglais (États-Unis) [1252]

Part 2: Reading the XML file

2.1 The DEF file

Once we have exported the document in XML, we notice two files have been created, a XML document and a file DEF. The file DEF contains the list of all the parameters of the tables we have exported. The XML document contains all the information of the particular Project we are modelling, so the XML document refers to the DEF file to read all the different possibilities and indicate the ones chosen in the project created.

We open the generated DEF file using Notepad (txt reader).



```
<?xml version="1.0" encoding="UTF-16" standalone="yes"?>

<def_project xmlns="http://www.scia.cz">

   <def_container id="{AC021036-C943-4B46-88E4-72CFB9D9391C}" clsid="{AC021036-C943-4B46-88E4-72CFB9D9391C}" progid="EP_GraphicDsObjects.EP_BaseDataProjectHeader.1" hetero="yes">
      <def_table id="36EB01E2-C373-4D99-B22D-75EA0040EBA5" name="XML\default" type="vertical" clsid="{CB23B6EA-630A-11D4-A3F1-000000000000}" progid="ProjectData.EP_ProjectData.1" size="48">

         <def_property x="0" name="Licence name" type="string" id="{5FF3B51E-F859-4C63-896C-00F4F7A5A635}"/>


         <def_property x="1" name="Licence number" type="string" id="{97295F33-D902-40BA-A019-76462DB0F274}"/>


         <def_property x="2" name="National code" type="string" readonly="yes" flag="1" id="{EC57E0A3-8330-42D6-BD1E-AE1BA3E081F5}"/>


         <def_property x="3" name="Version" type="string" readonly="yes" flag="1" id="{F8AD0383-E0A8-4F6F-864C-5F904DCA9BEE}"/>


         <def_property x="4" name="Structure" type="enum" id="{EDA4B59E-8A80-11D4-A414-000000000000}">
         <def_enum size="9">
         <it_enum v="0" t="Beam"/>
         <it_enum v="1" t="Truss XZ"/>
         <it_enum v="2" t="Frame XZ"/>
         <it_enum v="3" t="Truss XYZ"/>
         <it_enum v="4" t="Frame XYZ"/>
         <it_enum v="5" t="Grid XY"/>
         <it_enum v="6" t="Plate XY"/>
         <it_enum v="7" t="Wall XY"/>
         <it_enum v="8" t="General XYZ"/>
         </def_enum>
         </def_property>
```

We can see at the top the information related to the xml version, the standard used (Unicode) and the type of Protection (Standalone)

Containers

In order to read both XML document and DEF file, we can orientate with the different "Containers". Each container is one of the tables added in the Document created in SCIA.

The first one is in fact Project and contains all the parameters we chose when starting a new project. We can see them defined as def_properties. If we read all the DEF file, we can see then Materials, Sections, etc., in which all the general information are reported.

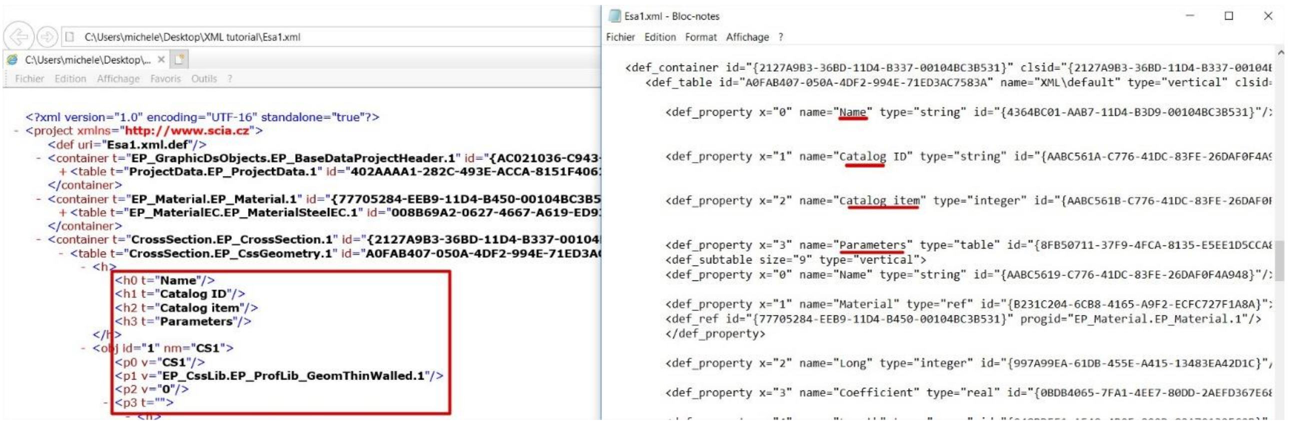
2.2 The XML document

We open the generated XML document using Notepad (txt reader) or the Browser Explorer (Html format).

Let's focus on the third Container: Cross Sections. As shown in the image below the DEF File (on the right) contains the general information to define a cross section and the XML file (on the left) contains the information related to the project we have defined.

In our case the properties Name, Catalog ID, Catalog Item and Parameters are recalled from the DEF File and defined as:

- Name: CS1, in the DEF file is defined as string, so a word.
- Catalog ID: ThinWalled Section (in the DEF file is defined as string, the library path has been recalled),
- Catalog Item: I section (in the DEF file is defined as integer. As it is the first one of the catalog, the number 0 has been assigned)
- Parameters: in the DEF file a table is defined with 9 parameters.

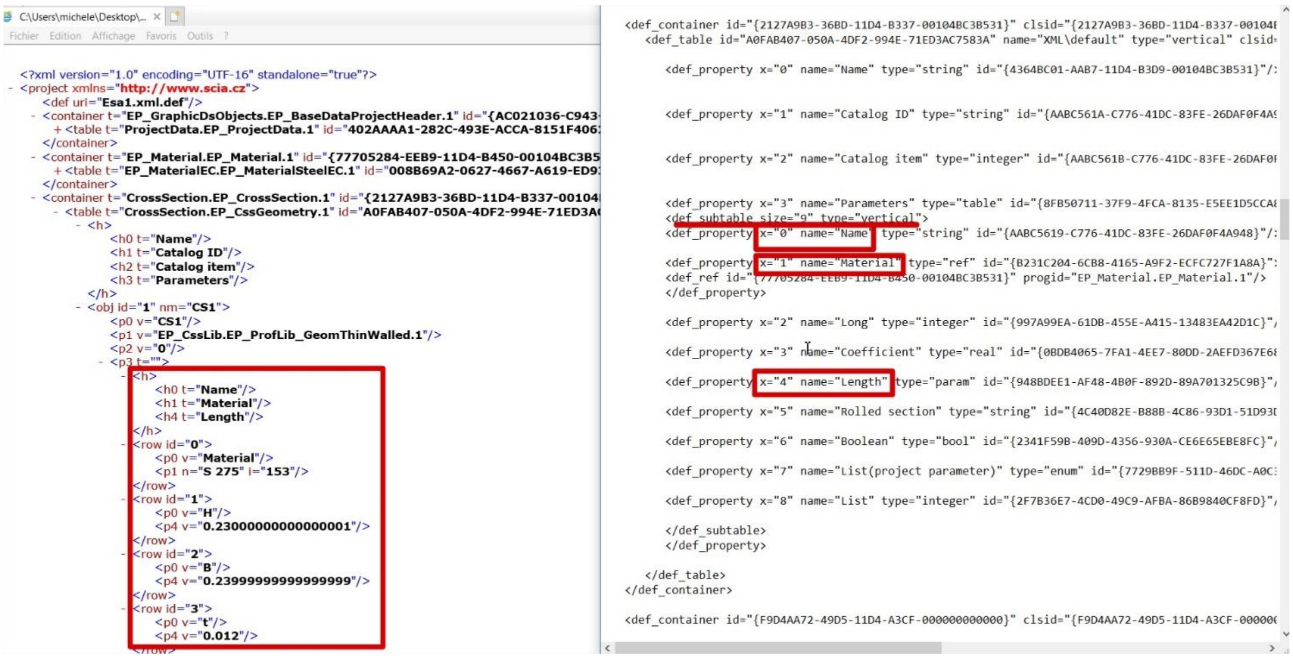


```
C:\Users\michele\Desktop\XML tutorial\Esa1.xml

<?xml version="1.0" encoding="UTF-16" standalone="true"?>
- <project xmlns="http://www.scia.cz">
     <def uri="Esa1.xml.def"/>
   - <container t="EP_GraphicDsObjects.EP_BaseDataProjectHeader.1" id="{AC021036-C943
       + <table t="ProjectData.EP_ProjectData.1" id="402AAAA1-282C-493E-ACCA-8151F406
     </container>
   - <container t="EP_Material.EP_Material.1" id="{77705284-EEB9-11D4-B450-00104BC3B5
       + <table t="EP_MaterialEC.EP_MaterialSteelEC.1" id="008B69A2-0627-4667-A619-ED9
     </container>
   - <container t="CrossSection.EP_CrossSection.1" id="{2127A9B3-36BD-11D4-B337-00104
     - <table t="CrossSection.EP_CssGeometry.1" id="A0FAB407-050A-4DF2-994E-71ED3A
       - <h>
             <h0 t="Name"/>
             <h1 t="Catalog ID"/>
             <h2 t="Catalog item"/>
             <h3 t="Parameters"/>
         </h>
       - <obj id="1" nm="CS1">
             <p0 v="CS1"/>
             <p1 v="EP_CssLib.EP_ProfLib_GeomThinWalled.1"/>
             <p2 v="0"/>
           - <p3 t="">
```

```
Esa1.xml - Bloc-notes

<def_container id="{2127A9B3-36BD-11D4-B337-00104BC3B531}" clsid="{2127A9B3-36BD-11D4-B337-00104
   <def_table id="A0FAB407-050A-4DF2-994E-71ED3AC7583A" name="XML\default" type="vertical" clsid=

      <def_property x="0" name="Name" type="string" id="{4364BC01-AAB7-11D4-B3D9-00104BC3B531}"/

      <def_property x="1" name="Catalog ID" type="string" id="{AABC561A-C776-41DC-83FE-26DAF0F4A

      <def_property x="2" name="Catalog item" type="integer" id="{AABC561B-C776-41DC-83FE-26DAF0

      <def_property x="3" name="Parameters" type="table" id="{8FB50711-37F9-4FCA-8135-E5EE1D5CCA
      <def_subtable size="9" type="vertical">
      <def_property x="0" name="Name" type="string" id="{AABC5619-C776-41DC-83FE-26DAF0F4A948}"/

      <def_property x="1" name="Material" type="ref" id="{B231C204-6CB8-4165-A9F2-ECFC727F1A8A}"
      <def_ref id="{77705284-EEB9-11D4-B450-00104BC3B531}" progid="EP_Material.EP_Material.1"/>
      </def_property>

      <def_property x="2" name="Long" type="integer" id="{997A99EA-61DB-455E-A415-13483EA42D1C}"

      <def_property x="3" name="Coefficient" type="real" id="{0BDB4065-7FA1-4EE7-80DD-2AEFD367E6
```

For the forth property, number 3, a subtable has been defined and 9 possible parameters can be recalled.

In this case, in the project only 3 of these are defined: Name, Material and Length.
In the first row, number 0, the name is Material, and the material is defined; in the other rows, number 1-2-3-4-5, the name is one of the parameters of the section, and a value (Length) is defined for each one of the properties.



```
C:\Users\michele\Desktop\...

<?xml version="1.0" encoding="UTF-16" standalone="true"?>
- <project xmlns="http://www.scia.cz">
     <def uri="Esa1.xml.def"/>
   - <container t="EP_GraphicDsObjects.EP_BaseDataProjectHeader.1" id="{AC021036-C943
       + <table t="ProjectData.EP_ProjectData.1" id="402AAAA1-282C-493E-ACCA-8151F406
     </container>
   - <container t="EP_Material.EP_Material.1" id="{77705284-EEB9-11D4-B450-00104BC3B5
       + <table t="EP_MaterialEC.EP_MaterialSteelEC.1" id="008B69A2-0627-4667-A619-ED9
     </container>
   - <container t="CrossSection.EP_CrossSection.1" id="{2127A9B3-36BD-11D4-B337-00104
     - <table t="CrossSection.EP_CssGeometry.1" id="A0FAB407-050A-4DF2-994E-71ED3A
       - <h>
             <h0 t="Name"/>
             <h1 t="Catalog ID"/>
             <h2 t="Catalog item"/>
             <h3 t="Parameters"/>
         </h>
       - <obj id="1" nm="CS1">
             <p0 v="CS1"/>
             <p1 v="EP_CssLib.EP_ProfLib_GeomThinWalled.1"/>
             <p2 v="0"/>
           - <p3 t="">
             - <h>
                   <h0 t="Name"/>
                   <h1 t="Material"/>
                   <h4 t="Length"/>
               </h>
             - <row id="0">
                   <p0 v="Material"/>
                   <p1 n="S 275" i="153"/>
               </row>
             - <row id="1">
                   <p0 v="H"/>
                   <p4 v="0.23000000000000001"/>
               </row>
             - <row id="2">
                   <p0 v="B"/>
                   <p4 v="0.23999999999999999"/>
               </row>
             - <row id="3">
                   <p0 v="t"/>
                   <p4 v="0.012"/>
```

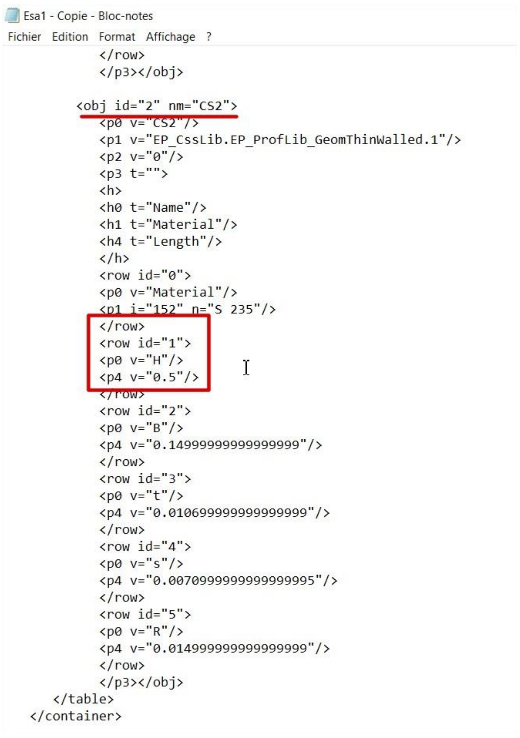
```
<def_container id="{2127A9B3-36BD-11D4-B337-00104BC3B531}" clsid="{2127A9B3-36BD-11D4-B337-00104
   <def_table id="A0FAB407-050A-4DF2-994E-71ED3AC7583A" name="XML\default" type="vertical" clsid=

      <def_property x="0" name="Name" type="string" id="{4364BC01-AAB7-11D4-B3D9-00104BC3B531}"/

      <def_property x="1" name="Catalog ID" type="string" id="{AABC561A-C776-41DC-83FE-26DAF0F4A

      <def_property x="2" name="Catalog item" type="integer" id="{AABC561B-C776-41DC-83FE-26DAF0

      <def_property x="3" name="Parameters" type="table" id="{8FB50711-37F9-4FCA-8135-E5EE1D5CCA
      <def_subtable size="9" type="vertical">
      <def_property x="0" name="Name" type="string" id="{AABC5619-C776-41DC-83FE-26DAF0F4A948}"/

      <def_property x="1" name="Material" type="ref" id="{B231C204-6CB8-4165-A9F2-ECFC727F1A8A}"
      <def_ref id="{77705284-EEB9-11D4-B450-00104BC3B531}" progid="EP_Material.EP_Material.1"/>
      </def_property>

      <def_property x="2" name="Long" type="integer" id="{997A99EA-61DB-455E-A415-13483EA42D1C}"

      <def_property x="3" name="Coefficient" type="real" id="{0BDB4065-7FA1-4EE7-80DD-2AEFD367E6

      <def_property x="4" name="Length" type="param" id="{948BDEE1-AF48-4B0F-892D-89A701325C9B}"

      <def_property x="5" name="Rolled section" type="string" id="{4C40D82E-B88B-4C86-93D1-51D93

      <def_property x="6" name="Boolean" type="bool" id="{2341F59B-409D-4356-930A-CE6E65EBE8FC}"

      <def_property x="7" name="List(project parameter)" type="enum" id="{7729BB9F-511D-46DC-A0C

      <def_property x="8" name="List" type="integer" id="{2F7B36E7-4CD0-49C9-AFBA-86B9840CF8FD}"

      </def_subtable>
      </def_property>

   </def_table>
</def_container>

<def_container id="{F9D4AA72-49D5-11D4-A3CF-000000000000}" clsid="{F9D4AA72-49D5-11D4-A3CF-00000
```

We know now how to read the XML document, so we are able to modify it in order to obtain the desired results.

In this paragraph we show how to amend the XML document manually. In the next chapter we will use instead a Macro created in Excel in order to modify more parameters automatically.

Modify the XML document

In this case for example we want to modify the height of the rafter and update our model with this update. We copy the XML document and DEF file. We open the copy of the XML document and we modify the height of the beam in the appropriate container:



Esa1 - Copie - Bloc-notes

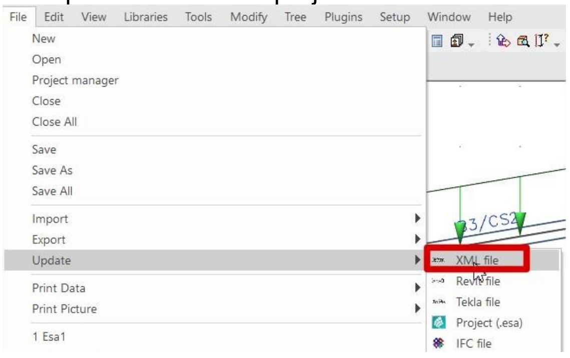
Fichier Edition Format Affichage ?

```
         </row>
         </p3></obj>

      <obj id="2" nm="CS2">
         <p0 v="CS2"/>
         <p1 v="EP_CssLib.EP_ProfLib_GeomThinWalled.1"/>
         <p2 v="0"/>
         <p3 t="">
         <h>
         <h0 t="Name"/>
         <h1 t="Material"/>
         <h4 t="Length"/>
         </h>
         <row id="0">
         <p0 v="Material"/>
         <p1 i="152" n="S 235"/>
         </row>
         <row id="1">
         <p0 v="H"/>
         <p4 v="0.5"/>
         </row>
         <row id="2">
         <p0 v="B"/>
         <p4 v="0.14999999999999999"/>
         </row>
         <row id="3">
         <p0 v="t"/>
         <p4 v="0.010699999999999999"/>
         </row>
         <row id="4">
         <p0 v="s"/>
         <p4 v="0.0070999999999999995"/>
         </row>
         <row id="5">
         <p0 v="R"/>
         <p4 v="0.014999999999999999"/>
         </row>
         </p3></obj>
   </table>
</container>
```

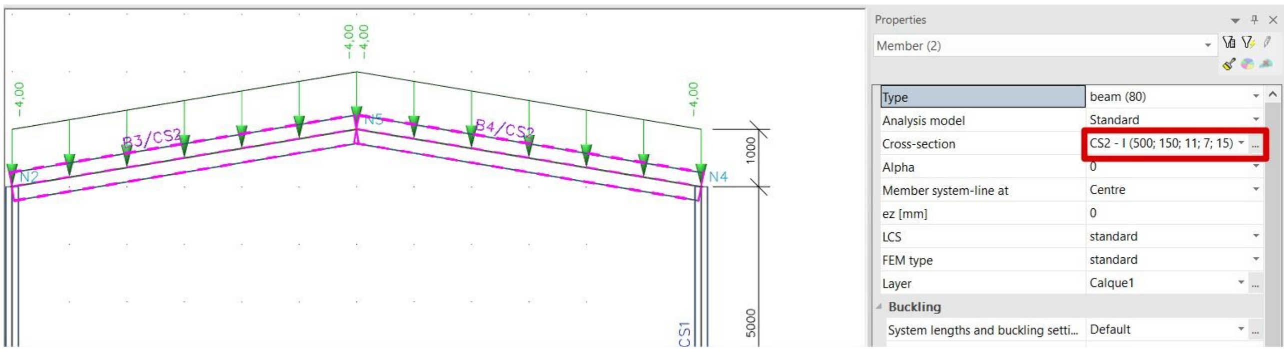
We save and close the XML document.

Update the structure with XML

We update the current project with this XML-file via File > Update > XML-file



As the software reads the XML document, the changes are applied.



Properties

Member (2)

Type	beam (80)
Analysis model	Standard
Cross-section	CS2 - I (500; 150; 11; 7; 15)
Alpha	0
Member system-line at	Centre
ez [mm]	0
LCS	standard
FEM type	standard
Layer	Calque1
Buckling	
System lengths and buckling setti...	Default

Part 3: Link Excel-XML-SCIA

3.1 Modify an XML file using Macros in Excel

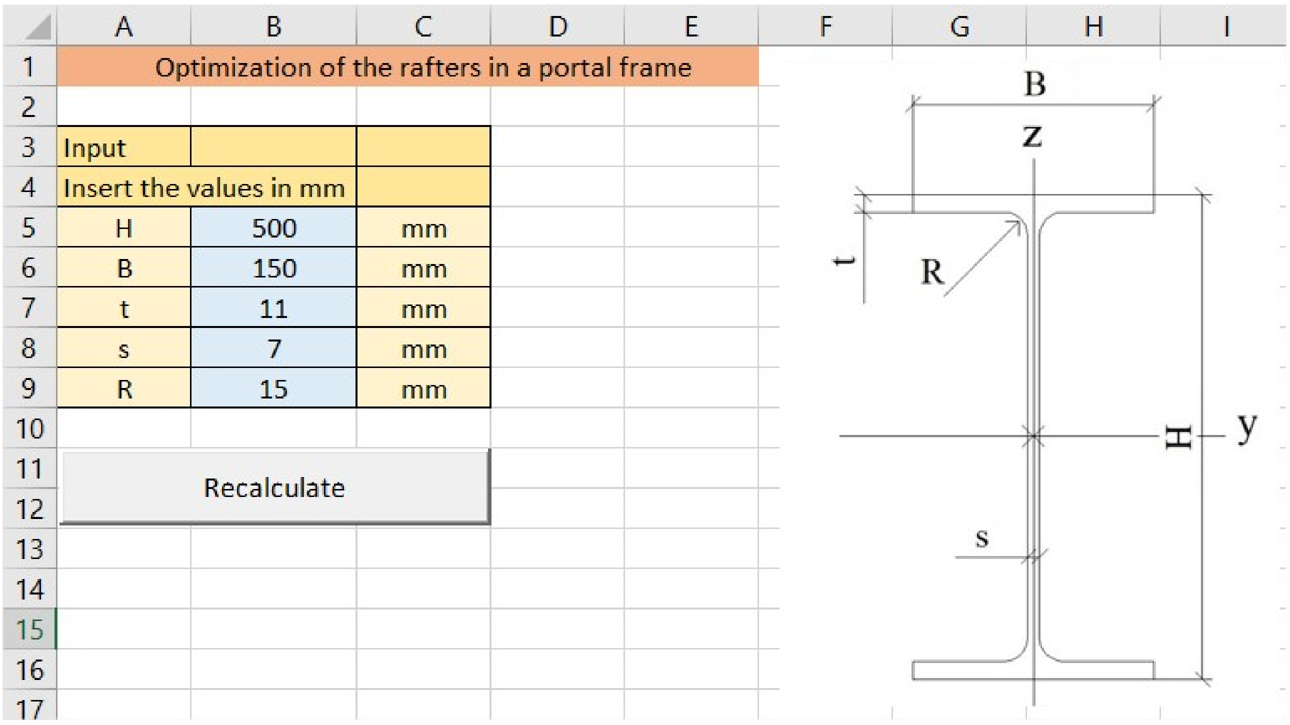
We will see in this chapter how to create a Macro which gives as output an XML document modified with the parameters we have inserted in Excel.

Notes: from this Chapter on, a strong basis of Excel VBA is required.

The Excel sheet will allow us to modify externally the dimensions of the section of the rafters of the project we have modelled in the previous chapter.

Let us now start MS EXCEL, create a new document and define two sheets. Let us name them "Table" and "XML". The first one (Table) will be our "User interface", i.e. it will be the sheet that will be used by the user. The other one (XML) will be used as an auxiliary sheet to hold the contents of the XML file we have to generate.

We open Excel and we create a table with the parameters we want to influence, as for the image below.



	A	B	C
1	Optimization of the rafters in a portal frame		
2			
3	Input		
4	Insert the values in mm		
5	H	500	mm
6	B	150	mm
7	t	11	mm
8	s	7	mm
9	R	15	mm

We go now to the second Excel sheet, called “XML”, in which we copy in the second column the XML document created before. In this sheet we are going then to add the references for the script we are going to write.

We scroll the script and we position the references H, B, t, s and R as for image below:

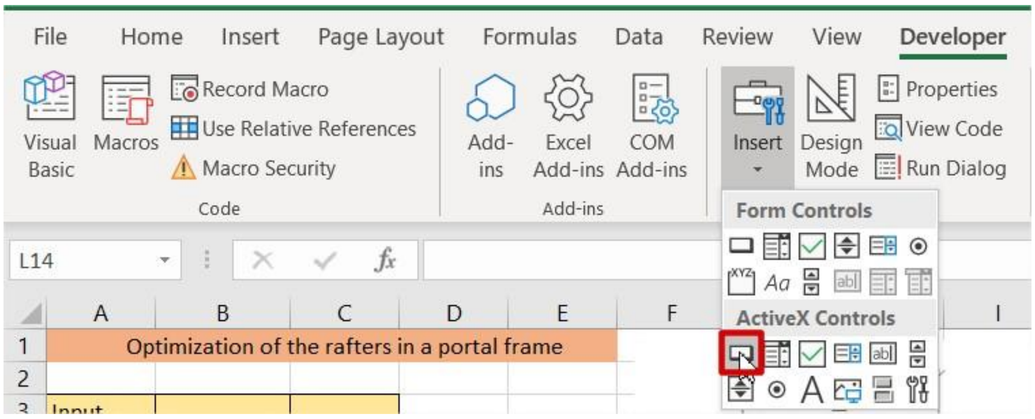
	A	B
		[…]
190		<obj id="2" nm="CS2">
191		<p0 v="CS2"/>
192		<p1 v="EP_CssLib.EP_ProfLib_GeomThinWalled.1"/>
193		<p2 v="0"/>
194		<p3 t="">
195		<h>
196		<h0 t="Name"/>
197		<h1 t="Material"/>

198		<h4 t="Length"/>
199		</h>
200		<row id="0">
201		<p0 v="Material"/>
202		<p1 i="152" n="S 235"/>
203		</row>
204		<row id="1">
205		<p0 v="H"/>
206	H	<p4 v="0.29999999999999999"/>
207		</row>
208		<row id="2">
209		<p0 v="B"/>
210	B	<p4 v="0.14999999999999999"/>
211		</row>
212		<row id="3">
213		<p0 v="t"/>
214	t	<p4 v="0.010699999999999999"/>
215		</row>
216		<row id="4">
217		<p0 v="s"/>
218	s	<p4 v="0.0070999999999999995"/>
219		</row>
220		<row id="5">
221		<p0 v="R"/>
222	R	<p4 v="0.014999999999999999"/>
223		</row>
224		</p3></obj>
225		</table>
		[…]

Create the VBA

We create now the VBA script, which allows us to create a new XML document by clicking on “Recalculate” in the first Excel sheet, and which will contain the new values of the dimensions of the cross section “CS2”.

To add the Recalculate command, we go to developer and we add a Command Button (active X control). We go to the properties and we can change the name and reference to “Recalculate”



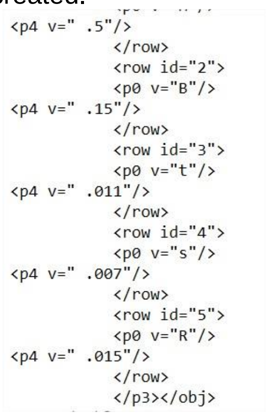
Optimization of the rafters in a portal frame

Input

We can create our script. The example below can be used (please control and modify the path of the folders following their location in the machine):

```
Private Sub Recalculate_Click()
' Open XML file for writing'
Dim fs, f
Set fs = CreateObject("Scripting.FileSystemObject")
Set f = fs.CreateTextFile("C:\Users\michele\Desktop\XML_tutorial\XMLex.xml", True, True)
' we assume that the data are stored in folder C:\Users\michele\Desktop\ XML_tutorial
' Generate XML file using the two input values
Dim SomethingToWrite As Boolean
SomethingToWrite = True
Dim mystring As String
Dim i As Integer
i = 1
Do
mystring = Worksheets("XML").Cells(i, 2).Value
If mystring <> "" Then
If Worksheets("XML").Cells(i, 1).Value = "H" Then
' this line stores the value of the diameter'
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(5, 2).Value / 1000) & """/>"
End If
If Worksheets("XML").Cells(i, 1).Value = "B" Then
' this line storesthe value of the thickness'
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(6, 2).Value / 1000) & """/>"
End If
If Worksheets("XML").Cells(i, 1).Value = "t" Then
' this line storesthe value of the thickness'
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(7, 2).Value / 1000) & """/>"
End If
If Worksheets("XML").Cells(i, 1).Value = "s" Then
' this line storesthe value of the thickness'
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(8, 2).Value / 1000) & """/>"
End If
If Worksheets("XML").Cells(i, 1).Value = "R" Then
' this line storesthe value of the thickness'
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(9, 2).Value / 1000) & """/>"
End If
f.WriteLine (mystring)
i = i + 1
Else
SomethingToWrite = False
End If
Loop While SomethingToWrite = True
f.Close
End Sub
```

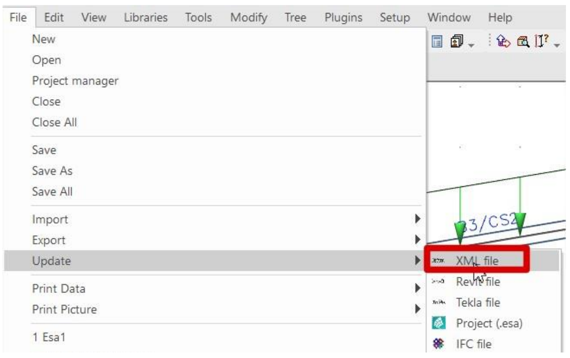
The VBA script will create a file XML, in the folder in which we have saved the DEF file, with the new dimensions of the section of the rafters, which overwrite the one of the script before.
We can open and check the new XML file created.



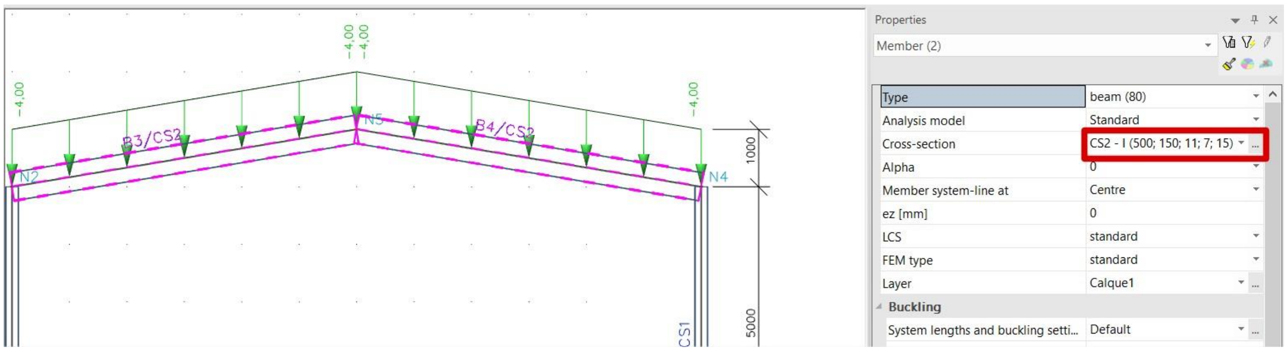
```
<p4 v=" .5"/>
            </row>
            <row id="2">
            <p0 v="B"/>
<p4 v=" .15"/>
            </row>
            <row id="3">
            <p0 v="t"/>
<p4 v=" .011"/>
            </row>
            <row id="4">
            <p0 v="s"/>
<p4 v=" .007"/>
            </row>
            <row id="5">
            <p0 v="R"/>
<p4 v=" .015"/>
            </row>
            </p3></obj>
```

Update the structure with XML

We update the current project with this XML-file via File > Update > XML-file



As the software reads the XML document, the changes are applied.



Type	beam (80)
Analysis model	Standard
Cross-section	CS2 - I (500; 150; 11; 7; 15)
Alpha	0
Member system-line at	Centre
ez [mm]	0
LCS	standard
FEM type	standard
Layer	Calque1
Buckling	
System lengths and buckling setti...	Default

Part 4: Running a calculation

4.1 the esa_XML.exe application

In order to run SCIA in the background, the application esa_XML has been created. It is executable from another program, that is capable of using XML file to modify ESA project data and obtain outputs in various formats.
It is a command line program and is able to perform the following tasks.

- open the existing ESA project
- read the given XML format and use its data to override the project
- perform the given type of calculation
- regenerate the existing document in the ESA project and export it into the selected format (HTML,TXT,ESA). In case of several documents, one can specify the name of the document, otherwise the current document is used.
- regenerate the existing output XML format and use it to generate XML file. If there are more than one output formats in the project, it is possible to select the format by name.

Command line parameters of esa_XML

Calculation type :
NONE = without any recalculations
NOC = No calculation
LIN = Linear calculation (Delete all calculated results when exists)
NEL = Nonlinear calculation
CON = Nonlinear concrete calculation
EIG = Eigen frequencies calculation
STB = Stability calculation
INF = Influence lines calculation
MOB = Mobile loads calculation
TDA = TDA calculation
SLN = Soilin calculation
PHA = Phases calculation
NPH = Nonlinear phases
CSS = Recalculation of cross sections
NST = Nonlinear stability
TID = Test of input data - solver link only

If text CMD is input, it is possible to perform arbitrary number of actions within one run of the program.

The second parameter is the name of text file, where each line has the same syntax as the whole program and leads to opening of one file (sub-levels are not allowed in CMD).
- input ESA project (file name including path)
- input XML file (file name including path) – optional parameter

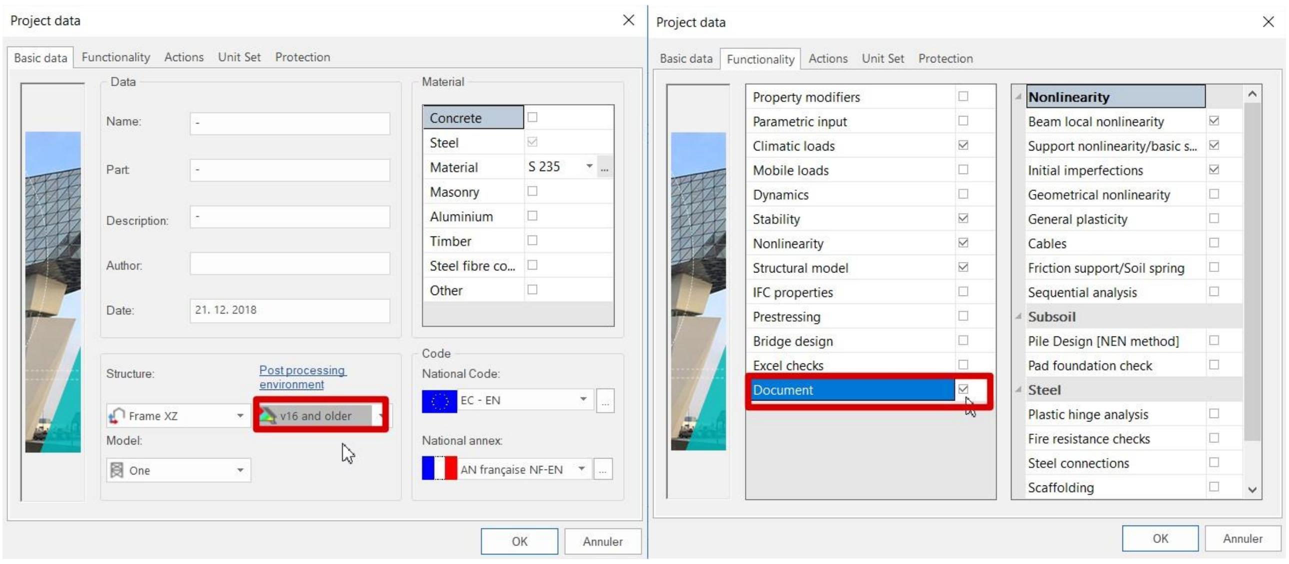
Third parameter: Switches – starting with '/' or '-' character

/t – output file type – TXT, HTML,ESA
if not stated, the output file type is set to ESA
/l – log file name
/o – output file name, the extension must be specified by used according to the selected output file type
if not stated, no output is performed
/x – name of output XML file
if not stated, no output is performed
/d – name of document from which the output is performed, if not stated, the current document is used
/m – name of output XML format, if not stated, the current one is used

4.2 VBA script to run esa_XML

We will prepare a VBA script (run from an XLS sheet) that will perform a calculation of the displacement and of the moment My in the rafters. SCIA Engineer as a calculation engine will be running in the background. The user won’t even spot it on the screen and will be able to think that the EXCEL sheet itself does everything.

All the results of the project need to be in the Document of the project. We open our esa1.exe file and in the Project menu we switch from v17 to v16 in order to activate the Document in the functionality menu.



Project data

Basic data | Functionality | Actions | Unit Set | Protection

Data

Name:	-
Part:	-
Description:	-
Author:	
Date:	21. 12. 2018

Material

Concrete	☐
Steel	☑
Material	S 235
Masonry	☐
Aluminium	☐
Timber	☐
Steel fibre co...	☐
Other	☐

Structure: Frame XZ

Post processing environment: v16 and older

Model: One

Code

National Code: EC - EN

National annex: AN française NF-EN

OK | Annuler

Project data

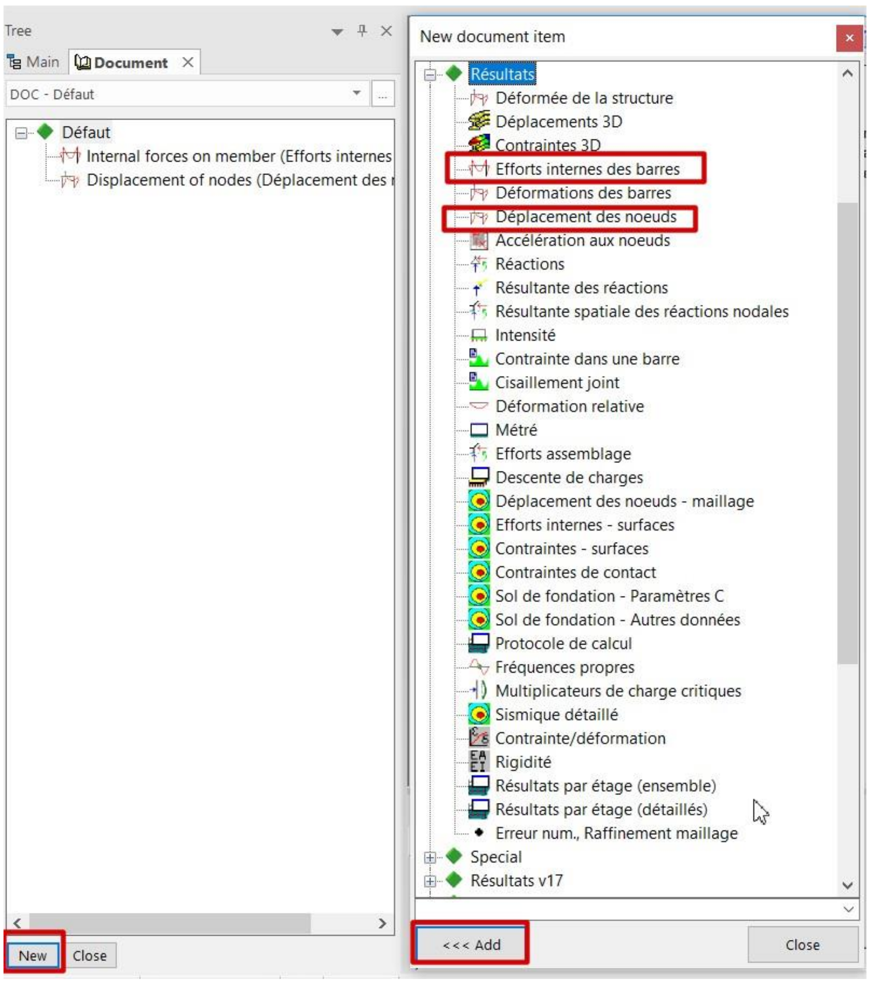
Basic data | Functionality | Actions | Unit Set | Protection

Property modifiers	☐
Parametric input	☐
Climatic loads	☑
Mobile loads	☐
Dynamics	☐
Stability	☑
Nonlinearity	☑
Structural model	☑
IFC properties	☐
Prestressing	☐
Bridge design	☐
Excel checks	☐
Document	☑

Nonlinearity	
Beam local nonlinearity	☑
Support nonlinearity/basic s...	☑
Initial imperfections	☑
Geometrical nonlinearity	☐
General plasticity	☐
Cables	☐
Friction support/Soil spring	☐
Sequential analysis	☐
Subsoil	
Pile Design [NEN method]	☐
Pad foundation check	☐
Steel	
Plastic hinge analysis	☐
Fire resistance checks	☐
Steel connections	☐
Scaffolding	☐

OK | Annuler

We go to document and add the requested results.



Tree

Main | Document

DOC - Défaut

Défaut
- Internal forces on member (Efforts internes
- Displacement of nodes (Déplacement des

New | Close

New document item

Résultats
- Déformée de la structure
- Déplacements 3D
- Contraintes 3D
- Efforts internes des barres
- Déformations des barres
- Déplacement des noeuds
- Accélération aux noeuds
- Réactions
- Résultante des réactions
- Résultante spatiale des réactions nodales
- Intensité
- Contrainte dans une barre
- Cisaillement joint
- Déformation relative
- Métré
- Efforts assemblage
- Descente de charges
- Déplacement des noeuds - maillage
- Efforts internes - surfaces
- Contraintes - surfaces
- Contraintes de contact
- Sol de fondation - Paramètres C
- Sol de fondation - Autres données
- Protocole de calcul
- Fréquences propres
- Multiplicateurs de charge critiques
- Sismique détaillé
- Contrainte/déformation
- Rigidité
- Résultats par étage (ensemble)
- Résultats par étage (détaillés)
- Erreur num., Raffinement maillage

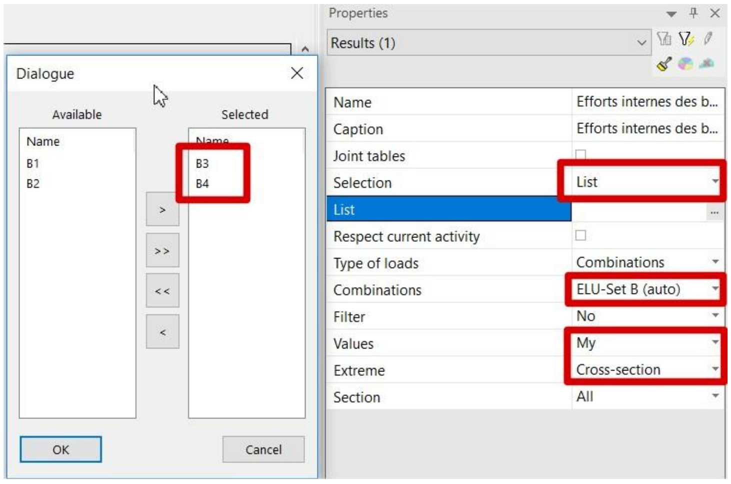
Special

Résultats v17

<<< Add | Close

Then we set for each one of the results the list of elements and the combination we want to export.

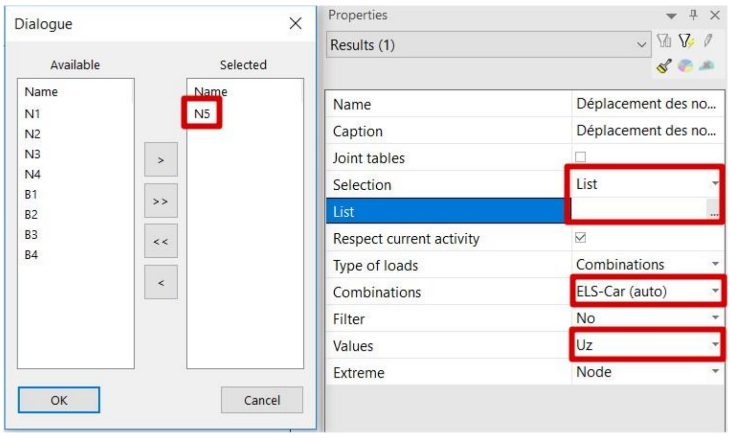
For the moment My:



Available	Selected
Name	Name
B1	B3
B2	B4

Name	Efforts internes des b...
Caption	Efforts internes des b...
Joint tables	☐
Selection	List
List	
Respect current activity	☐
Type of loads	Combinations
Combinations	ELU-Set B (auto)
Filter	No
Values	My
Extreme	Cross-section
Section	All

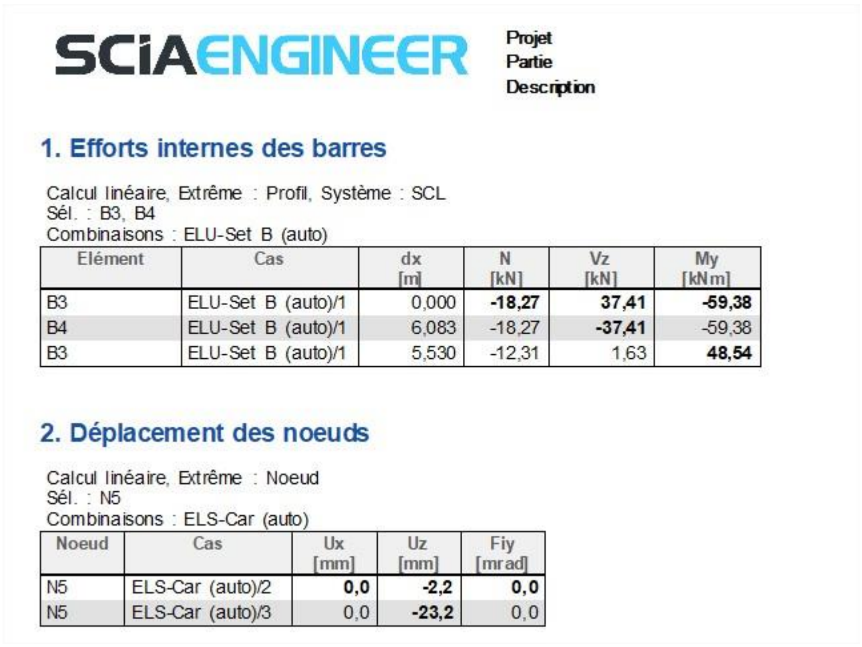
For the displacement Uz:



Available	Selected
Name	Name
N1	N5
N2	
N3	
N4	
B1	
B2	
B3	
B4	

Name	Déplacement des no...
Caption	Déplacement des no...
Joint tables	☐
Selection	List
List	
Respect current activity	☑
Type of loads	Combinations
Combinations	ELS-Car (auto)
Filter	No
Values	Uz
Extreme	Node

Final document:



SCIAENGINEER

Projet
Partie
Description

1. Efforts internes des barres

Calcul linéaire, Extrême : Profil, Système : SCL
Sél. : B3, B4
Combinaisons : ELU-Set B (auto)

Elément	Cas	dx [m]	N [kN]	Vz [kN]	My [kNm]
B3	ELU-Set B (auto)/1	0,000	**-18,27**	**37,41**	**-59,38**
B4	ELU-Set B (auto)/1	6,083	-18,27	**-37,41**	-59,38
B3	ELU-Set B (auto)/1	5,530	-12,31	1,63	**48,54**

2. Déplacement des noeuds

Calcul linéaire, Extrême : Noeud
Sél. : N5
Combinaisons : ELS-Car (auto)

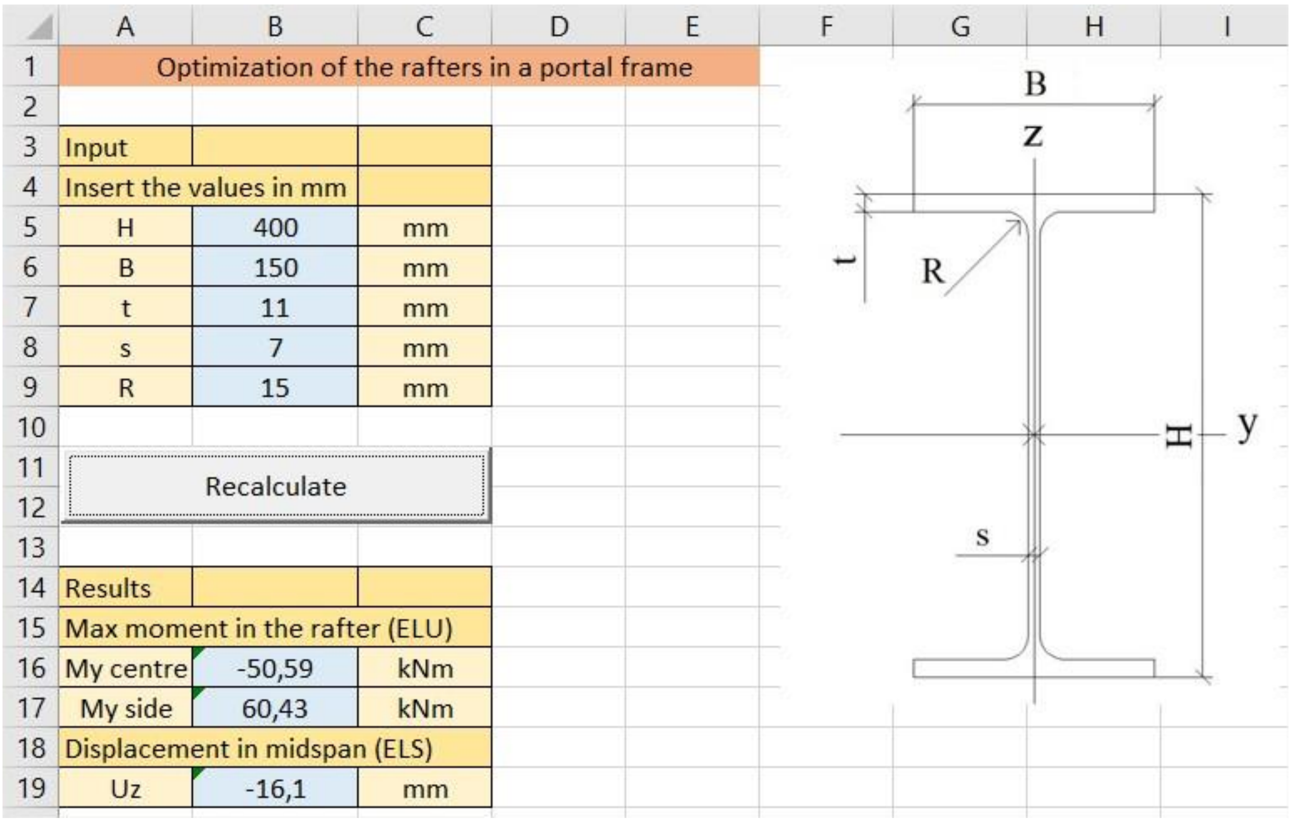
Noeud	Cas	Ux [mm]	Uz [mm]	Fiy [mrad]
N5	ELS-Car (auto)/2	**0,0**	**-2,2**	**0,0**
N5	ELS-Car (auto)/3	0,0	**-23,2**	0,0

Then we can start to create our script. The example below can be used (please control and modify the path of the folders following their location in the machine):

```
Private Sub Recalculate_Click()
' Open XML file for writing'
Dim fs, f
Set fs = CreateObject("Scripting.FileSystemObject")
Set f = fs.CreateTextFile("C:\Users\michele\Desktop\XML_tutorial\XMLex.xml", True)
' we assume that the data are stored in folder C:\Users\michele\Desktop\ XML_tutorial
' Generate XML file using the two input values
Dim SomethingToWrite As Boolean
SomethingToWrite = True
Dim mystring As String
Dim i As Integer
i = 1
Do
mystring = Worksheets("XML").Cells(i, 2).Value
If mystring <> "" Then
If Worksheets("XML").Cells(i, 1).Value = "H" Then
' this line stores the value of the diameter'
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(5, 2).Value / 1000) & """/>"
End If
If Worksheets("XML").Cells(i, 1).Value = "B" Then
' this line storesthe value of the thickness'
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(6, 2).Value / 1000) & """/>"
End If
If Worksheets("XML").Cells(i, 1).Value = "t" Then
' this line storesthe value of the thickness'
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(7, 2).Value / 1000) & """/>"
End If
If Worksheets("XML").Cells(i, 1).Value = "s" Then
' this line storesthe value of the thickness'
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(8, 2).Value / 1000) & """/>"
End If
If Worksheets("XML").Cells(i, 1).Value = "R" Then
' this line storesthe value of the thickness'
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(9, 2).Value / 1000) & """/>"
End If
f.WriteLine (mystring)
i = i + 1
Else
SomethingToWrite = False
End If
Loop While SomethingToWrite = True
f.Close
Shell "C:\Program Files (x86)\SCIA\Engineer18.1\Esa_XML.exe LIN
C:\Users\michele\Desktop\XML_tutorial\Esa1.esa C:\Users\michele\Desktop\XML_tutorial\XMLex.xml /tTXT
/oC:\Users\michele\Desktop\XML_tutorial\Results.xls"
'Assumptions:
'SCIA Engineer installed in folder C:\Program Files (x86)\SCIA\Engineer18.1
' project file stored in folder C:\Users\michele\Desktop\XML_tutorial
' XML file generated to folder C:\Users\michele\Desktop\XML_tutorial
'Note:
'File Esa_XML.exe is an integral part of SCIA Engineer installation
' Parameters of "SHELL" command
' C:\Program Files (x86)\SCIA\Engineer18.1\Esa_XML.exe = location of Esa_XML.exe
' LIN= linear calculation
' C:\Users\michele\Desktop\XML_tutorial\Esa1.esa = original project file
' C:\Users\michele\Desktop\XML_tutorial\XMLex.xml = XML file for the update function
' /tTXT /oC:\Users\michele\Desktop\XML_tutorial\Results.xls = export the document stored inside the project
file as TXT file with location and name C:\Users\michele\Desktop\XML_tutorial\Results.xls
Application.Wait (Now + TimeValue("0:00:15"))
```

```
'wait in order to guarantee that SCIA Engineer completes the calculation before the control of the operation is returned to the VBA Script
'Read result into XLS sheet C:\Users\michele\Desktop\XML_tutorial\Results.xls'
' Insert the result value into the appropriate cell of the "Table" sheet
Dim momentcentre As Variant
Dim momentside As Variant
Dim deformation As Variant
Workbooks.Open Filename:="C:\Users\michele\Desktop\XML_tutorial\Results.xls"
momentcentre = Sheets("Results").Cells(14, 6).Value
momentside = Sheets("Results").Cells(16, 6).Value
deformation = Sheets("Results").Cells(36, 4).Value
'The Cell (14,6) of the "document file" contains the required moment My in the centre
'The Cell (16,6) of the "document file" contains the required moment My on the side
'The Cell (36, 4) of the "document file" contains the required vertical displacement
'to find out this, you have to open the generated XLS file manually (only once) and note down the cell coordinates
Workbooks("Results.xls").Close
Worksheets("Table").Cells(16, 2).Value = momentcentre
Worksheets("Table").Cells(17, 2).Value = momentside
Worksheets("Table").Cells(19, 2).Value = deformation
End Sub
```

The result will be the following:



	A	B	C
1	Optimization of the rafters in a portal frame		
2			
3	Input		
4	Insert the values in mm		
5	H	400	mm
6	B	150	mm
7	t	11	mm
8	s	7	mm
9	R	15	mm
10			
11	Recalculate		
12			
13			
14	Results		
15	Max moment in the rafter (ELU)		
16	My centre	-50,59	kNm
17	My side	60,43	kNm
18	Displacement in midspan (ELS)		
19	Uz	-16,1	mm

With a click we can check easily check the change in the moments and the displacement Uz.

Part 5: Advanced features

5.1 Start command to run Batch analysis

In this chapter we explain how to run calculation with the Start command. This is command run in the Command Prompt environment, so we will use the Excel VBA to call this command, which will then run in the background.

The Start command allows us to use the string /WAIT to set the time for the calculation of our application.

Creation of the application .cmd

The first step in this case is to create the application (called XMLtutorial.cmd in this case). We open an empty txt file and we write for example the following:

```
START "XMLtutorial" /WAIT "C:\Program Files (x86)\SCIA\Engineer18.1\ESA_XML.exe" LIN
"C:\Users\michele\Desktop\XML_tutorial\Esa1.esa" "C:\Users\michele\Desktop\XML_tutorial\XMLex.xml"
/tTXT /o"C:\Users\michele\Desktop\XML_tutorial\Results.xls"
```

Notes:

- C:\Program Files (x86)\SCIA\Engineer18.1\Esa_XML.exe = location of Esa_XML.exe
- C:\Users\michele\Desktop\XML_tutorial\Esa1.esa = original project file
- XML file generated to folder C:\Users\michele\Desktop\XML_tutorial
- /tTXT /oC:\Users\michele\Desktop\XML_tutorial\Results.xls = export the document stored inside the project file as TXT file with location and name C:\Users\michele\Desktop\XML_tutorial\Results.xls (it is an Excel file, it could be also C:\Users\michele\Desktop\XML_tutorial\Results.txt

We save it as XMLtutorial.txt and then we change the format to .cmd .

First Macro to call the application XMLtutorial.cmd

The following step is to create a first Macro Command, (which we can call Recalculate as for previous example) which script could be the following (please control and modify the path of the folders following their location in the machine):

```
Private Sub Recalculate_Click()
Dim fs, f
Set fs = CreateObject("Scripting.FileSystemObject")
Set f = fs.CreateTextFile("C:\Users\michele\Desktop\XML_tutorial\XMLex.xml", True)
Dim SomethingToWrite As Boolean
SomethingToWrite = True
Dim mystring As String
Dim i As Integer
i = 1
Do
mystring = Worksheets("XML").Cells(i, 2).Value
If mystring <> "" Then
If Worksheets("XML").Cells(i, 1).Value = "H" Then
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(5, 2).Value / 1000) & """/>"
End If
If Worksheets("XML").Cells(i, 1).Value = "B" Then
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(6, 2).Value / 1000) & """/>"
End If
If Worksheets("XML").Cells(i, 1).Value = "t" Then
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(7, 2).Value / 1000) & """/>"
End If
If Worksheets("XML").Cells(i, 1).Value = "s" Then
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(8, 2).Value / 1000) & """/>"
```

```
End If
If Worksheets("XML").Cells(i, 1).Value = "R" Then
mystring = " <p4 v=""" & Str(Worksheets("Table").Cells(9, 2).Value / 1000) & """/>"
End If
f.WriteLine (mystring)
i = i + 1
Else
SomethingToWrite = False
End If
Loop While SomethingToWrite = True
f.Close
Shell "C:\Users\michele\Desktop\XML_tutorial\XMLtutorial.cmd"
End Sub
```

The command in this case will be run and even if the Command will be ended, in the background the Start apllication will continue until the command is complete.

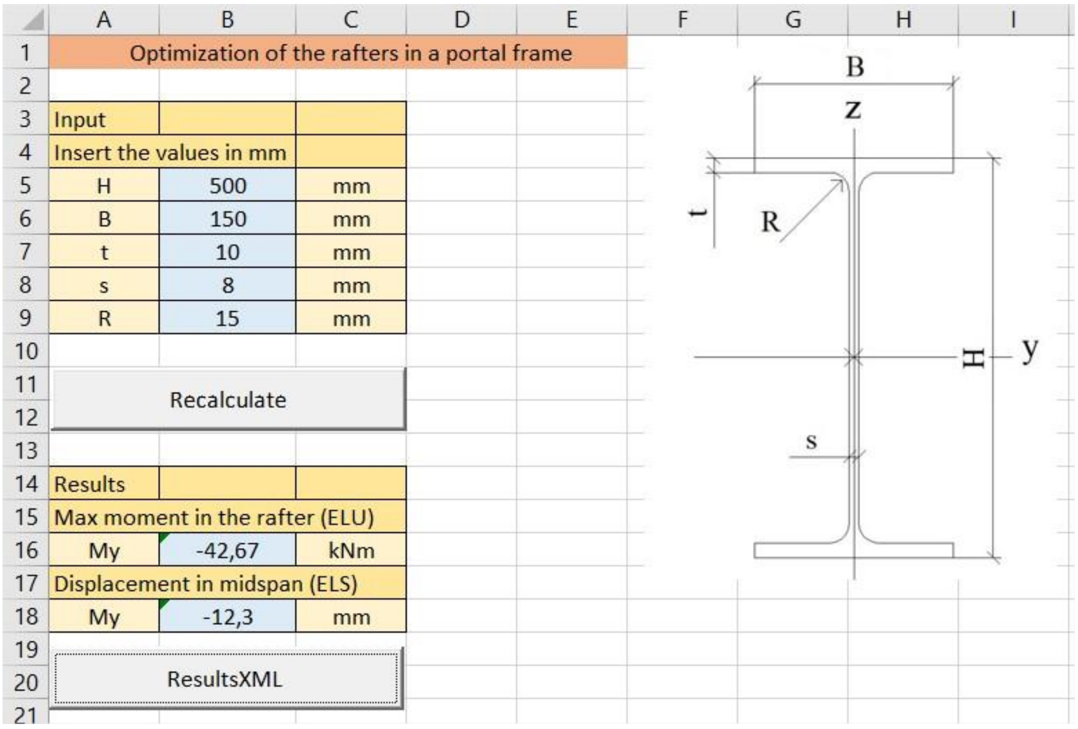
When the command is complete, we will see in the folder C:\Users\michele\Desktop\XML_tutorial the files XMLex.xml and Results.xls appear.

Second Macro to export the Results

The Second Macro to be created, for example ResultsXML, can have the following script:

```
Private Sub ResultsXML_Click()
Dim deformation As Variant
Dim moment As Variant
Workbooks.Open Filename:="C:\Users\michele\Desktop\XML_tutorial\Results.xls"
deformation = Sheets("Results").Cells(36, 4).Value
moment = Sheets("Results").Cells(15, 6).Value
Workbooks("Results.xls").Close
Worksheets("Table").Cells(18, 2).Value = deformation
Worksheets("Table").Cells(16, 2).Value = moment
End Sub
```

In this case, the Results file will be read and the different values reported in the Excel file.



	A	B	C
1	Optimization of the rafters in a portal frame		
2			
3	Input		
4	Insert the values in mm		
5	H	500	mm
6	B	150	mm
7	t	10	mm
8	s	8	mm
9	R	15	mm
10			
11	Recalculate		
12			
13			
14	Results		
15	Max moment in the rafter (ELU)		
16	My	-42,67	kNm
17	Displacement in midspan (ELS)		
18	My	-12,3	mm
19			
20	ResultsXML		
21			

This system will allow us to run important calculation without the need to set the correct value of the time waiting, as we did in the previous case.

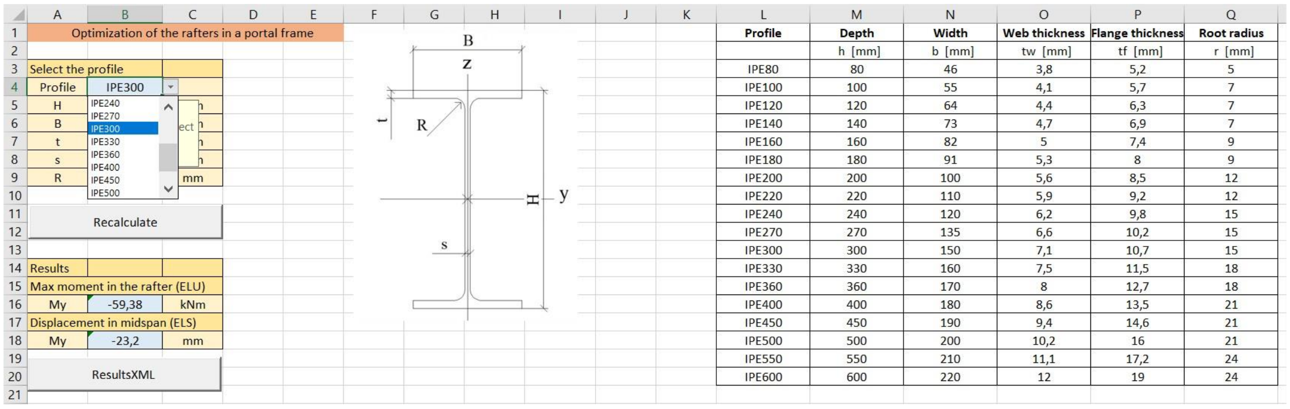
The start command allows us also to run batch calculation: For example, we can write in the .cmd file:

```
START "XMLtutorial" /WAIT "C:\Program Files (x86)\SCIA\Engineer18.1\ESA_XML.exe" LIN
"C:\Users\michele\Desktop\XML_tutorial\Esa1.esa" "C:\Users\michele\Desktop\XML_tutorial\XMLex.xml"
/tTXT /o"C:\Users\michele\Desktop\XML_tutorial\ResultsLIN.xls"

START "XMLtutorial" /WAIT "C:\Program Files (x86)\SCIA\Engineer18.1\ESA_XML.exe" NEL
"C:\Users\michele\Desktop\XML_tutorial\Esa1.esa" "C:\Users\michele\Desktop\XML_tutorial\XMLex.xml"
/tTXT /o"C:\Users\michele\Desktop\XML_tutorial\ResultsNEL.xls"
```

This will run both calculation and export the results in two different files.

5.2 Complete the Excel sheet with the profile's library

We complete now our spreadsheet by coping a Profile's Library of IPE in the Excel sheet. We can then create a drop-down list with the profile and connect directly all the dimensions to the profile selected.



Optimization of the rafters in a portal frame

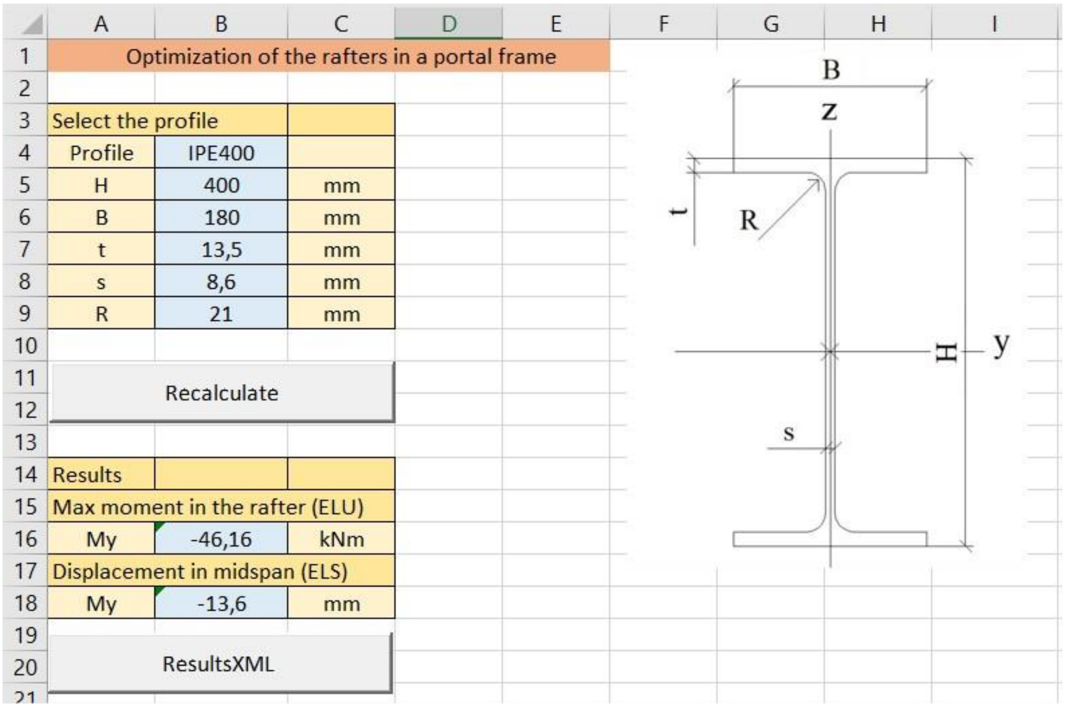
Select the profile		
Profile	IPE300	
H		
B		
t		
s		
R		mm

IPE240
IPE270
IPE300
IPE330
IPE360
IPE400
IPE450
IPE500

Results		
Max moment in the rafter (ELU)		
My	-59,38	kNm
Displacement in midspan (ELS)		
My	-23,2	mm

Profile	Depth	Width	Web thickness	Flange thickness	Root radius
	h [mm]	b [mm]	tw [mm]	tf [mm]	r [mm]
IPE80	80	46	3,8	5,2	5
IPE100	100	55	4,1	5,7	7
IPE120	120	64	4,4	6,3	7
IPE140	140	73	4,7	6,9	7
IPE160	160	82	5	7,4	9
IPE180	180	91	5,3	8	9
IPE200	200	100	5,6	8,5	12
IPE220	220	110	5,9	9,2	12
IPE240	240	120	6,2	9,8	15
IPE270	270	135	6,6	10,2	15
IPE300	300	150	7,1	10,7	15
IPE330	330	160	7,5	11,5	18
IPE360	360	170	8	12,7	18
IPE400	400	180	8,6	13,5	21
IPE450	450	190	9,4	14,6	21
IPE500	500	200	10,2	16	21
IPE550	550	210	11,1	17,2	24
IPE600	600	220	12	19	24

Now once we chose a profile, all the dimensions will be automatically updated in the Spreadsheet.
We chose for example an IPE400: we run "Recalculate" and "ResultsXML" and we get the final result.



Optimization of the rafters in a portal frame

Select the profile		
Profile	IPE400	
H	400	mm
B	180	mm
t	13,5	mm
s	8,6	mm
R	21	mm

Results		
Max moment in the rafter (ELU)		
My	-46,16	kNm
Displacement in midspan (ELS)		
My	-13,6	mm