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

Open Checks in SCIA Engineer

Open Checks with SCIA Design Forms

Contacts

Belgium Headquarters	France
SCIA nv Industrieweg 1007 B-3540 Herk-de-Stad Tel: +32 13 55 17 75 E-mail: info@scia.net Support Phone CAE (SCIA Engineer) Tel: +32 13 55 09 90 CAD (Allplan) Tel: +32 13 55 09 80 Support E-mail: support@scia.net	SCIA France sarl Centre d'Affaires 29, Grand' Rue FR-59100 Roubaix Tel.: +33 3.28.33.28.67 Fax: +33 3.28.33.28.69 E-mail: france@scia.net Agence commerciale 8, Place des vins de france FR-75012 Paris Tel.: +33 3.28.33.28.67 Fax: +33 3.28.33.28.69 E-mail: france@scia.net
Brazil	**USA**
SCIA do Brasil Software Ltda Rua Dr. Luiz Migliano, 1986 - sala 702 , CEP SP 05711-001 São Paulo Tel.: +55 11 4314-5880 E-mail: brasil@scia.net	SCIA North America 7150 Riverwood Drive 21046 Columbia, MD Tel.; +1 443-542-0638 Fax.:+1 410-290-8050 E-mail: usa@scia.net
Netherlands	**Switzerland**
SCIA Nederland B.V. Wassenaarweg 40 NL-6843 NW ARNHEM Tel.:+31 26 320 12 30 Fax.: +31 26 320 12 39 E-mail: info@scia.net	SCIA Swiss Office Dürenbergstrasse 24 CH-3212 Gurmels Tel.: +41 26 341 74 11 Fax: +41 26 341 74 13 E-mail: info@scia.ch
Czech Republic	**Slovakia**
SCIA CZ s.r.o. Praha Evropská 2591/33d 160 00 Praha 6 Tel.: +420 226 205 600 Fax: +420 226 201 673 E-mail: info.praha@scia.cz SCIA CZ s.r.o. Brno Slavickova 827/1a 638 00 Brno Tel.: +420 530 501 570 Fax: +420 226 201 673 E-mail: info.brno@scia.cz	SCIA SK s.r.o. Murgašova 1298/16 SK-010 01 Žilina Tel.: +421 415 003 070 Fax: +421 415 003 072 E-mail: info@scia.sk

Austria	*Germany*
SCIA Datenservice Ges.m.b.H.	SCIA Software GmbH
Dresdnerstrasse 68/2/6/9	Technologie Zentrum Dortmund, Emil-Figge-Strasse 76-80
A-1200 WIEN	D-44227 Dortmund
Tel.: +43 1 7433232-11	Tel.: +49 231/9742586
Fax: +43 1 7433232-20	Fax: +49 231/9742587
E-mail: info@scia.at	E-mail: info@scia.de
Support	
Tel.: +43 1 7433232-12	
E-mail: support@scia.net	

Document created: 18 / 07 / 2017

SCIA Engineer 17.0

Version & Protection Information

Welcome to the Manual for Open Checks: Link with SCIA Design Forms.

This document illustrates the link between SCIA Engineer and SCIA Design Forms by means of worked out examples.

Version info

Document Title	Manual – Open Checks: Link with SCIA Design Forms	
Release	14.0	
Revision	04/2014	

Protection info

In order to use the link with SCIA Design Forms, the following module needs to be activated in the License:

- ESA.33 - Integrated Design Tools

The examples provided in this Manual have been created using the following versions:

- SCIA Engineer Release 14 version 14.0.76
- SCIA Design Forms Release 5.1 version 5.1.21.0

Introduction

Within this manual, guided examples and information is given on the use of Open Checks: Link with SCIA Design Forms.

In SCIA Engineer, a large amount of advanced checks are available for a 1D member: Concrete Reinforcement Design, Steel Code Checks, Aluminium Design, Timber Design …

It is off course possible that a user would like a special kind of check, something which is not currently implemented in SCIA Engineer.

This is where the link with SCIA Design Forms comes up: using this module, the user can define his/her own type of check within SCIA Design Forms and link this to SCIA Engineer. During the check, the input data from SCIA Engineer (like internal forces, materials, cross-section data …) are sent to the Design Form and the results are read back. The fully detailed output of the Form can even be displayed directly within SCIA Engineer.

In general, the linking of an Open Check involves the following steps:

- Step 1: Put the .CLS file in the 'User' folder
- Step 2: Set the ESA ID for each variable which needs to be linked
- Step 3: Optionally define Member data
- Step 4: Define the Check Header
- Step 5: Export the Form to a .CLC
- Step 6: Import the .CLC into the Check Manager
- Step 7: Optionally edit the Check Header and provide icons
- Step 8: Execute the Check
- Step 9: Evaluate the Results
- Step 10: Optionally review the DataCache and Trace file

In this manual, the above procedure for the Open Checks is illustrated using guided examples.

Example 1	In the first example the general principles of the link with SCIA Design Forms are explained. A simplified Steel Bending Check is used as a practical example to illustrate the workings of the module.
Example 2	In the second example, the use of Member data is illustrated. For this purpose, the Form of the first example is extended with additional data which will be accessed and modified through the Member data.
Example 3	In the third example, the usage of Libraries is illustrated with emphasis on the difference between a Library in standalone and acquiring the same data when linked to SCIA Engineer. As a practical example a simplified Concrete calculation is used which determines the amount of tension reinforcement needed for a rectangular section.

Together with this document, the following files are provided:

- Manual_Example_1.cls
- Manual_Example_1.esa
- Manual_Example_1_Icon.bmp
- Manual_Example_2.cls
- Manual_Example_2_Icon.bmp
- Manual_Example_2_MemberData_Icon.bmp
- Manual_Example_2.esa
- Manual_Example_3.cls
- Manual_Example_3.esa

These files can be downloaded here: Download

Within each Example it is specified where to place which file and when to open it.

The Annexes of this manual contain additional information for the Advanced user.

Annex A gives background information on the way how data is transferred between SCIA Engineer and SCIA Design Forms as well as additional Command line switches which can be used with SCIA Engineer to get more in depth info and perform troubleshooting.

Annex B provides a basic set of ESA ID's which are used to link data between SCIA Engineer and SCIA Design Forms. The second part of this Annex gives some more info on the use of shortcuts to refer to ESA ID's.

Annex C provides solutions to common issues which might occur during the run of a SCIA Design Forms Check from SCIA Engineer.

Default folders and file system

This chapter provides information about the folders used for Open Checks within SCIA Engineer as well as the structure of the different files used for Open Checks.

Default folders

For Open Checks within SCIA Engineer, two different folders are used by default: a 'System folder' and a 'User folder'. In general these folders can be written as follows:

- System Folder: **"C:\Program Files (x86)\SCIA\EngineerXXXX.X\OpenChecks\"**

The first directory refers to the installation folder of SCIA Engineer. This folder contains Checks which are set as "System Checks" and provided by SCIA. This folder is fixed and cannot be changed. When installing a new version of SCIA Engineer the content of this folder can be completely changed or removed.

- User Folder: **"C:\Users\#USER#\Documents\ESAXX.X\OpenChecks\"**

The second folder is created in the user's profile of Windows. This folder is created when starting SCIA Engineer for the first time and contains all Checks created by the user.

This folder can be modified from within SCIA Engineer through Setup > Options > Directories.

The logic of having two separate folders is the following: Any checks created by the user are stored in the 'User folder'. This folder remains untouched when installing a new version of SCIA Engineer. Data of this folder can thus easily be used also in other versions. In addition the path of this folder can be modified to a more convenient location or for example to a common network path etc.

> In order to structure different Checks subfolders can be used, for example a subfolder for Steel Checks, a subfolder for Concrete Checks etc.

Files, Databases and Transfer of checks between users

User-defined Checks

In order to link a user-defined SCIA Design Forms Check with SCIA Engineer the .CLC file of the Check needs to be put in the 'User folder'.

Through the Check Manager (this will be explained in the examples in **Step 6**) this .CLC file can be imported.

> The .CLS files do not have to be put in the 'User folder' however, when exporting a .CLS to a .CLC the .CLC is created in the same location as the .CLS file. Therefore it is convenient to put the .CLS files also in the 'User folder'.

System Checks

System Checks defined by SCIA have their .CLC files located in the 'System folder'.

Database files and transfer between users

The Check Manager within SCIA Engineer concerns a default Library Manager. This means that the standard import and export of **.db4** files is supported.

When adding any Check into the Check Manager, automatically the following .db4 files are created:

CheckManagerUserLib.db4
This file is created in the 'User folder' and contains all user-defined check definitions.

CheckManagerSystemLib.db4
This file is created in the 'System folder' and creates all Check definitions defined by SCIA.

Both database files are loaded automatically when opening a project. This means that any check which is defined is available in any project so the database files provide a 'program level' storage as opposed to a 'project level' storage.

The above also implies that the definition of a Check is not stored with a project. Thus if a project is opened on an installation which does not have the Check defined in its library the Check will not be visible.

In order to transfer a user-defined check to another user or another installation of SCIA Engineer both the .CLC file of the Check as well as the **CheckManagerUserLib.db4** file should be put in the 'User folder' of the other installation.

In case the **CheckManagerUserLib.db4** file already exists on that installation (because that user already created his/her own user-defined Checks) instead of replacing the **.db4** file the new file can be opened with the Check Manager and the relevant checks can be copied. That way they are automatically added into the **CheckManagerUserLib.db4** which already exists.

> This works in exactly the same way as for any other .db4 files, for example for Materials, Bolts, ... Reference is made to the SCIA Engineer WebHelp for more background information on Libraries.

Illustration 1: Transfer a Check to a user who doesn't have own defined Checks

The following example illustrates the transfer of Checks between two users/installations where the second user doesn't have any user-defined checks yet.

User A created his own user-defined Check labeled "**MyCheckA**".

Within his 'User folder' he has the file **MyCheckA.CLC** as well as the **CheckManagerUserLib.db4** containing the definition of his Check.

User B doesn't have any user-defined Checks.

User A now wants to send his Check to **User B** so he sends his **MyCheckA.CLC** and **CheckManagerUserLib.db4** files to **User B**.

User B puts both files in his 'User folder'.

When **User B** now uses SCIA Engineer, he will see the "**MyCheckA**" in the tree.

Illustration 2: Transfer a Check to a user who has own defined Checks

The following example illustrates the transfer of Checks between two users/installations where the second user already has user-defined checks.

User A created his own user-defined Check labeled "**MyCheckA**".

Within his 'User folder' he has the file **MyCheckA.CLC** as well as the **CheckManagerUserLib.db4** containing the definition of his Check.

User B created his own user-defined Check labeled "**MyCheckB**".

Within his 'User folder' he has the file **MyCheckB.CLC** as well as the **CheckManagerUserLib.db4** containing the definition of his Check.

User A now wants to send his Check to **User B** so he sends his **MyCheckA.CLC** and **CheckManagerUserLib.db4** files to **User B**.

User B puts the **MyCheckA.CLC** file in his 'User folder'.

Since he already has a **CheckManagerUserLib.db4** file in his 'User folder' he will not overwrite the existing one because that way he would lose his definition of his own "**MyCheckB**".

He therefore puts the **CheckManagerUserLib.db4** from **User A** in some temporary folder, for example on the Desktop.

User B then opens SCIA Engineer, goes to the Check Manager and uses the 'Read from file' command to open the **CheckManagerUserLib.db4** from **User A**. He then copies the definition of "**MyCheckA**" into his own database.

The new content of his database will now be saved automatically in his own **CheckManagerUserLib.db4** located in his user folder and it contains both the definition of "**MyCheckA**" and the definition of "**MyCheckB**".

When **User B** now uses SCIA Engineer, he will see the "**MyCheckA**" and "**MyCheckB**" in the tree.

He can now also remove the **CheckManagerUserLib.db4** from **User A** which he put in a temporary folder.

Note that a much more convenient way for **User B** is to directly import the **MyCheckB.CLC** file into the Check Manager, without the use of any **db4** file. This is illustrated in the examples in Step 6.

Example 1: General Principles

In this first example the general principles of the link between SCIA Engineer and SCIA Design Forms are explained.

A simplified Steel Bending Check is used as a practical example to illustrate the workings of the module.

The following pictures show the output of the Design Form and the related script:

Example 1: Steel Bending Check

Section Properties

W_{el} = 14000 mm^3

Internal Forces

Bending moment $M_{y,Ed}$ = 20 kNm

Material Characteristics

f_y = 420 N/mm^2

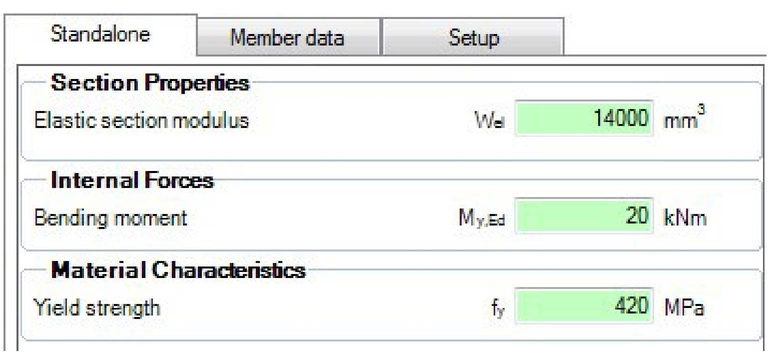
Verification

$M_{Rd} = f_y \cdot W_{el} = 420*10^6 \cdot 14*10^{-6} = 5{,}88$ kNm

$$UC_{Combined} = \frac{abs(M_{y,Ed})}{M_{Rd}} = \frac{abs(20000)}{5880} = 3{,}4$$

```
TEXT("Example 1: Steel Bending Check");

TEXT("Section Properties");
TEXT("W_el = " & VAL(1000000000*W_el, 2) & " mm^3");

TEXT("Internal Forces");
TEXT("Bending moment M_y,Ed = " & VAL(10^-3*M_y,Ed, 2) & " kNm");

TEXT("Material Characteristics");
TEXT("f_y = " & VAL(10^-6*f_y, 2) & " N/mm^2");

TEXT("Verification");
M_Rd = f_y*W_el;

UC_Combined = ABS(M_y,Ed)/M_Rd;
```

The Form uses the following variables as manual input when running standalone:



The purpose of this example will be to link these variables to the data of SCIA Engineer and run this Form from within SCIA Engineer.

As specified in the introduction, the following steps will be followed:

- Step 1: Put the .CLS file in the 'User' folder

Within SCIA Design Forms:

- Step 2: Set the ESA ID for each variable which needs to be linked
- Step 3: Optionally define Member data
- Step 4: Define the Check Header
- Step 5: Export the Form to a .CLC

Within SCIA Engineer:

- Step 6: Import the .CLC into the Check Manager
- Step 7: Optionally edit the Check Header and provide icons
- Step 8: Execute the Check
- Step 9: Evaluate the Results
- Step 10: Optionally review the DataCache and Trace file

Step 1: Put the .CLS file in the 'User Folder'

Before starting, make sure the Form Manual_Example_1.CLS is located in the 'User folder' as outlined in the Chapter on Default folders.

> As indicated in the paragraph for the User-defined Checks, the .CLS file does not have to be put in the 'User folder' however, when exporting a .CLS to a .CLC the .CLC is created in the same location as the .CLS file. Therefore it is convenient to put the .CLS file also in the 'User folder'. This approach will be used for all examples in this Manual

Step 2: Set the ESA ID for each variable which needs to be linked

Upon executing the check from within SCIA Engineer, an export will be made of all relevant model data (cross-sections, materials, internal forces ...) into a so called 'DataCache'.

Each of those data has a specific ESA ID by which it can be identified within the Design Form.

Annex B gives an overview of the ESA ID's for the most basic data available in the 'DataCache'.

In this first step, the ESA ID's for the relevant variables which will be linked are defined in the Form.

Launch the SCIA Design Forms Builder and open the Form **Manual_Example_1.CLS**.

Step 2.1: Set the ID for the Input data

Within this Form, the grid on the 'Calculation' tab shows the double values to be manually inputted in a green color:

Double | String | Boolean | Structured

ID	Description	Symbol	Value	Unit	Precision
	Elastic section modulus	W_{el}	14000	mm^3	2
	Bending moment	$M_{y,Ed}$	20	kNm	2
	Yield strength	f_y	420	MPa	2
	Unity Check	$UC_{Combined}$	3.4		2
	Bending resistance	M_{Rd}	5.88	kNm	2

The first column labeled 'ID' will be used to set the respective ESA ID for these input variables.

Using the basic information from Annex B the following ID's are found to represent the data needed by the form:

ESA ID	Variable
CS.Chars.Wely	Elastic section modulus about the principal y-axis
InternalForces.My	Bending moment about the principal y-axis
Material.EC.fy	Yield strength of the material for Eurocode EC-EN

These ID's are inputted within the grid of the Form:

Double | String | Boolean | Structured

ID	Description	Symbol	Value	Unit
CS.Chars.Wely	Elastic section modulus	W_{el}	14000	mm^3
InternalForces.My	Bending moment	$M_{y,Ed}$	20	kNm
Material.EC.fy	Yield strength	f_y	420	MPa
	Unity Check	$UC_{Combined}$	3.4	
	Bending resistance	M_{Rd}	5.88	kNm

This concludes the first part of this step; all input variables for the Form are now properly linked to ESA.ID's.

Step 2.2: Set the ID for the Output (Result) data

In the same way as the input variables were linked also output variables are defined within the Form.

Output variables typically represent result values which the Form will send back to SCIA Engineer.

Output variables are defined using the ID **Result.X** where **X** represents a number from 1 to 'n'.

In this example there is one result value which will be defined, the Unity Check value $\mathbf{UC}_{Combined}$. This variable will thus be given the ID **Result.1**.

Double | String | Boolean | Structured

ID	Description	Symbol	Value	Unit
CS.Chars.Wely	Elastic section modulus	W_{el}	14000	mm^3
InternalForces.My	Bending moment	$M_{y,Ed}$	20	kNm
Material.EC.fy	Yield strength	f_y	420	MPa
Result.1	Unity Check	$UC_{Combined}$	3.4	
	Bending resistance	M_{Rd}	5.88	kNm

> The first output variable, defined using the ID Result.1 is also the variable which will be used by SCIA Engineer during the AutoDesign process. When defining multiple output variables it is thus advised to set the 'overall' unity check as the first result value.

In order to let the Form take into account the ID's defined in this Step refresh the Form either by pressing the button Refresh or the F5 key.

This concludes the definition of the in- and output variables.

Step 3: Optionally define Member data

In this example no Member data is defined, this feature is explained in Example 2.

Step 4: Define the Check Header

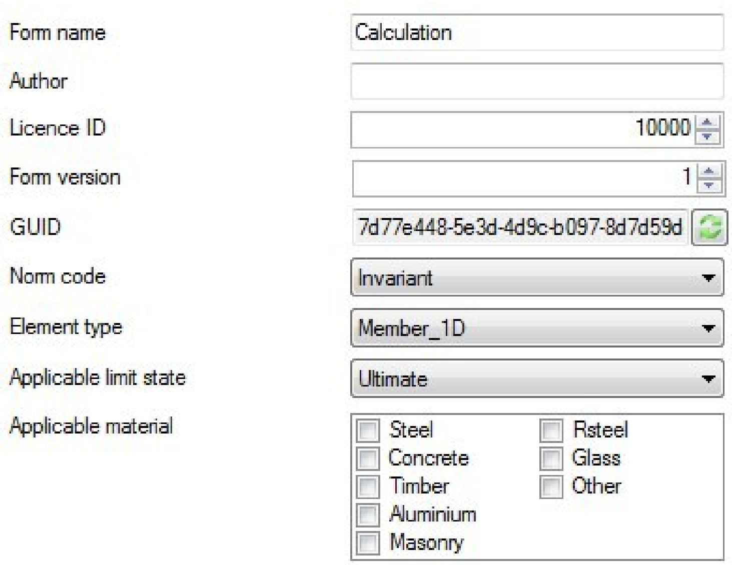
After setting the ESA.ID's and optionally inputting Member data the 'Check Header' can be defined.

Switch to the tab **'Header'** to define the Check Header.

The Check Header is divided into two separate parts: the first part allows the definition of the check name, specify for which material and codes it is valid etc. The second part shows the previously defined Result values and provides output settings for these values.

Step 4.1 Define the Check Header Part 1

When initially opening the **'Header'** tab the following data is shown by default:



The **Form name** shows the name by which this check will be shown within the Tree of SCIA Engineer.

For this example, the name is changed to **Bending Check**.

The **Author** field is used to specify the name of the author of this Form. For this example the field is left blank.

The **License ID** is used to set the license number which is needed to use this Form. For this example the License is left on the default public license number 10000.

The **Form version** can be used by the author to specify the version of the Form. This is convenient when making updates to the Form for example, so the version can be increased accordingly. In this example the version is kept on 1.

The **GUID** field shows the Globally Unique IDentifier of the Form. The GUID is a unique reference number which is used to differentiate between Forms. No two Forms can have the same GUID. The button can be used to generate a new GUID if needed. For this example the default generated GUID is kept.

The **Norm code** field is used to indicate the default code to which this Check applies. Typically Checks are code dependent and thus have one specific code which can be indicated. In case the code selection is of no importance for a Form this field can be set to 'Invariant'. Within SCIA Engineer, a Check will only be visible in case the specified code is selected. For this example the code is set to **EuroCode**.

Since the EuroCode in turn has multiple National Annexes an additional selection appears for the National Annex. In this example no selection of the National Annex is made.

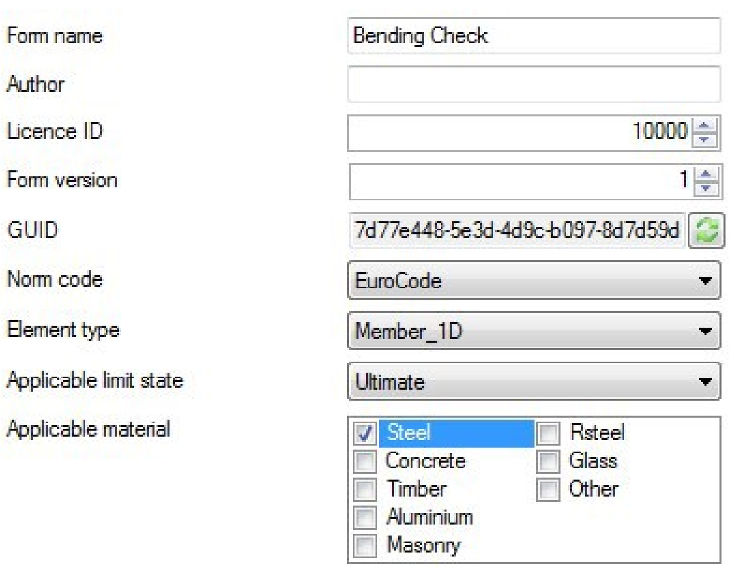
The **Element type** is used to indicate the type of element to which this check applies. A check can either be a 0D Check (a check on nodal elements like a Punching Check, Pad Foundation Check, Connection Check, ...), a 1D Check (a check on beam/column elements like a Steel Beam Check, Concrete Column Check, Timber Girder Check, ...) or a 2D Check (a check on slab/wall elements like a Concrete Slab Check, Timber Wall Check, ...). Since this example concerns a Steel Bending Check for beam elements the type is left on the default **Member 1D**.

The **Applicable limit state** field is used to filter the type of results which will be available for this check. Within SCIA Engineer, the type of combinations in the Check service are filtered based on this setting. When this setting is set to 'Ultimate' only Ultimate combinations are visible in the Check Service. When this setting is set to 'Serviceability' only Serviceability combinations are visible in the Check Service. Using 'Both' any type of combination is shown. For this example, since it concerns a Steel Bending Check the type is left on the default **Ultimate**.

The **Applicable material** setting is used to indicate the material(s) to which this check applies. Only elements with all of the selected materials will be used for the check. Since this example concerns a **Steel** Check only the Steel material is activated.

> In case for example a Composite Steel-Concrete check is defined, both the Steel and Concrete materials should be activated. In that case only those members which have cross-sections with BOTH Steel & Concrete will be used for the check.

After the above modifications the Check Header definition looks as follows:



As a summary of the above settings:

- The Check will be visible only in **EuroCode**

- The Check service will show only **Ultimate** combinations

- The Check is executed only on **1D Members** which have cross-sections with material **Steel**

Step 4.2 Define the Check Header Part 2

The bottom part of the Check Header definition shows a grid which automatically lists all variables that were defined as output results in Step 2.2.

ESA_ID	Variable name	ESA unit	Extreme	Hide zero values	Reverse axis	Draw direction	Drawing limit - minimum	Drawing limit - maximum	Units
Result.1	$UC_{Combined}$	Undefined	Maximum			Z	0,25	1	

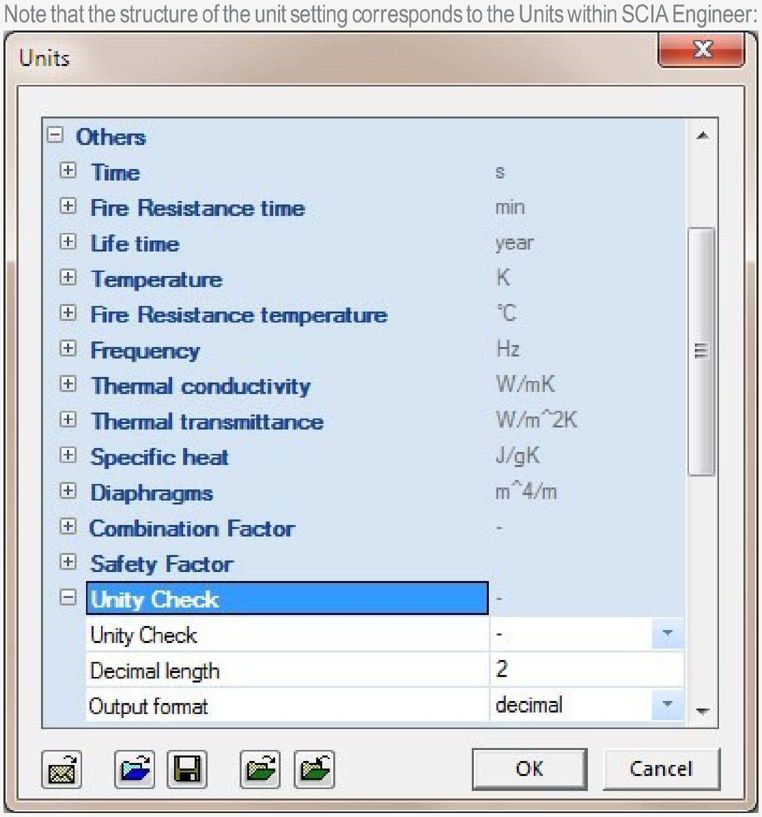
In this example the **$UC_{Combined}$** was set as **Result.1** so this is the only variable shown here.

For each output variable different settings can now be defined which are used by the **Brief** output in SCIA Engineer.

The ESA unit is used to link the variable to a specific unit setting of SCIA Engineer. This way, when unit settings within SCIA Engineer are modified, the Brief output of the Open Check is modified accordingly. A typical example is the unit change of [m] to [mm] or the change of the number of decimal places.

In this example, the **$UC_{Combined}$** variable concerns a unity check. The unit is therefore set to **Others - Unity Check [-]**.

Note that the structure of the unit setting corresponds to the Units within SCIA Engineer:



The **Extreme** setting is used to indicate what the extreme is for this variable

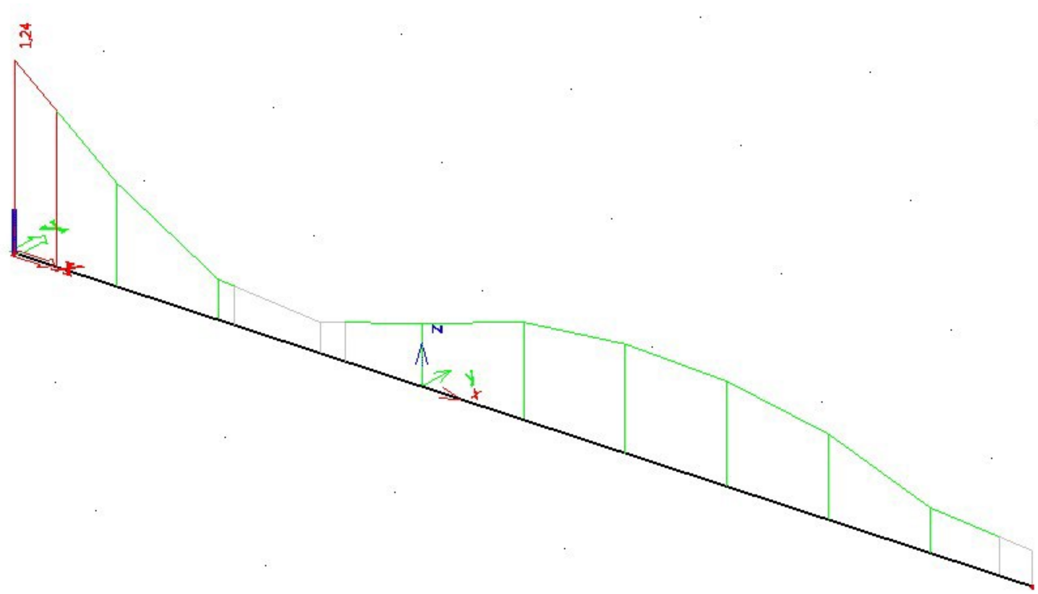
- 'Maximum' signifies that the maximal positive value is searched as an extreme.
- 'Minimum' signifies that the minimal negative value is searched as an extreme.

- 'Both' signifies that both of the above two settings are used.

Since in this example the output variable concerns a unity check the extreme is left on the default **Maximum**.

The **Hide zero values** checkbox is used to indicate that this variable will not be printed in the Brief output in case it is zero for all sections in the output table. This can be used when multiple output values are presented of which several are zero in specific cases, in order not to overload the output with many zeros. In this example the checkbox will remain **unchecked**.

Within SCIA Engineer, Check results are also graphically shown on the screen. The following picture shows an example of such a graphical result:



As the picture shows, for a typical check the result is drawn in the Positive direction of the z-axis of the beam. This can be defined in the Check Header using the **Draw direction** and **Reverse axis** settings. With a **Draw Direction** of Z the result will be plotted in the positive Z-direction. In case the **Reverse axis** checkbox is checked a positive result will be plotted in the negative Z-direction.

For this example, since it concerns a standard check, the defaults are kept so the **Draw Direction** is left on **Z** and the **Reverse axis** checkbox remains **unchecked**.

The above picture also shows that the color of the graphical result changes based on its value. Typically within SCIA Engineer, a unity check below a limit of 0,25 is insignificant and shown in grey. Any unity check above 1,00 is shown in red and anything in between is shown in green.

Within the Check Header these limits can be set in the **Drawing limit - minimum** and **Drawing limit - maximum** fields. In this example the defaults of **0.25** and **1.00** are kept.

The final column **Units** shows the unit of the output variable which was inputted for it within the Form. Since in this Form no unit was inputted for **$UC_{Combined}$** nothing is shown here.

After the above modifications the second part of the Check Header definition looks as follows:

ESA_ID	Variable name	ESA unit	Extreme	Hide zero values	Reverse axis	Draw direction	Drawing limit - minimum	Drawing limit - maximum	Units
Result.1	$UC_{Combined}$	Others - Unity Check [-]	Maximum	☐	☐	Z	0,25	1	

This concludes the definition of the Check Header; all required inputs have now been made.

Step 5: Export the Form to a .CLC

The final step within the Design Form concerns saving the changes made to the source file (.CLS) and exporting this into a .CLC file.

To this end, first save the Form using the Save button.

Next, export the file to a .CLC using the Export CLC button.

This .CLC file has the same name as the .CLS file and will be generated in the same folder as the .CLS file.

As specified in the chapter on Default folders, this file must be in the 'User folder' in order for it to be visible from within SCIA Engineer.

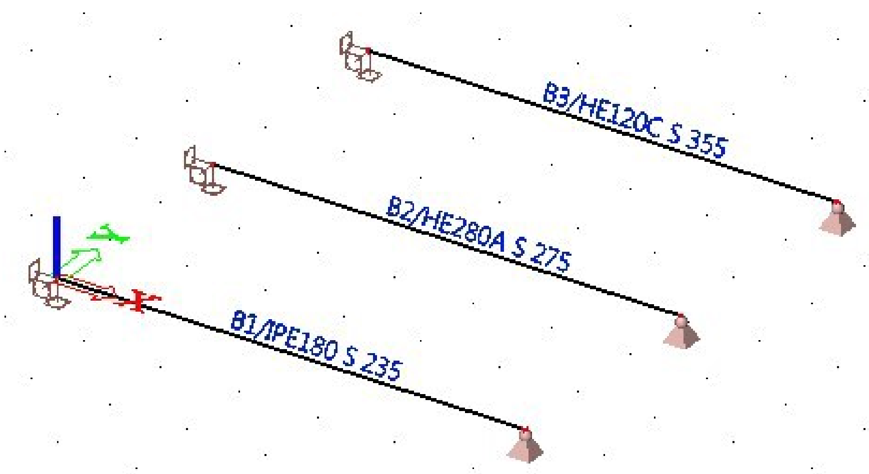
This concludes all steps needed within SCIA Design Forms. The SDF application can now be closed.

Step 6: Import the .CLC into the Check Manager

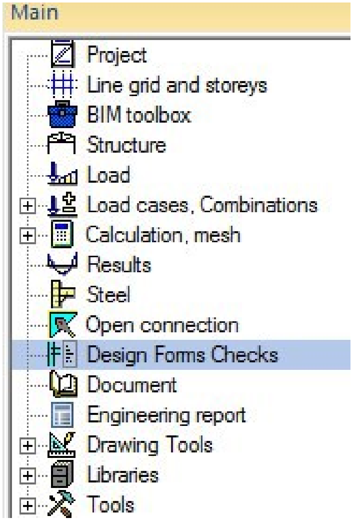
In this specific step, the .CLC file created in the previous Step will be imported within SCIA Engineer.

Launch SCIA Engineer and open the file **Manual_Example_1.esa**.

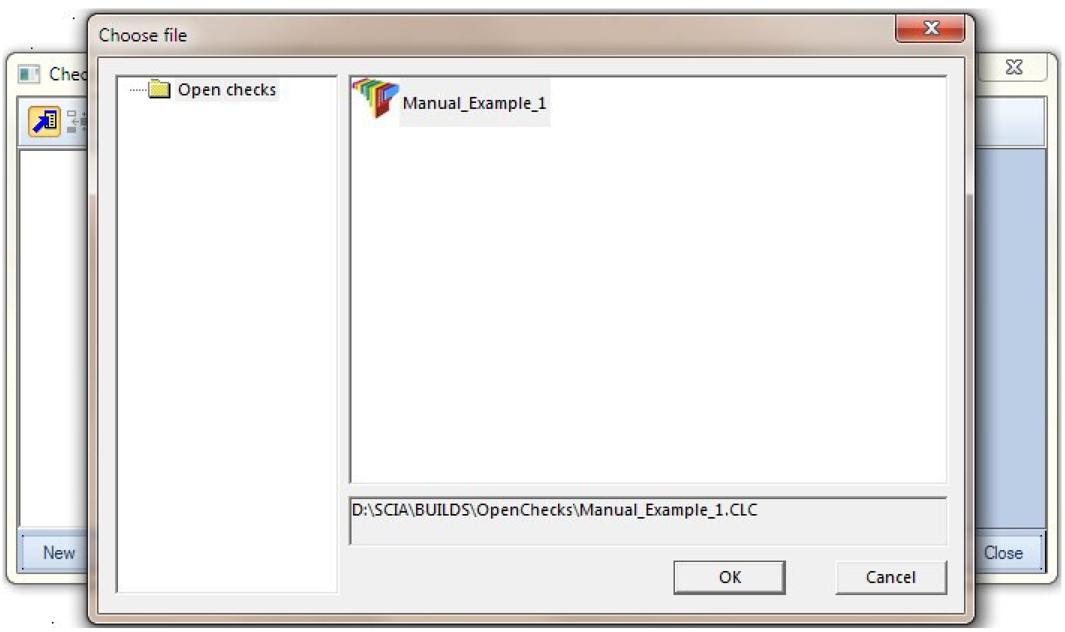
Within this example project, three steel members with different cross-sections and materials are given:



Go to the Design Forms Checks service.



In the Design Forms Checks service open the Check manager .



When opening the **Check Manager** for the first time, it will automatically open a dialog for choosing a .CLC file.

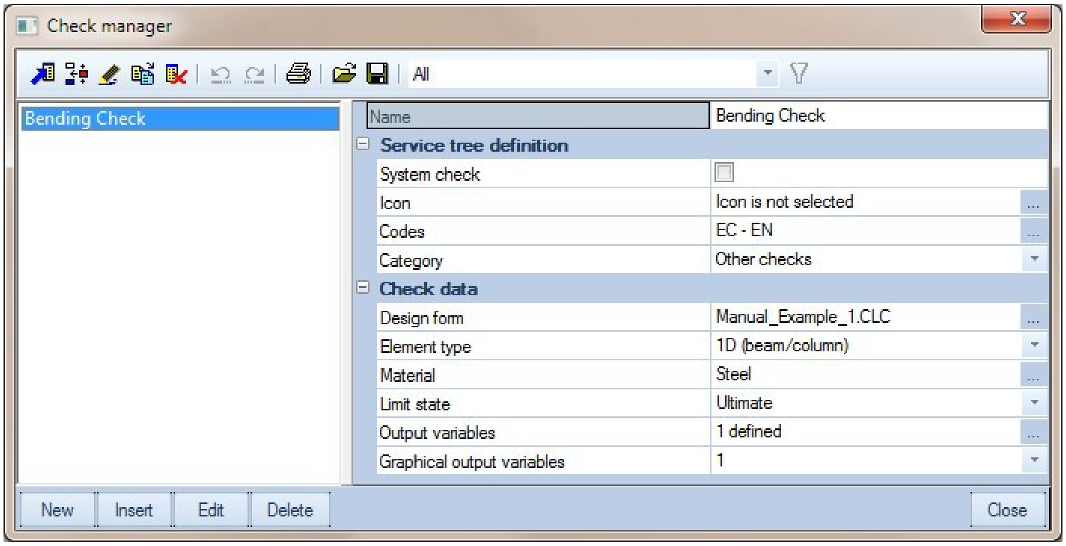
The dialog points to the OpenChecks 'User folder' as defined in the Default folders chapter.

Select the **Manual_Example_1.CLC** file and press [OK].

As specified in the Default folders chapter, when having multiple CLC files it is possible to structure them using subfolders, for example for different types of checks, materials etc.

In case the folder contains also an .ICO icon file with the same name as the .CLC file this icon will be displayed in the above dialog.

The **Check Manager** automatically imports the Check header information from the CLC:



The **Name**, **Codes**, **Element type**, **Material**, **Limit state** and **Output variables** are shown as defined in the CLC in Step 4.

> The Check Manager concerns a global repository i.e. it is common for the whole SCIA Engineer installation and thus not for this specific project.
>
> In other words, the Check which was just imported here will be visible in any project opened in this installation of SCIA Engineer.

Step 7: Optionally edit the Check Header and provide icons

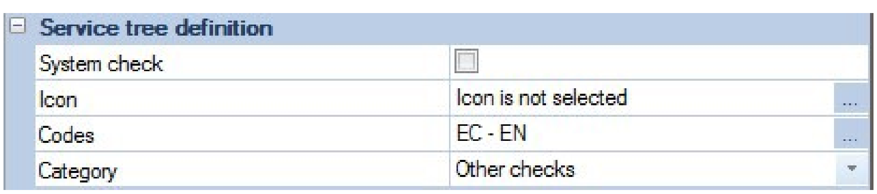
In essence, merely importing the CLC is already sufficient since all required data has been defined.

It is however possible to add additional info to the definition of the check or edit the data which was defined in the Form.

In this step the different settings found in the Check Manager will be reviewed.

Step 7.1: Service tree definition

The **Service tree definition** shows the info which will be used to display the check in the tree of SCIA Engineer.



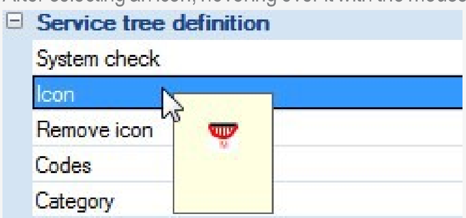
The **System check** checkbox is used to indicate if this is a user-defined check or a system check defined by SCIA. In case this checkbox is checked, SCIA Engineer will look for the .CLC file in the System folder as specified in the Default folders chapter.

Since this example concerns a user check, this checkbox is left **unchecked**.

Note that this checkbox is only visible for dealer licenses, a standard user will always create a user check automatically.

The **Icon** setting can be used to give the check a customized icon in the tree. This icon should be a **bmp** of **16x15 pixels**. For this example, press the ... button and browse for the file **Manual_Example_1_Icon.bmp**

After selecting an icon, hovering over it with the mouse will show the icon in a tooltip



The **Codes** setting shows the code as specified in the Check Header in Step 4. The check will only be visible in SCIA Engineer in case the code of the project matches one of the codes listed here.

This setting can be used to make a check also available for other codes. For this example, no change is made and the setting is left on **EC-EN**.

The **Category** is used to specify the subgroup in which the check will be placed in the tree. For this example the subgroup is set to **Steel checks**.

After the above modifications the **Service tree definition** looks as follows:

Service tree definition	
System check	☐
Icon	Icon is selected
Remove icon	
Codes	EC - EN
Category	Steel checks

Step 7.2: Check data

The **Check data** subgroup shows all data relevant to the check.

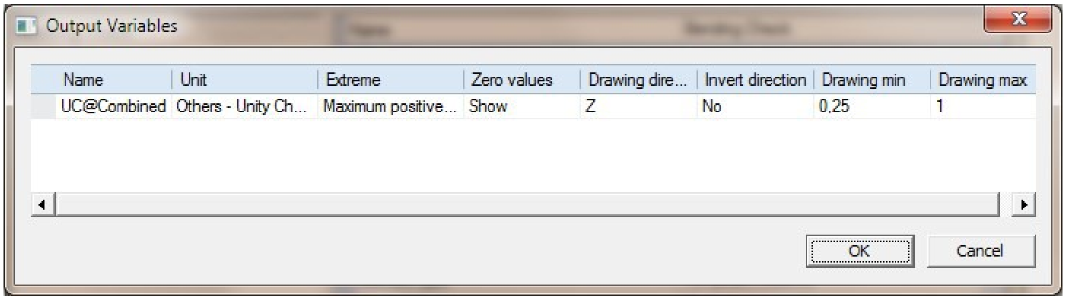
Check data	
Design form	Manual_Example_1.CLC
Element type	1D (beam/column)
Material	Steel
Limit state	Ultimate
Output variables	1 defined
Graphical output variables	1

All of these settings have been discussed in Step 4 during the definition of the Check Header.

For this example, no changes are made to these settings.

The only setting shown here in addition is the one for **Graphical output variables**. By default, all output variables can be selected in the Values field of the Check Service, so all output values can be displayed also graphically. Using this setting only the 'n' first variables can be selected for the graphical output. In this example there is only one output variable so this setting is left on **1**.

To review the **Output variables**, click on the ... button.



All settings shown in this dialog have been defined in Step 4.2 during the definition of the Check Header.

The @ symbol shown in the variable Name UC@Combined is used to indicate a subscript. This will become clear in later Steps where this name will be shown on the output of SCIA Engineer as $UC_{Combined}$.

This dialog can be closed with [OK].

The **Check Manager** can now be closed by pressing [Close].

The **Check Manager** contains an additional subgroup labeled SEN Functionality. This can be used to indicate that for example the AutoDesign action buttons (AutoDesign, Split, Unify) should also be visible for this specific check.

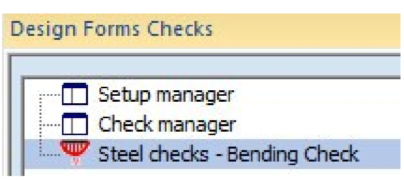
The **Update Definition** action button at the bottom of the **Check Manager** is used to read the data from the .CLC file once more. This can be used in case the .CLC file has been modified.

Step 8: Execute the Check

All preparations have been made; the Check is linked to SCIA Engineer and can now be executed.

Step 8.1: Display the Brief and Graphical output

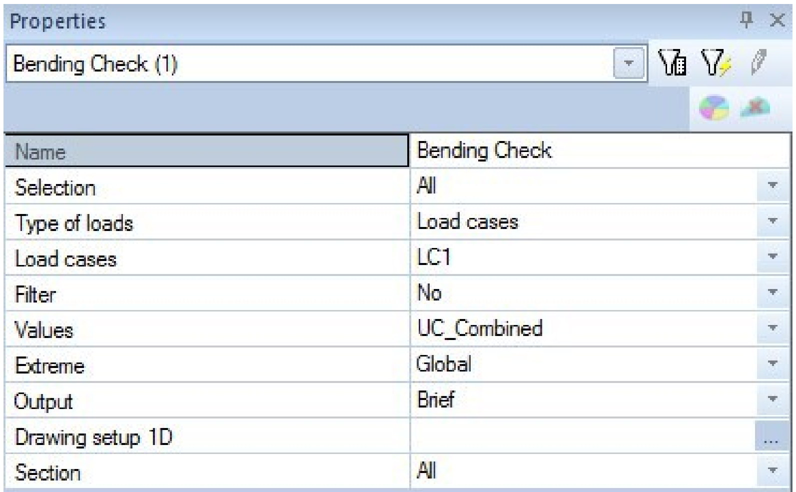
After closing the **Check Manager** the new check will be shown in the **Design Forms Checks** service:



As with all Check services in SCIA Engineer, Checks are only visible in case the project is calculated. So if the new Check is not shown, perform a linear calculation.

The tree shows the subgroup **Steel checks** in which the **Bending check** is listed. Since in this case there is only one item in the **Steel checks** subgroup this item is shown on the same line and its icon is shown at the front of the subgroup. Later on in this manual, when more Checks are added to this subgroup the different Checks will be shown indented.

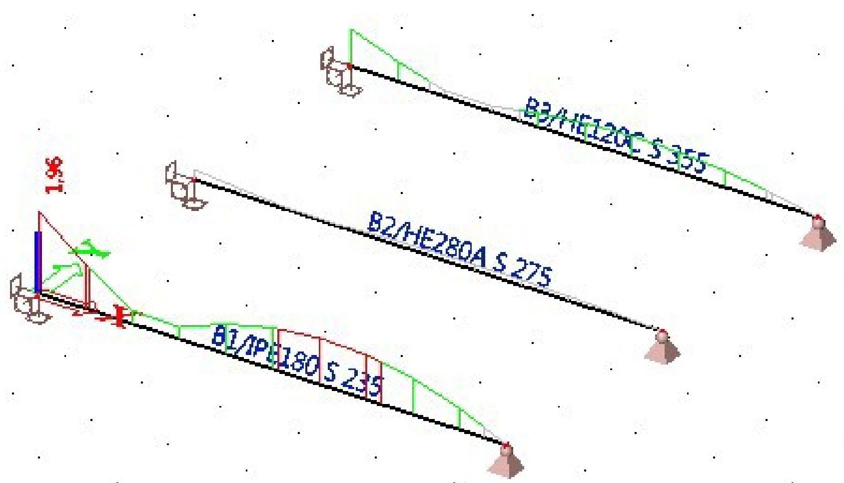
Select the **Steel checks - Bending Check** in the service. The Property window will show the properties of the Check:



This Property window shows the standard properties as known also for other 1D member checks like Steel, Timber, Aluminium,...

The property window shows that the Check will be executed for value **UC_Combined** (note that in this window the subscript was replaced by an underscore) and that the **Brief** output will be shown.

Press the **Refresh** action button to run the check.



First of all it can be seen that the result is graphically shown in the Z-direction of the member axis system, just as it was set in the Check Header. In addition the color scheme is used as defined i.e. with values shown in red for results exceeding 1,00.

In addition each member shows a different check result which is expected since each member has a different cross-section and material.

Press the **Preview** action button to display the preview of the Brief check:

Bending Check

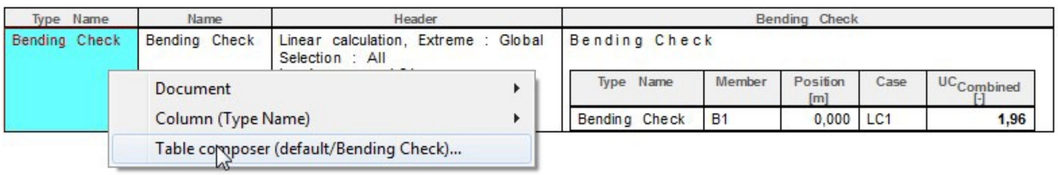
Type Name	Name	Header	Bending Check
Bending Check	Bending Check	Linear calculation, Extreme : Global Selection : All Load cases : LC1	Bending Check Type Name | Member | Position [m] | Case | UCCombined [-] Bending Check | B1 | 0,000 | LC1 | **1,96**

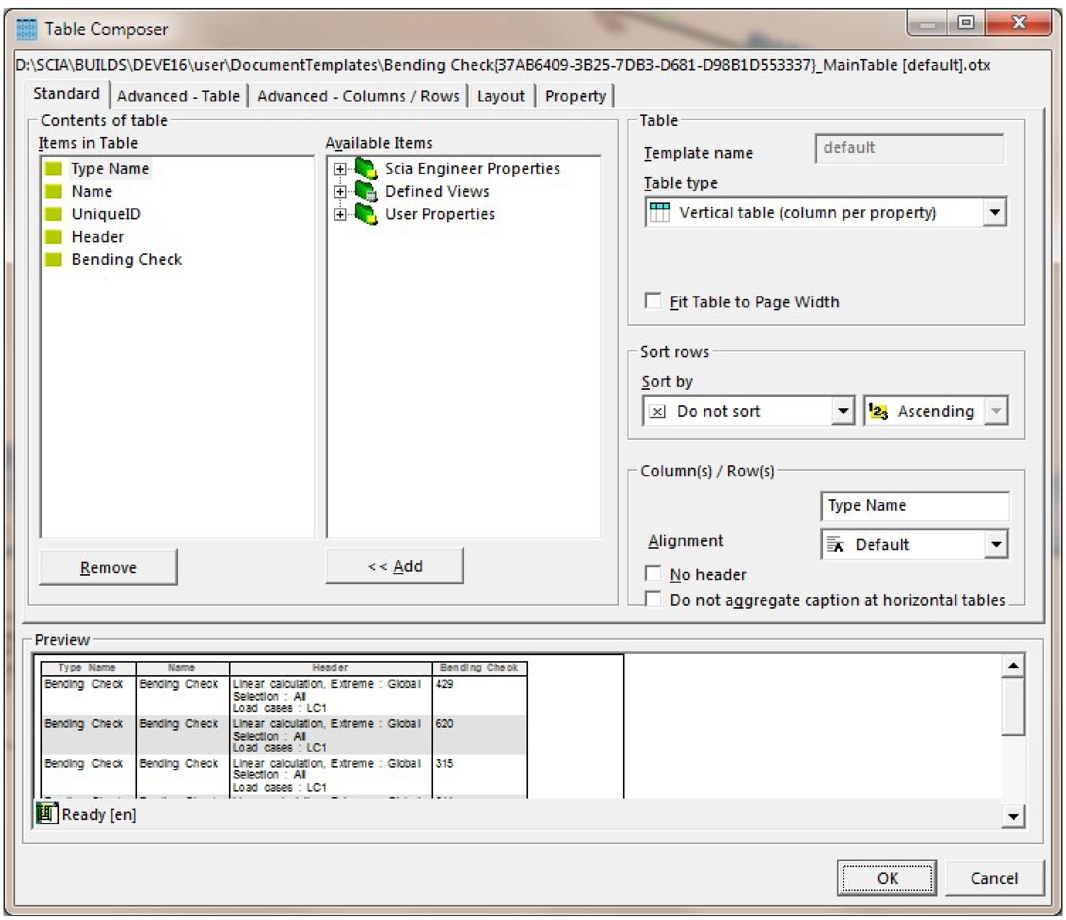
Since this concerns a completely new check, the output is generated using the standard table layout. Using the **Table composer** this can be modified.

Intermezzo: Using the Table composer to edit the Brief output

This Intermezzo illustrates how the output table can be edited through the **Table composer**. For more information on the usage of the **Table composer**, reference is made to the SCIA Engineer WebHelp.

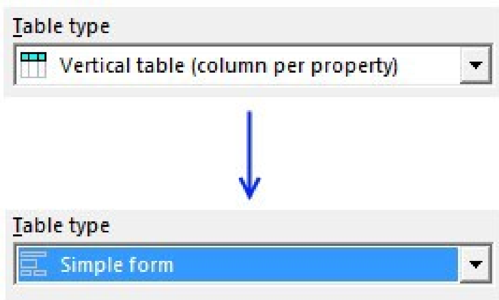
- First right click on the main table, for example in the **Type name** column and select **Table composer** from the context menu.





- In the **Items in Table** column, remove the **Type Name** and **Name**.

- Change the **Table type** from **Vertical table** to **Simple form**.



- Confirm the inputs with [OK].

- Press the **Refresh** action button

Bending Check

Linear calculation, Extreme : Global
Selection : All
Load cases : LC1

Bending Check

Type Name	Member	Position [m]	Case	$UC_{Combined}$ [-]
Bending Check	B1	0,000	LC1	**1,96**

Note the subscript shown here for the $\mathbf{UC}_{Combined}$ variable.

Switching the **Extreme** setting to **Member** and pressing the **Refresh** action button shows the check for all three members:

Bending Check

Linear calculation, Extreme : Member
Selection : All
Load cases : LC1
Bending Check

Type Name	Member	Position [m]	Case	$UC_{Combined}$ [-]
Bending Check	B1	0,000	LC1	**1,96**
Bending Check	B2	0,000	LC1	**0,24**
Bending Check	B3	0,000	LC1	**0,89**

Step 8.2: Display the Detailed output

The previous Step illustrated how the Graphical and Brief output can be displayed. In this step the Detailed output is discussed which shows the rendered output of the Design Form.

In the Property window change the **Extreme** setting back to **Global** and set the **Output** to **Detailed**.

Extreme	Global
Output	Detailed

Press the **Refresh** action button to run the check.

Bending Check

Linear calculation, Extreme : Global
Selection : All
Load cases : LC1

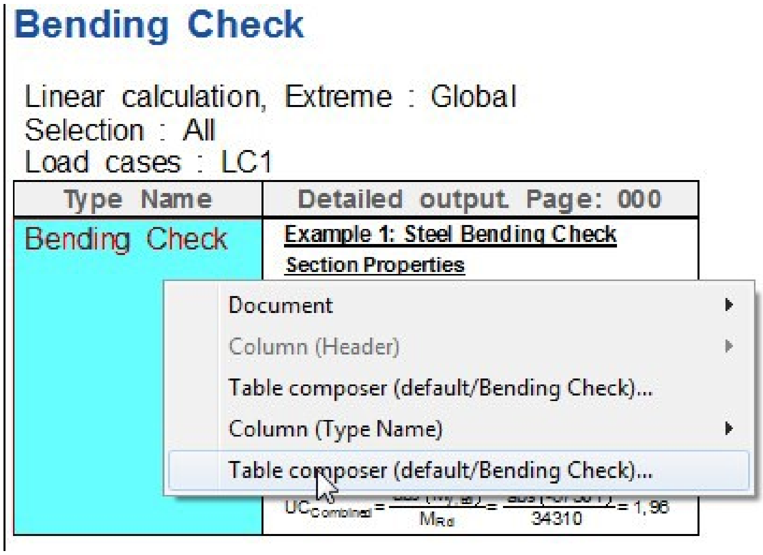
Type Name	Detailed output Page: 000
Bending Check	**Example 1: Steel Bending Check** **Section Properties** W_{el} = 146000 mm³ **Internal Forces** Bending moment $M_{y,Ed}$ = -67,3 kNm **Material Characteristics** f_y = 235 N/mm² **Verification** $M_{Rd} = f_y \cdot W_{el} = 235*10^{6} \cdot 146*10^{-6} = 34{,}3$ kNm $UC_{Combined} = \frac{abs(M_{y,Ed})}{M_{Rd}} = \frac{abs(-67301)}{34310} = 1{,}96$

As indicated above, this new output is again generated using the standard table layout. Using the **Table composer** this can be modified.

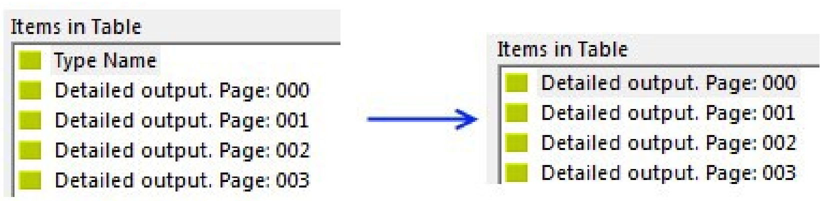
Intermezzo: Using the Table composer to edit the Detailed output

This Intermezzo illustrates how the output table can be edited through the **Table composer**. For more information on the usage of the **Table composer**, reference is made to the SCIA Engineer WebHelp.

- First right click on the main table, for example in the **Type name** column and select **Table composer** at the bottom of the context menu.



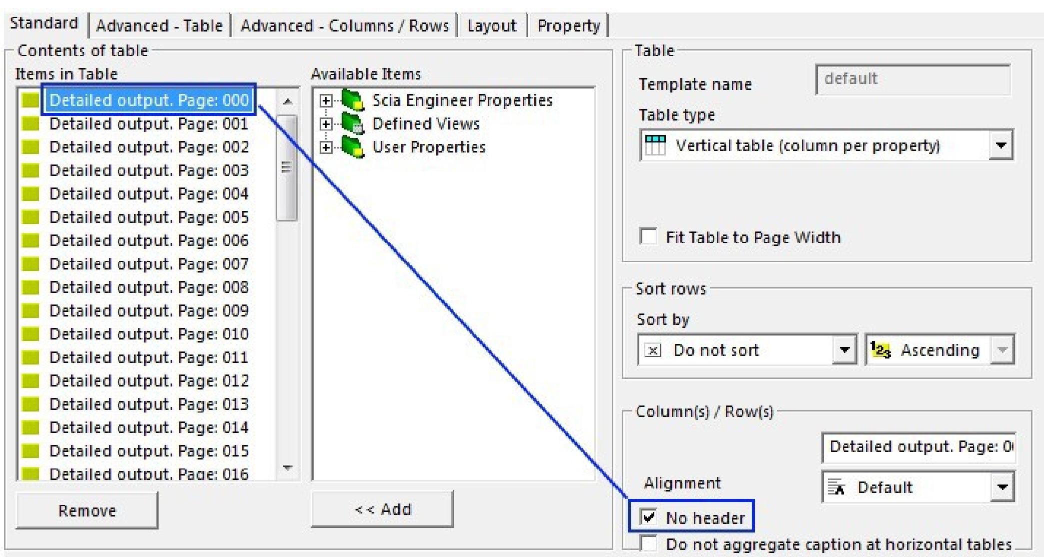
- In the **Items in Table** column, remove the **Type Name**.



The Detailed output for a Design Form Check can sent up to 50 pages of output back to SCIA Engineer.

In this example only the first page is used, so only the **Detailed output. Page: 000** is modified.

- In the **Items in Table** select the **Detailed output. Page: 000** and activate the **No header** checkbox.



- Switch to the tab **Advanced - Columns / Rows**.

- In the **Items in Table** select the **Detailed output. Page: 000** and change the **Width** and **Height** of the **Picture Size** both to **50 mm**.

Example 1: Steel Bending Check

Section Properties

$W_{el} = 146000\ mm^3$

Internal Forces

Bending moment $M_{y,Ed} = -67{,}3\ kNm$

Material Characteristics

$f_y = 235\ N/mm^2$

Verification

$$M_{Rd} = f_y \cdot W_{el} = 235 * 10^6 \cdot 146 * 10^{-6} = 34{,}3\ kNm$$

$$UC_{Combined} = \frac{abs(M_{y,Ed})}{M_{Rd}} = \frac{abs(-67301)}{34310} = 1{,}96$$

The output for example shows the yield strength **f_y** of **235 N/mm²** of this member.

Repeat the procedure for the second member:

Example 1: Steel Bending Check

Section Properties

$W_{el} = 1.01 * 10^6\ mm^3$

Internal Forces

Bending moment $M_{y,Ed} = -66{,}6\ kNm$

Material Characteristics

$f_y = 275\ N/mm^2$

Verification

$$M_{Rd} = f_y \cdot W_{el} = 275 * 10^6 \cdot 1.01 * 10^{-3} = 279\ kNm$$

$$UC_{Combined} = \frac{abs(M_{y,Ed})}{M_{Rd}} = \frac{abs(-66646)}{278575} = 0{,}239$$

The output here shows the yield strength **f_y** of **275 N/mm²** of this member.

Repeat the procedure for the third member:

Example 1: Steel Bending Check

Section Properties

$W_{el} = 213600\ mm^3$

Internal Forces

Bending moment $M_{y,Ed} = -67{,}3\ kNm$

Material Characteristics

$f_y = 355\ N/mm^2$

Verification

$$M_{Rd} = f_y \cdot W_{el} = 355 * 10^6 \cdot 214 * 10^{-6} = 75{,}8\ kNm$$

$$UC_{Combined} = \frac{abs(M_{y,Ed})}{M_{Rd}} = \frac{abs(-67346)}{75828} = 0{,}888$$

The output here shows the yield strength f_y of 355 N/mm² of this member.

This quick overview of the results shows that the export of data was done correctly.

Step 10: Optionally review the DataCache and Trace file

The final step of this example illustrates the internal transfer of data between SCIA Engineer and SCIA Design Forms.

Before going through this step please read the information contained within Annex A which explains the Command line switches and generation of the **.trace** file.

- First of all, close SCIA Engineer. The changes to the project file do not need to be saved.

- Next, launch SCIA Engineer using the Command line switch **-OCFILES**.

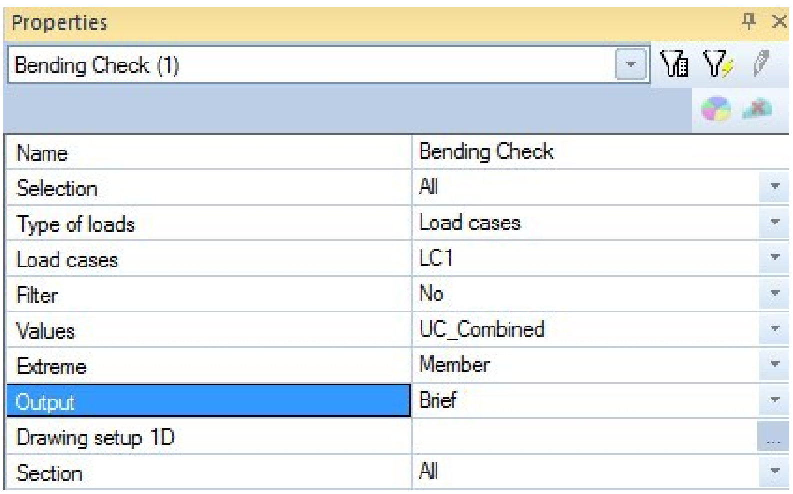
- Open the file **Manual_Example_1.esa**.

- Perform a **Linear** calculation to refresh the results.

- Go to the Design Forms Checks service.

- Select the Steel checks - Bending Check in the service.

- Set the following Properties for the Check:



- Press the **Refresh** action button to run the check

- Using a file explorer like **Windows Explorer**, navigate to the **TEMP** folder of SCIA Engineer and locate the subfolder **ESA_Model_Data**

This subfolder shows a folder for each member which has been checked as well as the different log files as explained in Annex A.

> The thread numbers used for the log files can differ on each run, this is determined by the Operating System.

Now go to the subfolder **Beam.1**. This folder contains all input data (**.emd**), output data (**.results**) and the **.trace** file.

1DMemberInfo	emd
Beam	emd
BucklingData	emd
ConcreteMemberDataNew	emd
CrossSection	emd
Deformations	emd
InternalForces	emd
Load	emd
Materials	emd
MemberInfo	emd
Sections	emd
Beam.1	results
Beam.1	trace

When looking inside the **Materials.emd** file using a standard text editor like **NotePad** the yield strength of 235 N/mm² (2,35 e8 N/m²) of this member can be found.

```
(:Material :Type eSteel :ID 152 :Name "S 235" )
  (:EC :E 2.1e+011 :G 8.07692e+010 :fy 2.35e+008 :fu 3.6e+008 )
```

In the same way for example the **InternalForces.emd** file shows the moment My of -67,3 kNm (-67301,3 Nm) in the begin section of the member:

```
(:Section :ID 0 )
  (:InternalForces :ID 0 :N 0 :Vy 0 :Vz 56216.9 :Mx 0 :My -67301.3 :Mz 0 )
(:Section :ID 1 )
  (:InternalForces :ID 0 :N 0 :Vy 0 :Vz 47216.9 :Mx 0 :My -36271.2 :Mz 0 )
(:Section :ID 2 )
  (:InternalForces :ID 0 :N 0 :Vy 0 :Vz 38216.9 :Mx 0 :My -10641 :Mz 0 )
(:Section :ID 3 )
  (:InternalForces :ID 0 :N 0 :Vy 0 :Vz 29216.9 :Mx 0 :My 9589.1 :Mz 0 )
```

The **.results** file shows the unity check of 1,96 in the begin section of the member:

```
(:Res :Section 0 :Combi 0 :Values 1.961565; )
(:Res :Section 1 :Combi 0 :Values 1.057161; )
(:Res :Section 2 :Combi 0 :Values 310.1428e-3; )
(:Res :Section 3 :Combi 0 :Values 279.4841e-3; )
```

The **.trace** file gives a full overview of the internals of the Design Form during the check. Since the check is run for each section, data for all sections are shown in this file.

The following picture shows the trace for the first section:

```
-------------------------------------------
MEMBER = Beam.1; SECTION = 0; COMBI = 0
-------------------------------------------
Reading values from IO: {
  W↓el← = CS.Chars.Wely = 146*10↑-6←
  M↓y,Ed← = InternalForces.My = -67301
  f↓y← = Material.EC.fy = 235*10↑6←
}
TEXT(Example 1: Steel Bending Check)
TEXT(Section Properties)
TEXT(W↓el← = 146000 mm↑3←)
TEXT(Internal Forces)
TEXT(Bending moment M↓y,Ed← = -67,3013 kNm)
TEXT(Material Characteristics)
TEXT(f↓y← = 235 N/mm↑2←)
TEXT(Verification)
M↓Rd← = 34310
UC↓Combined← = 1.96
```

The first part shows the data which is read from the DataCache as defined in Step 2.

The rest of the trace shows the different lines of the Design Form with filled in data.

This final Step shows how, using the **-OCFILES** switch, it is possible to debug the SCIA Design Forms Check and perform an in depth evaluation of the Check run.

Close SCIA Engineer and discard any changes made.

This concludes the first example which illustrated the General Principles of the Open Checks using SCIA Design Forms.

Example 2: Using Member Data

In this second example the principles of Member Data are explained. Member Data can be assigned to elements within SCIA Engineer and allow the user to directly access the Input Dialog of SCIA Design Forms.

The Design Form of the previous example is therefore extended with additional parameters which can be set by the user of the Form.

The following pictures show the output of the Design Form and the related script:

Example 2: Extended Steel Bending Check

Section Properties

W_{el} = 14000 mm^3

W_{pl} = 15000 mm^3

Internal Forces

Bending moment $M_{y,Ed}$ = 20 kNm

Material Characteristics

f_y = 420 N/mm^2

γ_{M0} = 1,1

Settings

Elastic check only? True

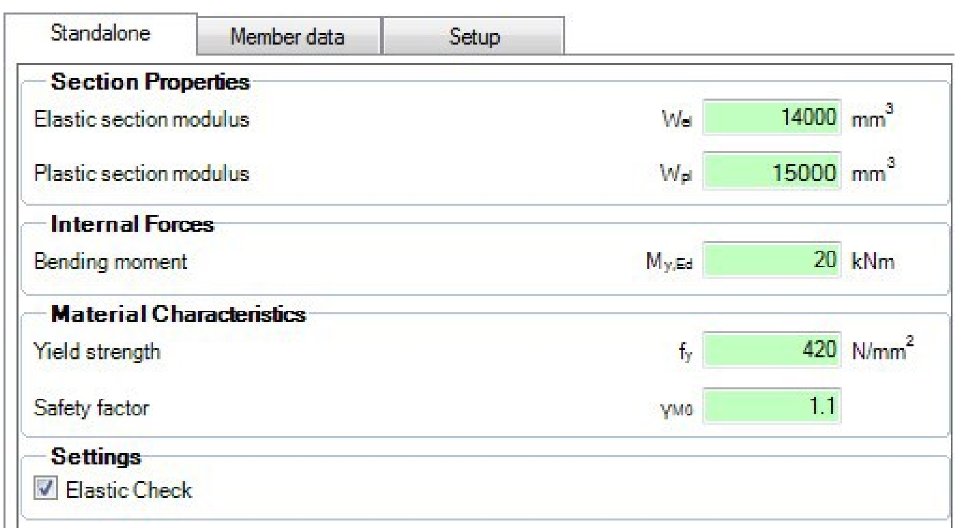
Verification

$$M_{Rd} = \frac{f_y \cdot W_{el}}{\gamma_{M0}} = \frac{420*10^{6} \cdot 14*10^{-6}}{1,1} = 5,35 \text{ kNm}$$

$$UC_{Combined} = \frac{abs(M_{y,Ed})}{M_{Rd}} = \frac{abs(20000)}{5345} = 3,74$$

```
TEXT("Example 2: Extended Steel Bending Check");

TEXT("Section Properties");
TEXT("W_el = " & VAL(1000000000*W_el, 2) & " mm^3");
TEXT("W_pl = " & VAL(1000000000*W_pl, 2) & " mm^3");

TEXT("Internal Forces");
TEXT("Bending moment M_y,Ed = " & VAL(10^-3*M_y,Ed, 2) & " kNm");

TEXT("Material Characteristics");
TEXT("f_y = " & VAL(10^-6*f_y, 2) & " N/mm^2");
TEXT("γ_M0 = " & γ_M0);

TEXT("Settings");
TEXT("Elastic check only? " & Elastic_Check);

TEXT("Verification");
IF(Elastic_Check)
{
    M_Rd = f_y*W_el/γ_M0;
}
ELSE
{
    M_Rd = f_y*W_pl/γ_M0;
}
UC_Combined = ABS(M_y,Ed)/M_Rd;
```

The Form uses the following variables as manual input when running standalone:



The main differences with the previous example are the following:

- In addition to the elastic section modulus $\mathbf{W}_{el}$ also the plastic modulus $\mathbf{W}_{pl}$ is used

- Using a **Checkbox** the selection between the elastic and plastic modulus is made

- The resistance is extended with a safety factor γ_{M0}

The purpose of this example is to illustrate how Member Data can be defined which contains the additional parameters used in this Form (the **Checkbox** and the factor γ_{M0}). In general, Member data allows the user to access any parameters directly through the Design Form dialog.

As specified in the introduction, the following steps will be followed:

- Step 1: Put the .CLS file in the 'User' folder

Within SCIA Design Forms:

- Step 2: Set the ESA ID for each variable which needs to be linked
- Step 3: Optionally define Member data
- Step 4: Define the Check Header
- Step 5: Export the Form to a .CLC

Within SCIA Engineer:

- Step 6: Import the .CLC into the Check Manager
- Step 7: Optionally edit the Check Header and provide icons
- Step 8: Execute the Check
- Step 9: Evaluate the Results
- Step 10: Optionally review the DataCache and Trace file

Step 1: Put the .CLS file in the 'User Folder'

Before starting, make sure the Form **Manual_Example_2.CLS** is located in the 'User folder' as outlined in the Chapter on Default folders.

Step 2: Set the ESA ID for each variable which needs to be linked

As demonstrated in the first example, upon executing the check from within SCIA Engineer, an export will be made of all relevant model data (cross-sections, materials, internal forces ...) to the 'DataCache'.

In this first step, the ESA ID's for the relevant variables which will be linked are defined in the Form.

Launch the SCIA Design Forms Builder and open the Form **Manual_Example_2.CLS**.

Step 2.1: Set the ID for the Input data

Within this Form, the grid on the '**Calculation**' tab shows the double values to be manually inputted in a green color:

Double | String | Boolean | Structured

ID	Description	Symbol	Value	Unit
	Elastic section modulus	W_{el}	14000	mm^3
	Plastic section modulus	W_{pl}	15000	mm^3
	Bending moment	$M_{y,Ed}$	20	kNm
	Yield strength	f_y	420	N/mm^2
	Unity Check	$UC_{Combined}$	3.74	
	Bending resistance	M_{Rd}	5.35	kNm
	Safety factor	γ_{M0}	1.1	

Compared to the first example, the additional parameter which can be obtained through the DataCache is the Plastic section modulus **Wpl**.

The safety factor γ_{M0} however is a property which does not exist in the DataCache, so this will not be linked directly but set later on as Member data.

Using the basic information from Annex B the following ID's are found to represent the data needed by the form:

ESA ID	Variable
CS.Chars.Wely	Elastic section modulus about the principal y-axis
CS.Chars.Wply	Plastic section modulus about the principal y-axis
InternalForces.My	Bending moment about the principal y-axis
Material.EC.fy	Yield strength of the material for Eurocode EC-EN

These ID's are inputted within the grid of the Form:

Double | String | Boolean | Structured

ID	Description	Symbol	Value	Unit
CS.Chars.Wely	Elastic section modulus	W_{el}	14000	mm^3
CS.Chars.Wply	Plastic section modulus	W_{pl}	15000	mm^3
InternalForces.My	Bending moment	$M_{y,Ed}$	20	kNm
Material.EC.fy	Yield strength	f_y	420	N/mm^2
	Unity Check	$UC_{Combined}$	3.74	
	Bending resistance	M_{Rd}	5.35	kNm
	Safety factor	γ_{M0}	1.1	

This concludes the first part of this step; all input variables for the Form which can be obtained from the DataCache are now properly linked to ESA.ID's.

Step 2.2: Set the ID for the Output (Result) data

In this example there is one result value which will be defined, the Unity Check value $UC_{Combined}$. This variable will thus be given the ID **Result.1**.

Double | String | Boolean | Structured

ID	Description	Symbol	Value	Unit	Prec
CS.Chars.Wely	Elastic section modulus	W_{el}	14000	mm^3	2
CS.Chars.Wply	Plastic section modulus	W_{pl}	15000	mm^3	2
InternalForces.My	Bending moment	$M_{y,Ed}$	20	kNm	2
Material.EC.fy	Yield strength	f_y	420	N/mm^2	2
Result.1	Unity Check	$UC_{Combine}$	3.74		2
	Bending resistance	M_{Rd}	5.35	kNm	2
	Safety factor	γ_{M0}	1.1		2

In order to let the Form take into account the ID's defined in this Step refresh the Form either by pressing the button ▶ Refresh or the **F5** key.

This concludes the definition of the in- and output variables.

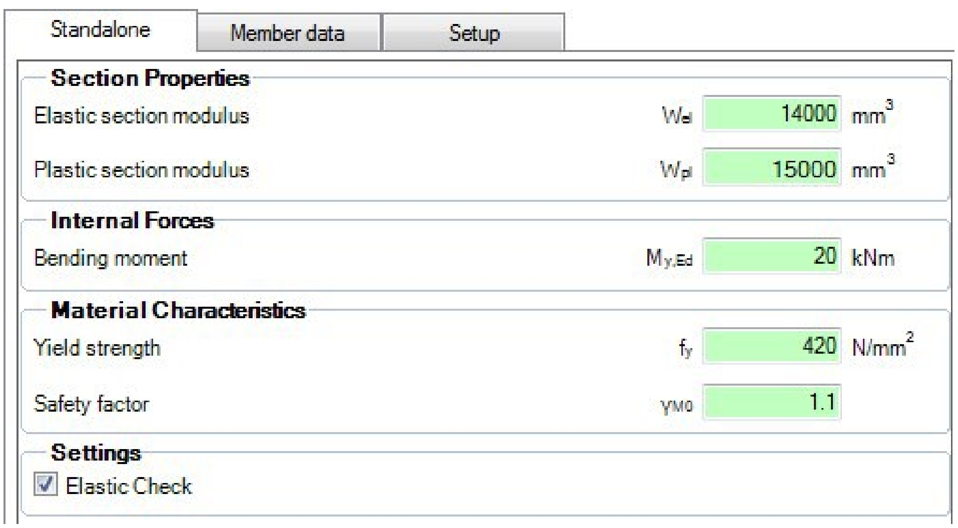
Step 3: Optionally define Member data

As indicated in the introduction of this example, this Form has two additional inputs for the user:

- A **Checkbox** to select either the elastic or plastic modulus

- A safety factor γ_{M0}

Switch to the '**Dialog**' tab to review the current Dialog defined for a Standalone run of the Form:



The Standalone dialog of course contains all inputs since all of them are required for a Standalone run.

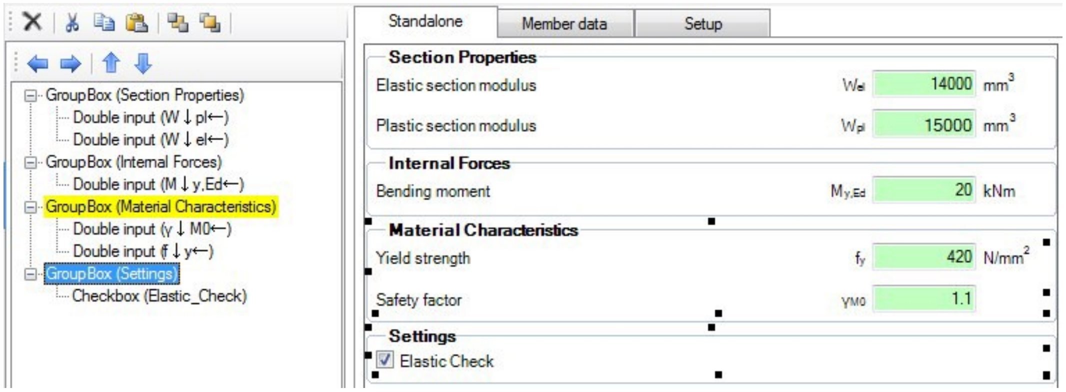
When running the Check from within SCIA Engineer, not all of these inputs should be visible. Therefore the Member data dialog is used: this dialog will be visible from within SCIA Engineer and should thus contain only the relevant information which is not set through the DataCache.

Step 3.1 Define the Member data dialog

In the **Standalone** dialog, switch to the Dialog outline to see all components of the dialog.

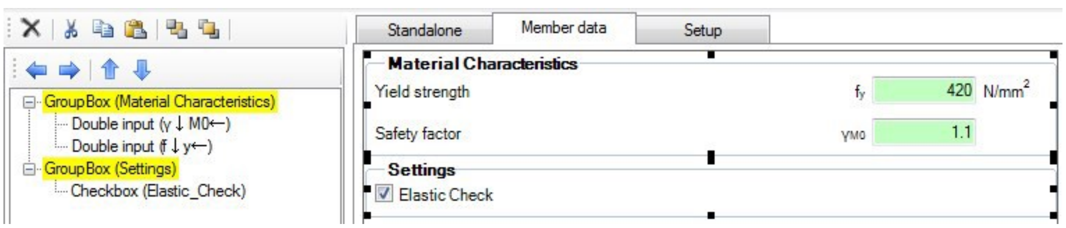
The **Dialog outline** setting can be found at the bottom left side of the screen.

Select the **GroupBox (Material Characteristics)** and **GroupBox (Settings)**. Both can be selected simultaneously by keeping the [CTRL] key pressed.



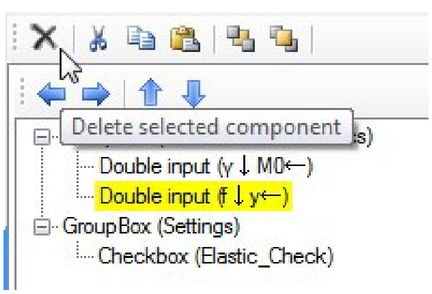
Use the Copy button  to copy these **GroupBoxes** to the clipboard.

Now switch to the '**Member data**' tab, click anywhere in the empty tab to put the focus on that tab and use the Paste button  to paste the components into this tab.

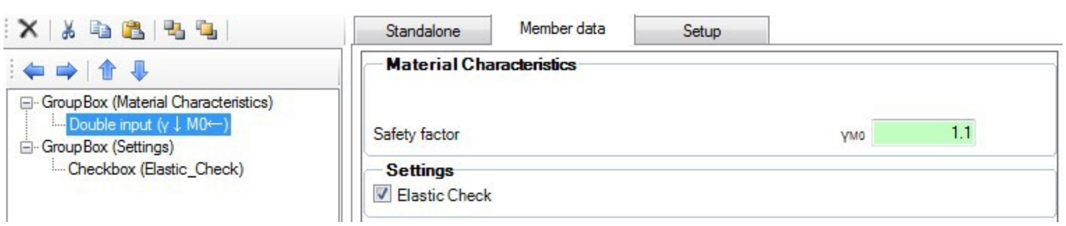


Press [Esc] to end the selection.

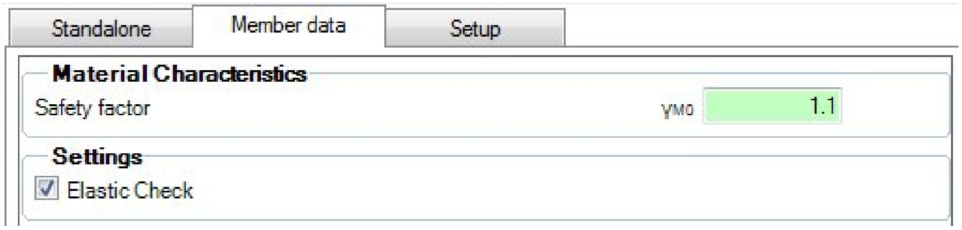
Now the Yield strength can be removed. Select this component either in the dialog or in the Outline and press the Delete button  to remove it.



Press [Esc] to end the selection.



The height of the **Material Characteristics GroupBox** can now be adjusted.



The member data dialog has now been defined. In the next part the properties within this dialog will be given ESA.ID's.

Step 3.2 Set the ID for the Member data Input data

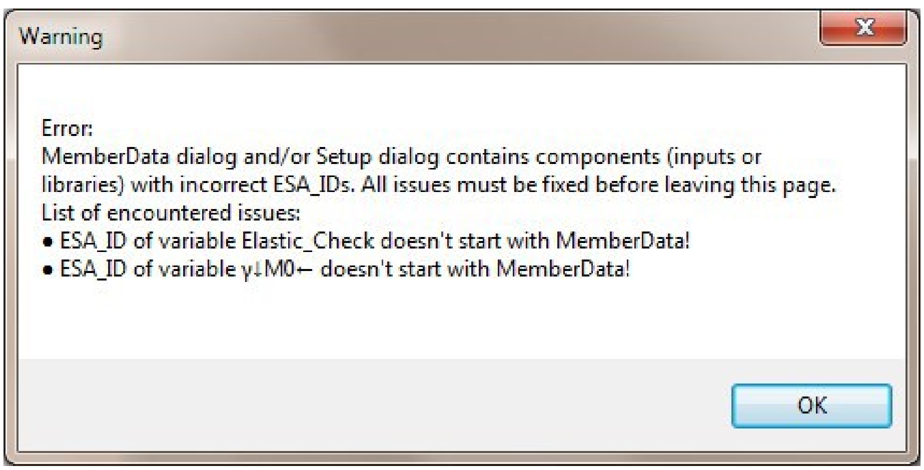
In Step 2 ID's were set for the input data which could be retrieved from the DataCache. Those properties have a fixed ID as specified in Annex B and the SCIA Design Forms WebHelp.

The properties defined as Member data however can be anything, ranging from input fields, checkboxes, combo boxes to pictures, table inputs etc. The author of the Form defines which properties will be set and thus also the author can freely define the ID for these properties.

The Member data dialog will be visible from within SCIA Engineer. Any data inputted in that dialog will automatically be added to the DataCache using the ID set by the Form author after which the Form retrieves it for the check.

The only limitation is that any property defined as Member data has to have an ID starting with the prefix **MemberData**.

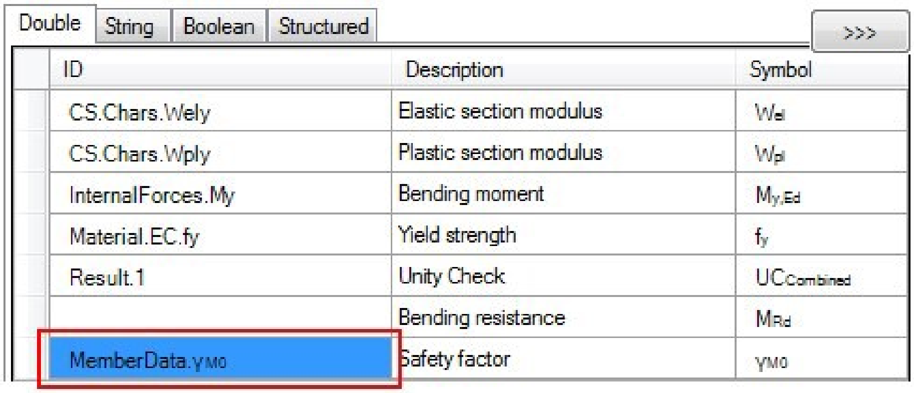
To illustrate this, try changing to the tab '**Calculation**'. SCIA Design Forms will not allow exiting the Dialog tab as long as the Member data properties do not have the proper ID and shows a warning message:



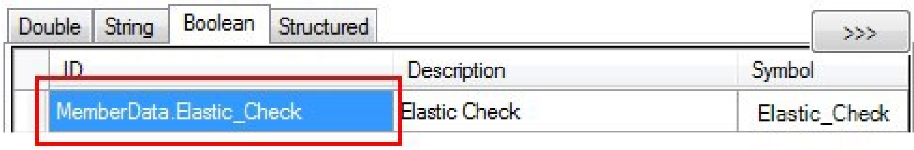
Confirm this message with [OK].

First of all, switch to the  so the grid with input variables becomes visible again.

On the '**Double**' tab, set the ID for the safety factor, in this example **MemberData**.γ_{M0}.



On the '**Boolean**' tab, set the ID for the checkbox, in this example **MemberData.Elastic_Check**.



All properties defined on the Member data dialog now have a properly defined ID. Now it is again possible to switch to the '**Calculation**' tab.

This concludes the definition of the Member data dialog and corresponding ID's.

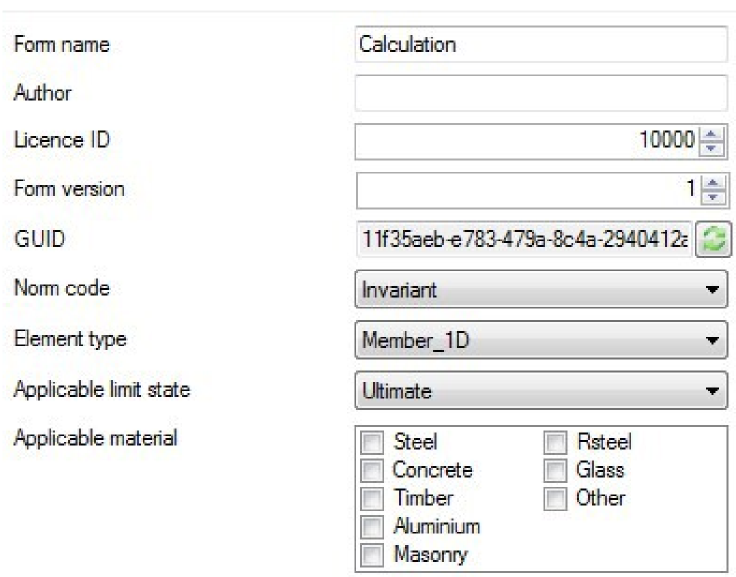
Step 4: Define the Check Header

After setting the ESA.ID's and inputting Member data the 'Check Header' can be defined.

Switch to the tab '**Header**' to define the Check Header.

Step 4.1 Define the Check Header Part 1

When initially opening the 'Header' tab the following data is shown by default:



For this example the **Form name** is set to **Extended Bending Check**.

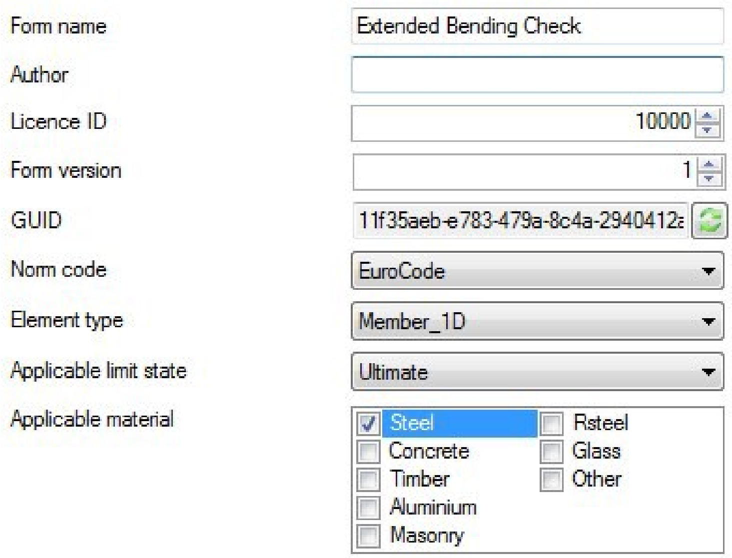
The **Norm code** is set to **EuroCode**.

The **Element type** is set to **Member 1D**.

Applicable limit state is set to **Ultimate**.

For the **Applicable material** the material **Steel** is selected.

After the above modifications the Check Header definition looks as follows:



Step 4.2 Define the Check Header Part 2

The bottom part of the Check Header definition shows a grid which automatically lists all variables that were defined as output results in Step 2.2.

In this example the **UC**$_{Combined}$ was set as **Result.1** so this is the only variable shown here.

The **UC**$_{Combined}$ variable concerns a unity check. The unit is therefore set to **Others - Unity Check [-]**.

Result settings

ESA_ID	Variable name	ESA unit
Result.1	$UC_{Combined}$	Others - Unity Check [-]

All other settings are left on their defaults since they match the typical settings for a unity check.

This concludes the definition of the Check Header; all required inputs have now been made.

Step 5: Export the Form to a .CLC

The final step within the Design Form concerns saving the changes made to the source file (.CLS) and exporting this into a .CLC file.

To this end, first save the Form using the Save button.

Next, export the file to a .CLC using the Export CLC button.

This .CLC file has the same name as the .CLS file and will be generated in the same folder as the .CLS file.

As specified in the chapter on Default folders, this file must be in the 'User folder' in order for it to be visible from within SCIA Engineer.

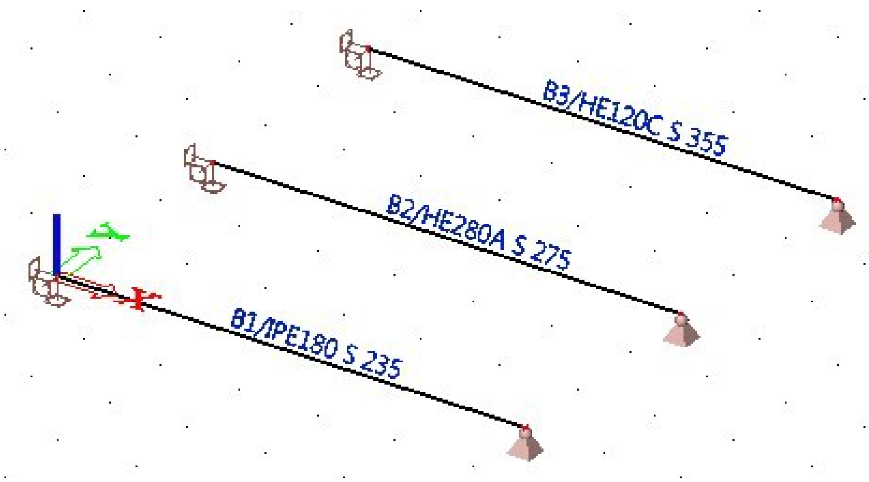
This concludes all steps needed within SCIA Design Forms. The SDF application can now be closed.

Step 6: Import the .CLC into the Check Manager

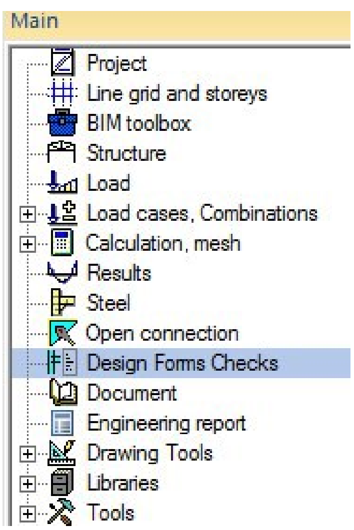
In this specific step, the .CLC file created in the previous Step will be imported within SCIA Engineer.

Launch SCIA Engineer and open the file **Manual_Example_2.esa**.

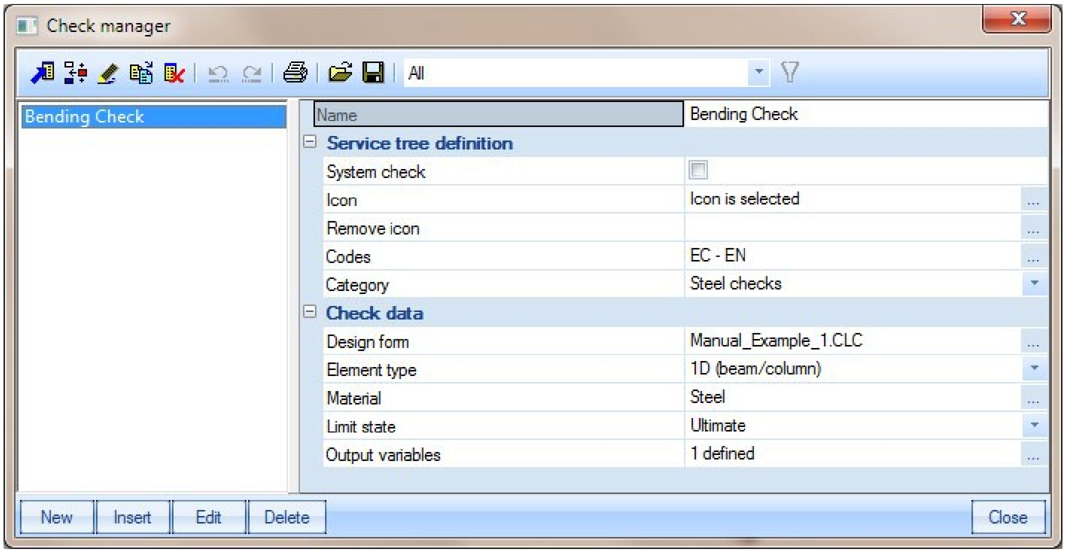
Within this example project, three steel members with different cross-sections and materials are given:



Go to the Design Forms Checks service.

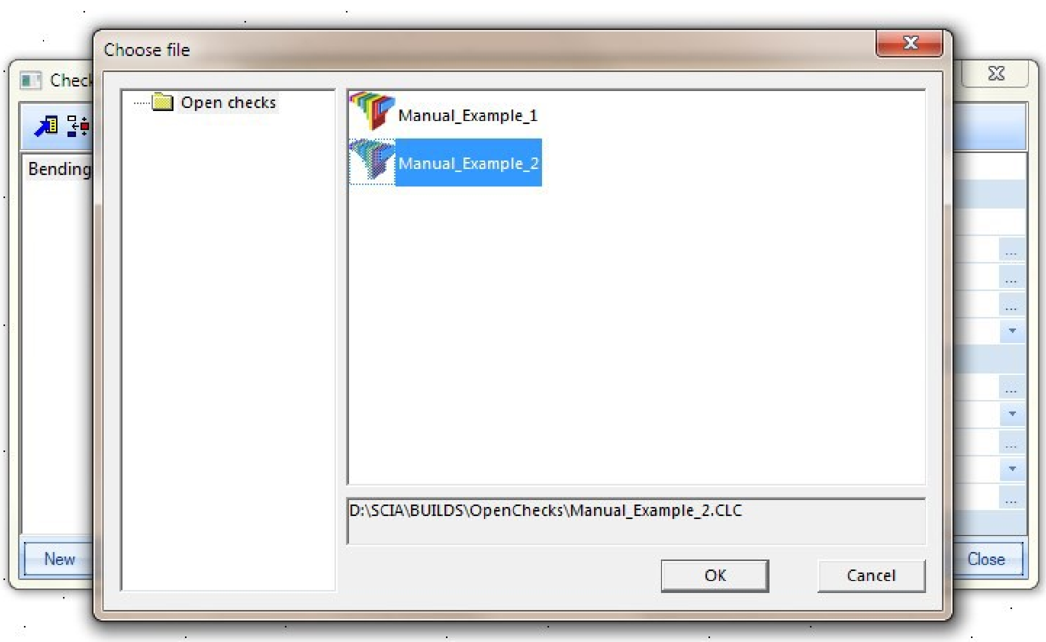


In the Design Forms Checks service open the Check manager.



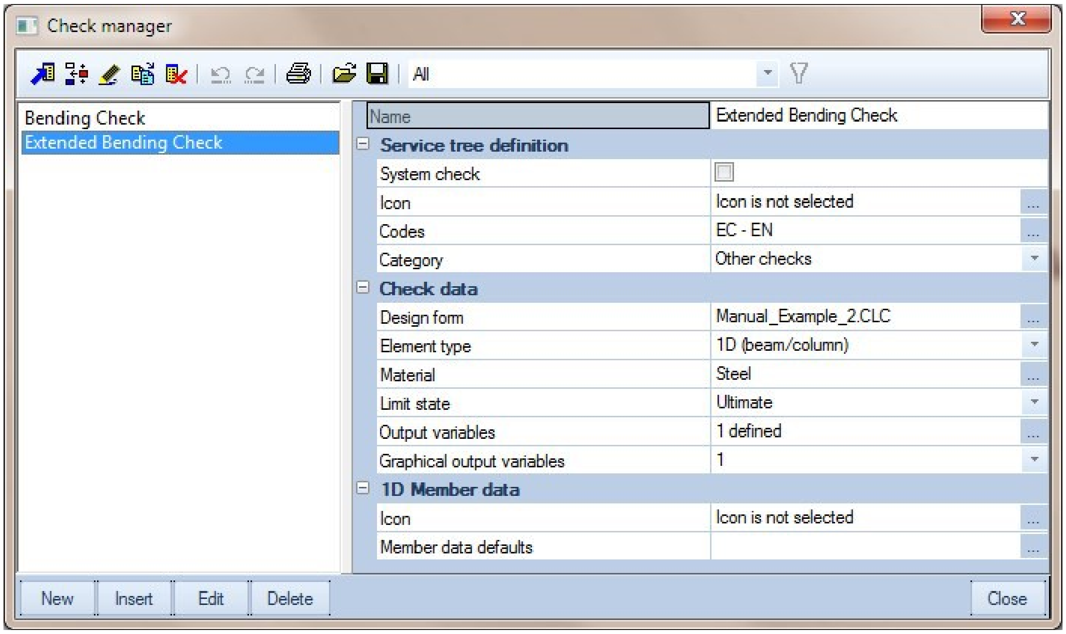
The check from the previous example is still listed here. As specified the **Check Manager** concerns a global repository which is used for the entire SCIA Engineer installation and thus not for a specific project.

Use the New button  to define a new Check. The Choose file dialog appears showing all .CLC files in the OpenChecks folder:



Select the **Manual_Example_2.CLC** file and press [OK].

The **Check Manager** automatically imports the Check header information from the .CLC:



The **Name**, **Codes**, **Element type**, **Material**, **Limit state** and **Output variables** are shown as defined in the .CLC in step 4.

In addition, since Member data was defined in this example an additional subgroup **1D Member data** is shown which contains the Member data dialog.

Step 7: Optionally edit the Check Header and provide icons

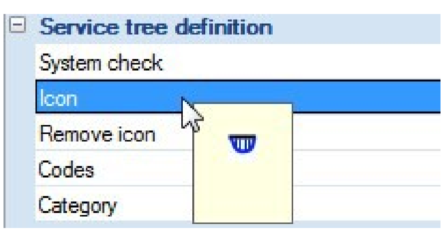
In essence, merely importing the CLC is already sufficient since all required data has been defined.

It is however possible to add additional info to the definition of the check or edit the data which was defined in the Form.

Step 7.1: Service tree definition

The **Service tree** definition shows the info which will be used to display the check in the tree of SCIA Engineer.

For this example a custom icon will be set for the check. Therefore press the ... button and browse for the file **Manual_Example_2_Icon.bmp**



In addition the Category will be set to **Steel checks**.

Step 7.2: Check data

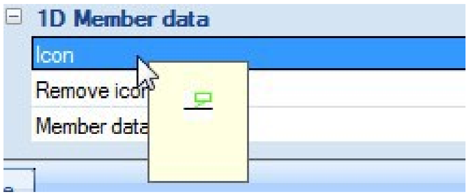
The **Check data** subgroup shows all data relevant to the check.

In this example no changes are made to the imported data.

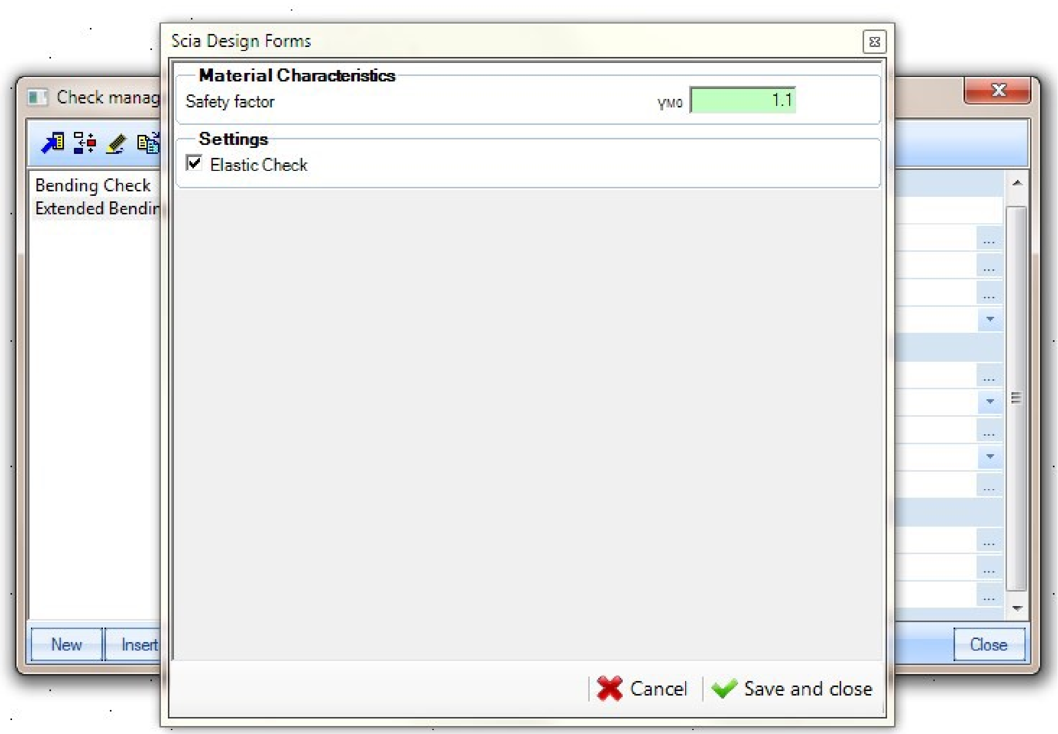
Step 7.3 Member data

Since Member data was defined in the .CLC file a new subgroup **1D Member data** is shown.

In the same way as for the Check itself the Icon setting provides the possibility to set a customized icon for the Member data. Press the ... button and browse for the file **Manual_Example_2_MemberData_Icon.bmp**



The **Member data defaults** setting is used to display the dialog defined in the.CLC file in Step 3 directly within SCIA Engineer. Press the ... button to review the Member data dialog:



This dialog shows the defaults which are used by the Form. Any member which does not have Member data defined will get these values as a default.

Changing the defaults themselves is something which needs to be done within the .CLC file (see Step 3). The dialog shown here is merely a 'Viewer'.

Close the defaults dialog by either Cancel or Save and close since for this dialog both will just close the dialog. In Step 9 the actual input of Member data is discussed which allows editing this data for specific members.

The **Check Manager** can now be closed by pressing [Close].

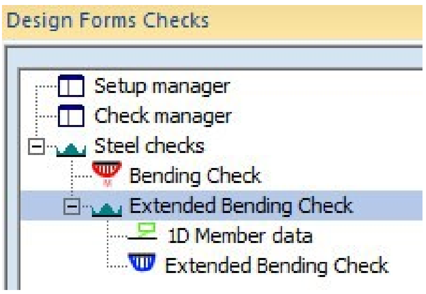
The **Check Manager** contains an additional subgroup labeled SEN Functionality. This can be used to indicate that for example the AutoDesign action buttons (AutoDesign, Split, Unify) should also be visible for this specific check.

Step 8: Execute the Check

All preparations have been made; the Check is linked to SCIA Engineer and can now be executed.

Step 8.1: Display the Brief and Graphical output

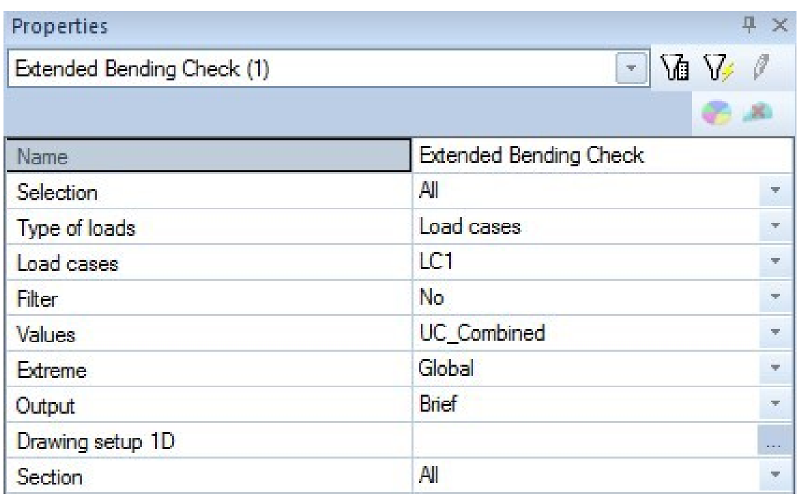
After closing the **Check Manager** the new check will be shown in the **Design Forms Checks** service:



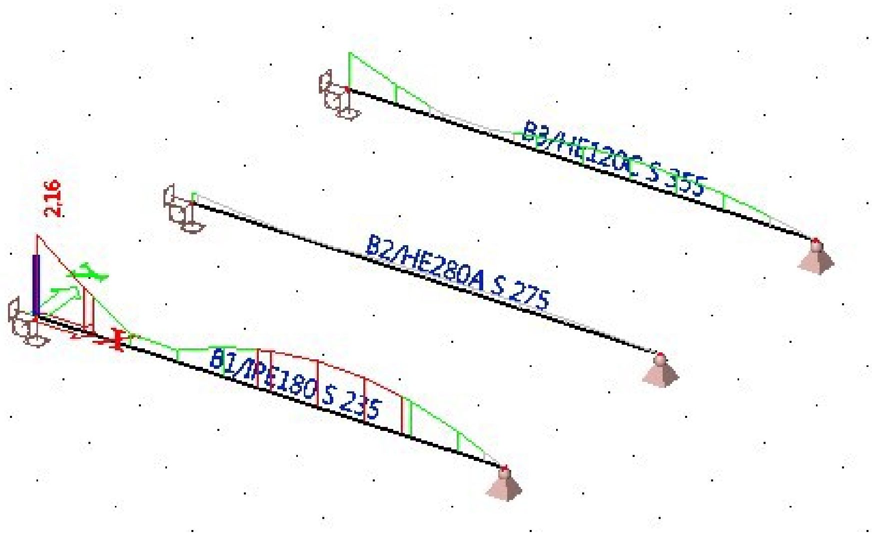
As with all check services in SCIA Engineer, checks are only visible in case the project is calculated. So if the new check is not shown, perform a linear calculation.

The tree shows the subgroup **Steel checks** in which both the **Bending Check** of the previous example and the new **Extended Bending Check** are listed. Since for the **Extended Bending Check** also Member data was defined also a **1D Member data** item is listed in the tree using its appropriate icon.

Select the Extended Bending Check in the service. The Property window will show the properties of the check:



Press the **Refresh** action button to run the check.



Press the **Preview** action button to display the preview of the Brief check:

Extended Bending Check

Type Name	Name	Header	Extended Bending Check
Extended Bending Check	Extended Bending Check	Linear calculation, Extreme : Global Selection : All Load cases : LC1	Extended Bending Check

Type Name	Member	Position [m]	Case	$UC_{Combined}$ [-]
Extended Bending Check	B1	0,000	LC1	**2,16**

Use the **Table composer** as explained in Example 1 to change the layout of the output:

Extended Bending Check

Linear calculation, Extreme : Global
Selection : All
Load cases : LC1
Extended Bending Check

Member	Position [m]	Case	$UC_{Combined}$ [-]
B1	0,000	LC1	**2,16**

Step 8.2: Display the Detailed output

In the Property window change the **Output** to **Detailed**.

Press the Refresh action button to run the check.

Extended Bending Check

Linear calculation, Extreme : Global
Selection : All
Load cases : LC1

Type Name	Detailed output. Page: 000
Extended Bending Check	**Example 2: Extended Steel Bending Check** **Section Properties** W_{el} = 146000 mm^3 W_{pl} = 166000 mm^3 **Internal Forces** Bending moment $M_{y,Ed}$ = -67,3 kNm **Material Characteristics** f_y = 235 N/mm^2 γ_{M0} = 1,1 **Settings** Elastic check only? True **Verification** $M_{Rd} = \frac{f_y \cdot W_{el}}{\gamma_{M0}} = \frac{235*10^6 \cdot 146*10^{-6}}{1,1} = 31,2$ kNm $UC_{Combined} = \frac{abs(M_{y,Ed})}{M_{Rd}} = \frac{abs(-67301)}{31191} = 2,16$

Use the **Table composer** as explained in Example 1 to change the layout of the output:

Extended Bending Check

Linear calculation, Extreme : Global
Selection : All
Load cases : LC1

Example 2: Extended Steel Bending Check

Section Properties

W_{el} = 146000 mm^3
W_{pl} = 166000 mm^3

Internal Forces

Bending moment $M_{y,Ed}$ = -67,3 kNm

Material Characteristics

f_y = 235 N/mm^2
γ_{M0} = 1,1

Settings

Elastic check only? True

Verification

$$M_{Rd} = \frac{f_y \cdot W_{el}}{\gamma_{M0}} = \frac{235*10^6 \cdot 146*10^{-6}}{1,1} = 31,2 \text{ kNm}$$

$$UC_{Combined} = \frac{abs(M_{y,Ed})}{M_{Rd}} = \frac{abs(-67301)}{31191} = 2,16$$

Step 9: Evaluate the Results

With the link and output completed, the results can now be evaluated for different inputs as well as the Member data.

Set the **Selection** to **Current**, select member **B1** and press the **Refresh** action button:

Extended Bending Check

Linear calculation, Extreme : Global
Selection : B1
Load cases : LC1

Example 2: Extended Steel Bending Check

Section Properties

$W_{el} = 146000\ mm^3$

$W_{pl} = 166000\ mm^3$

Internal Forces

Bending moment $M_{y,Ed} = -67,3$ kNm

Material Characteristics

$f_y = 235\ N/mm^2$

$\gamma_{M0} = 1,1$

Settings

Elastic check only? True

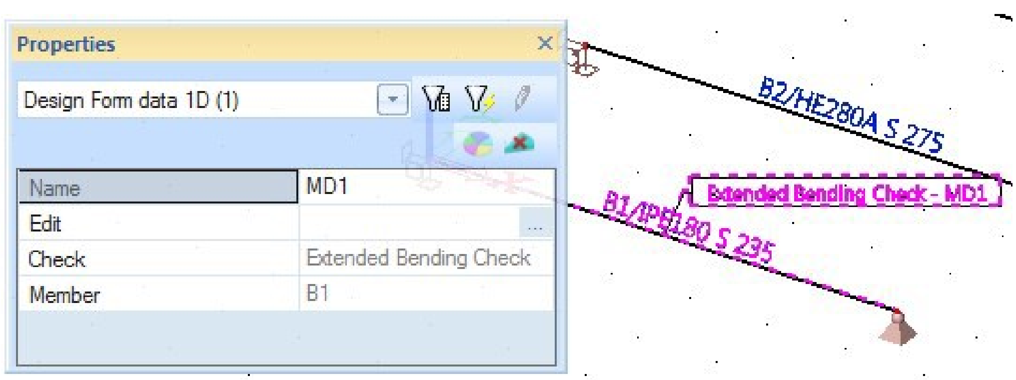
Verification

$$M_{Rd} = \frac{f_y \cdot W_{el}}{\gamma_{M0}} = \frac{235*10^6 \cdot 146*10^{-6}}{1,1} = 31,2\ kNm$$

$$UC_{Combined} = \frac{abs(M_{y,Ed})}{M_{Rd}} = \frac{abs(-67301)}{31191} = 2,16$$

Since no Member data was inputted this member is using the default settings as defined in the .CLC file and shown in Step 7.3. The γ_{M0} is thus set as **1.1** and the **Elastic check** is executed.

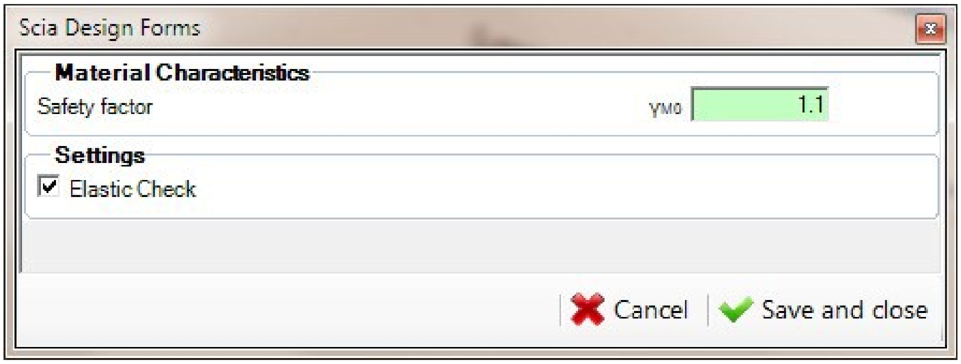
Now double click on the 1D Member data in the tree. Confirm the dialog which appears with [OK] and input this data on member **B1**.



The Property window shows the **Name** of the Member data as well as the **Check** to which it belongs.

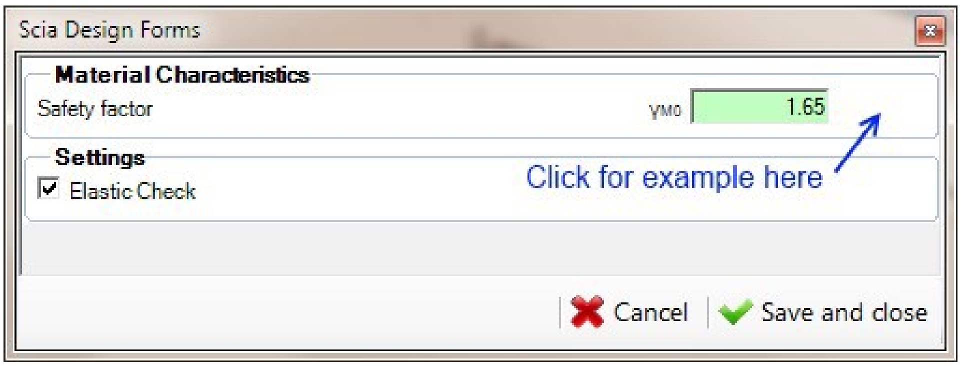
The viewflag shown on member **B1** automatically concatenates this information so it's visually always clear to which Check the data belongs.

Click on the ... button in the Property window to Edit the Member data and display the Member data dialog defined in the .CLC:



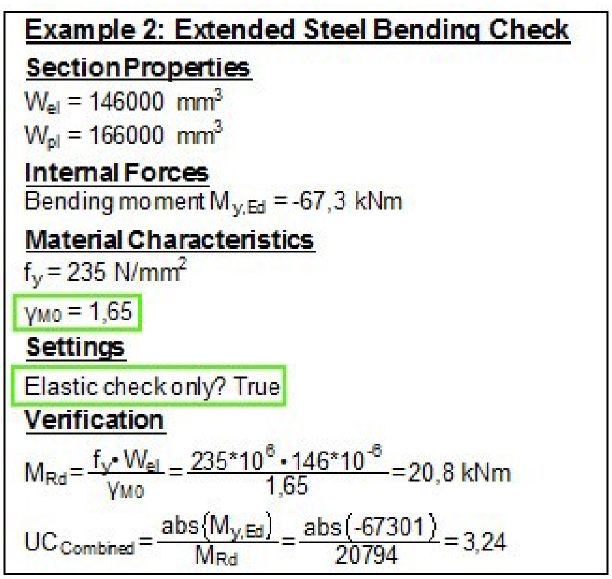
Change the safety factor γ_{M0} for example to **1.65**

After changing the value, click somewhere outside of this input field to make sure the value is confirmed.



Now close the dialog using the "Save and close" button.

Select the "Extended Bending Check" in the service and press the **Refresh** action button to execute the check once more:



Example 2: Extended Steel Bending Check

Section Properties

$W_{el} = 146000\ mm^3$

$W_{pl} = 166000\ mm^3$

Internal Forces

Bending moment $M_{y,Ed} = -67,3$ kNm

Material Characteristics

$f_y = 235\ N/mm^2$

$\gamma_{M0} = 1,65$

Settings

Elastic check only? True

Verification

$$M_{Rd} = \frac{f_y \cdot W_{el}}{\gamma_{M0}} = \frac{235*10^6 \cdot 146*10^{-6}}{1,65} = 20,8\ kNm$$

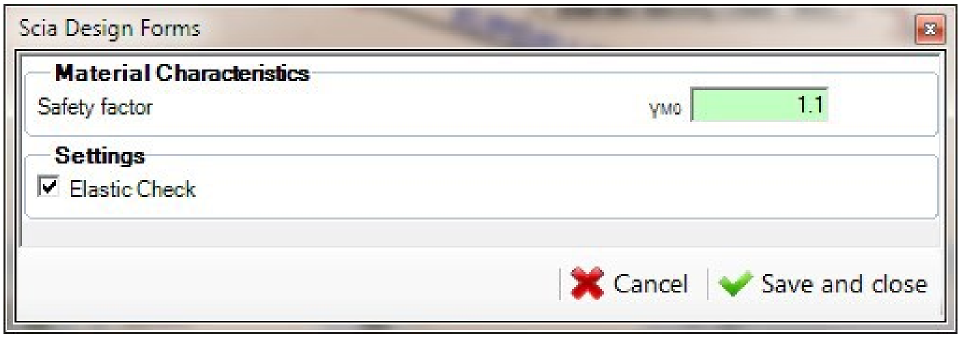
$$UC_{Combined} = \frac{abs(M_{y,Ed})}{M_{Rd}} = \frac{abs(-67301)}{20794} = 3,24$$

The output shows the usage of the Member data, the modified γ_{M0} of **1.65** is now used.

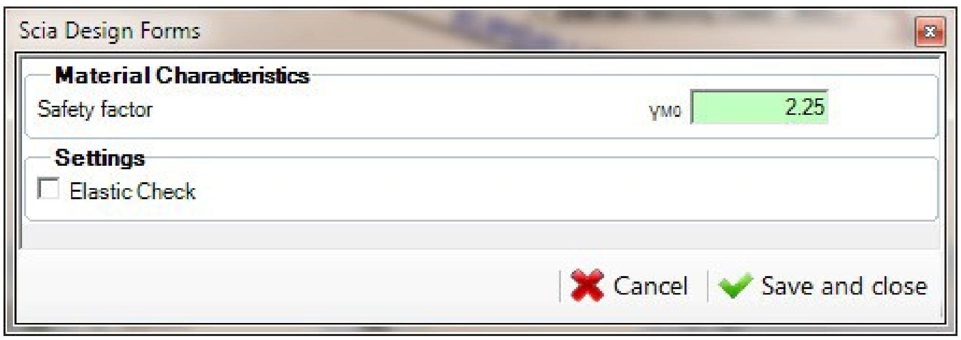
Press [Esc] to end the selection.

Again double click on the 1D Member data in the tree. Confirm the dialog which appears with [OK] and input this data on member **B2**.

Click on the ··· button to **Edit** the Member data on **B2**:



Change the safety factor γ_{M0} for example to **2.25** and de-activate the checkbox for the **Elastic Check**.



Again click somewhere in the dialog outside of the input fields to confirm the data and close the dialog using the Save and close button.

Select only member **B2**, go to the Extended Bending Check in the service and press the **Refresh** action button to execute the check:

Example 2: Extended Steel Bending Check

Section Properties

$W_{el} = 1.01*10^6$ mm^3

$W_{pl} = 1.11*10^6$ mm^3

Internal Forces

Bending moment $M_{y,Ed} = -66,6$ kNm

Material Characteristics

$f_y = 275$ N/mm^2

$\gamma_{M0} = 2,25$

Settings

Elastic check only? False

Verification

$$M_{Rd} = \frac{f_y \cdot W_{pl}}{\gamma_{M0}} = \frac{275*10^6 \cdot 1.11*10^{-3}}{2,25} = 136 \text{ kNm}$$

$$UC_{Combined} = \frac{abs(M_{y,Ed})}{M_{Rd}} = \frac{abs(-66646)}{135911} = 0,49$$

The output shows the usage of the Member data, the modified γ_{M0} of 2.25 is now used and the **Elastic check** is shown as **False**.

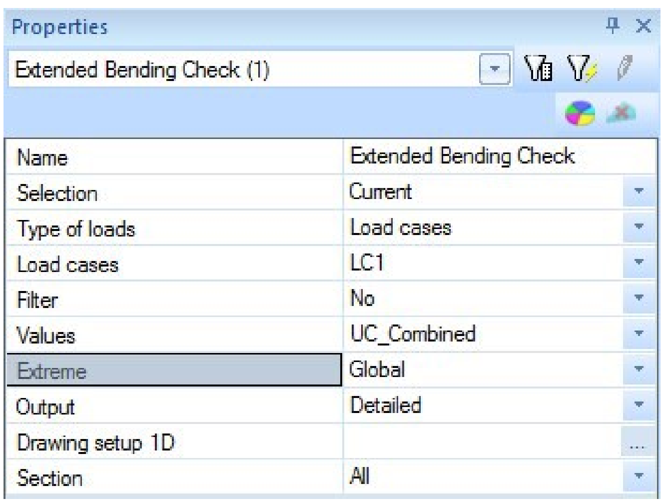
This quick overview of the results illustrates how Member data can be used to modify the input data for each member.

Step 10: Optionally review the DataCache and Trace file

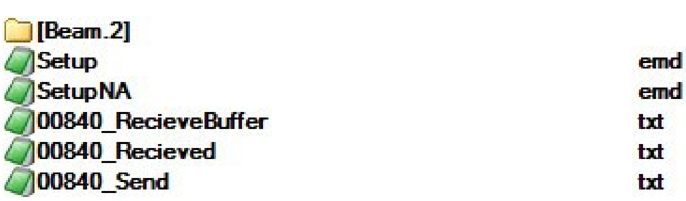
The final step of this example illustrates the internal transfer of data between SCIA Engineer and SCIA Design Forms.

Before going through this step please read the information contained within Annex A which explains the Command line switches and generation of the **.trace** file.

- First of all, close SCIA Engineer. The changes to the project file can be **saved** so the inputted member data is also stored within the project.
- Next, launch SCIA Engineer using the Command line switch **-OCFILES**.
- Open the file **Manual_Example_2.esa**.
- Perform a **Linear** calculation to refresh the results.
- Go to the Design Forms Checks service.
- Select the Extended Bending Check in the service.
- Set the following Properties for the Check:



- - Select member **B2**
- - Press the **Refresh** action button to run the check
- - Using a file explorer like **Windows Explorer**, navigate to the **TEMP** folder of SCIA Engineer and locate the subfolder **ESA_Model_Data**



Now go to the subfolder **Beam.2**. This folder contains all input data (**.emd**), output data (**.results**) and the **.trace** file.

1DMemberInfo	emd
Beam	emd
BucklingData	emd
ConcreteMemberDataNew	emd
CrossSection	emd
Deformations	emd
InternalForces	emd
Load	emd
Materials	emd
MemberData	emd
MemberInfo	emd
Sections	emd
Beam.2	results
Beam.2	trace
Beam.2-000	WMF

When looking inside the **MemberData.emd** file using a standard text editor like **NotePad** the settings for the Member data for this beam can be found:

```
(:MemberData :Elastic_Check False :γ↓M0← 2.25 )
```

This data is thus transferred to the DataCache using the ID's as defined in Step 3.2.

The following picture shows the **.trace** file content:

```
----------------------------------------------------------------------------------------
MEMBER = Beam.2; SECTION = 0; COMBI = 0; LAYOUT = 0; LANGUAGE = 0; PAGING = True
----------------------------------------------------------------------------------------
Reading values from IO: {
  W↓el← = CS.Chars.Wely = 1.01*10↑-3←
  W↓pl← = CS.Chars.Wply = 1.11*10↑-3←
  M↓y,Ed← = InternalForces.My = -66646
  f↓y← = Material.EC.fy = 275*10↑6←
  γ↓M0← = MemberData.γ↓M0← = 2.25
  Elastic_Check = MemberData.Elastic_Check = False
}
TEXT(Example 2: Extended Steel Bending Check)
TEXT(Section Properties)
TEXT(W↓el← = 1013000 mm↑3←)
TEXT(W↓pl← = 1112000 mm↑3←)
TEXT(Internal Forces)
TEXT(Bending moment M↓y,Ed← = -66,6456 kNm)
TEXT(Material Characteristics)
TEXT(f↓y← = 275 N/mm↑2←)
TEXT(γ↓M0← = 2,25)
TEXT(Settings)
TEXT(Elastic check only? False)
TEXT(Verification)
IF (False) {
  M↓Rd← = 135911
}
UC↓Combined← = 0.49
```

The first part shows the data which is read from the DataCache as defined in Step 2 as well as the Member data defined in Step 3.

The rest of the trace shows the different lines of the Design Form with filled in data.

This final Step shows how, using the **-OCFILES** switch, it is possible to debug the SCIA Design Forms Check and perform an in depth evaluation of the Check run including the use of Member data.

Close SCIA Engineer and discard any changes made.

This concludes the second example which illustrated the use of Member data to directly access the input dialog from SCIA Design Forms through SCIA Engineer.

Example 3: Using Standard and Custom Libraries

In this third example a Form using a Standard and a Custom library is used in order to illustrate how such libraries are accessed from within SCIA Engineer.

A simplified Concrete calculation is used which determines the amount of tension reinforcement needed for a rectangular section in bending.

The following pictures show the output of the Design Form and the related script:

Example 3: Concrete Reinforcement

Section Properties

b = 260 mm
h = 740 mm

Internal Forces

Bending moment $M_{y,Ed}$ = 250 kNm

Material Characteristics

$f_{ck} = 30\ N/mm^2$

$\gamma_c = 1,5$

$f_{yk} = 500\ N/mm^2$

$\gamma_s = 1,15$

Verification

$$f_{cd} = \frac{0.85 \cdot f_{ck}}{\gamma_c} = \frac{0.85 \cdot 30*10^6}{1,5} = 17\ N/mm^2$$

$$f_{yd} = \frac{f_{yk}}{\gamma_s} = \frac{500*10^6}{1,15} = 435\ N/mm^2$$

$$d = 0.9 \cdot h = 0.9 \cdot 0,74 = 666\ mm$$

$$\mu_d = \frac{M_{y,Ed}}{b \cdot d^2 \cdot f_{cd}} = \frac{250000}{0,26 \cdot 0,666^2 \cdot 17*10^6} = 0,128$$

$$=> \omega_1 = 0,140$$

$$A_s = \omega_1 \cdot \frac{f_{cd}}{f_{yd}} \cdot b \cdot d = 0,14 \cdot \frac{17*10^6}{435*10^6} \cdot 0,26 \cdot 0,666 = 945\ mm^2$$

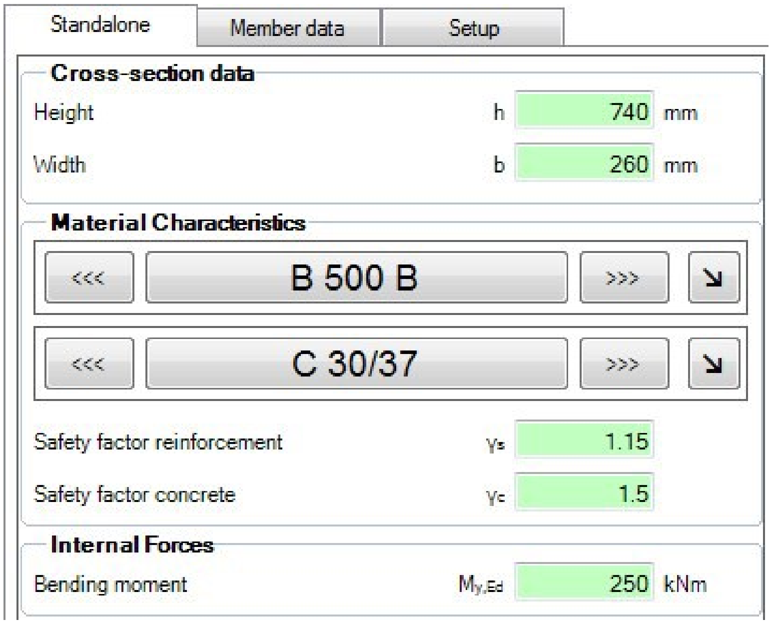
The actual Check starts on Line 417.

```
417 TEXT("Example 3: Concrete Reinforcement");
418
419 TEXT("Section Properties");
420 TEXT("b = " & VAL(1000*b, 2) & " mm");
421 TEXT("h = " & VAL(1000*h, 2) & " mm");
422
423 TEXT("Internal Forces");
424 TEXT("Bending moment M_y,Ed = " & VAL(10^-3*M_y,Ed, 2) & " kNm");
425
426 f_ck = IO.CONCRETE.EC.fck;
427 f_yk = IO.Reinforcement.EC.fyk;
428
429 TEXT("Material Characteristics");
430 TEXT("f_ck = " & VAL(10^-6*f_ck, 2) & " N/mm^2");
431 TEXT("γ_c = " & γ_c);
432 TEXT("f_yk = " & VAL(10^-6*f_yk, 2) & " N/mm^2");
433 TEXT("γ_s = " & γ_s);

434
435 TEXT("Verification");
436 f_cd = 0.85*f_ck/γ_c;
437 f_yd = f_yk/γ_s;
438 d = 0.9*h;
439
440 μ_d = M_y,Ed/(b*d^2*f_cd);
441
442 ω_1 = Get_ω_1(μ_d);
443 TEXT("=> ω_1 = " & ω_1.ToString("0.000"));
444 A_s = ω_1*(f_cd/f_yd)*b*d;
```

Lines 426 & 427 contain IO structure references to the data of the Libraries used for the standalone run of the Form.

Line 442 shows a function **Get_ω1** which determines the reinforcement percentage in function of the bending moment. The function and tabulated data is available on **Lines 1 to 415** but is not detailed further in this example.

The Form uses the following variables as manual input when running standalone:



The **Material Characteristics** are using two libraries:

- A 'Standard' library for the Concrete Material, provided by the Design Form.

- A 'Custom' library for the Reinforcement material

The purpose of this example is to illustrate how Standard and Custom libraries are linked with the DataCache. Also the difference in the IO structure between a standalone run of the Form and a run from within SCIA Engineer is discussed.

> In essence there is no difference between Standard and Custom libraries, both are dialog components linked to a specific .xml file.

As specified in the introduction, the following steps will be followed:

- Step 1: Put the .CLS file in the 'User' folder

Within SCIA Design Forms:

- Step 2: Set the ESA ID for each variable which needs to be linked
- Step 3: Optionally define Member data
- Step 4: Define the Check Header
- Step 5: Export the Form to a .CLC

Within SCIA Engineer:

- Step 6: Import the .CLC into the Check Manager
- Step 7: Optionally edit the Check Header and provide icons
- Step 8: Execute the Check
- Step 9: Evaluate the Results
- Step 10: Optionally review the DataCache and Trace file

Step 1: Put the .CLS file in the 'User Folder'

Before starting, make sure the Form **Manual_Example_3.CLS** is located in the 'User folder' as outlined in the Chapter on Default folders.

Step 2: Set the ESA ID for each variable which needs to be linked

As demonstrated in the previous examples, upon executing the check from within SCIA Engineer, an export will be made of all relevant model data (cross-sections, materials, internal forces ...) to the 'DataCache'.

In this first step, the ESA ID's for the relevant variables which will be linked are defined in the Form.

Launch the SCIA Design Forms Builder and open the Form **Manual_Example_3.CLS**.

Step 2.1: Set the ID for the Input data

Within this Form, the grid on the 'Calculation' tab shows the double values to be manually inputted in a green color:

Double | String | Boolean | Structured

ID	Description	Symbol	Value	Unit
	Iteration step	i	0	
	Bending moment	$M_{y,Ed}$	250	kNm
	Characteristic concrete compressive strength	f_{ck}	30	N/mm²
	Characteristic reinforcement yield strength	f_{yk}	500	N/mm²
	Safety factor concrete	γ_c	1.5	
	Safety factor reinforcement	γ_s	1.15	
	Design concrete compressive strength	f_{cd}	17	N/mm²
	Design reinforcement yield strength	f_{yd}	435	N/mm²
	Leverage arm	d	666	mm
	Width	b	260	mm
	Reduced value of the bending moment	μ_d	0.128	
	Height	h	740	mm
	Reinforcement percentage	ω_1	0.14	
	Reinforcement area	A_s	945	mm²

The iteration step i concerns a variable used internally by the **Get_ω1** function and is thus not a value to be inputted.

The safety factors γ_c and γ_s concern properties which do not exist in the DataCache, so these will be set later on as Member data.

Using the basic information from Annex B the following ID's are found to represent the data needed by the form:

ESA ID	Variable
InternalForces.My	Bending moment about the principal y-axis
CS.Geometry.B	Width of rectangular cross-section
CS.Geometry.H	Height of rectangular cross-section

These ID's are inputted within the grid of the Form:

Double | String | Boolean | Structured

ID	Description	Symbol	Value	Unit
	Iteration step	i	0	
InternalForces.My	Bending moment	$M_{y,Ed}$	250	kNm
	Characteristic concrete compressive strength	f_{ck}	30	N/mm^2
	Characteristic reinforcement yield strength	f_{yk}	500	N/mm^2
	Safety factor concrete	γ_c	1.5	
	Safety factor reinforcement	γ_s	1.15	
	Design concrete compressive strength	f_{cd}	17	N/mm^2
	Design reinforcement yield strength	f_{yd}	435	N/mm^2
	Leverage arm	d	666	mm
CS.Geometry.B	Width	b	260	mm
	Reduced value of the bending moment	μ_d	0.128	
CS.Geometry.H	Height	h	740	mm
	Reinforcement percentage	ω_1	0.14	
	Reinforcement area	A_s	945	mm^2

This covers the basic inputs however the libraries are referenced directly within the script.

Within this example, two libraries are used. The standard Concrete material library is pointing to the following **Target IO node** ('**Dialog**' tab):

Target IO node	**CONCRETE.EC**

The custom library for the Reinforcement material is pointing to the following **Target IO node**:

Target IO node	**Reinforcement.EC**

These **Target IO nodes** are available purely during the standalone run of the Form and are being used directly on **lines 426 & 427**:

```
426  fck = IO.CONCRETE.EC.fck;
427  fyk = IO.Reinforcement.EC.fyk;
```

Both **Target IO node** strings can in fact be freely defined by the author of the Form.

However, the DataCache does not contain this structure, so the above two lines are not valid in case the Form is run from within SCIA Engineer. When running the Form from within SCIA Engineer it is thus required to set the correct IO structure for these variables so they are read from the DataCache.

Instead of just providing the solution, let us walk through the DataCache to see what is available and how the $\mathbf{f}_{ck}$ and $\mathbf{f}_{yk}$ variables can be correctly retrieved both in the standalone run and in the linked run.

Intermezzo: Using the EMD Loader to examine what is in the DataCache

In this subpart of the example the **EMD Loader** is used to examine what is available in the DataCache. Annex A also contains information on the **EMD Loader.**

Using SCIA Engineer the DataCache files will be generated after which their content will be examined from within SCIA Design Forms.

- First of all, close SCIA Engineer in case it was still open.

- Next, launch SCIA Engineer using the Command line switch **-OCEMD**. This switch makes sure the required .emd files will be generated when executing the Check. Refer to Annex A for more information.

- Open the file **Manual_Example_3.esa**.

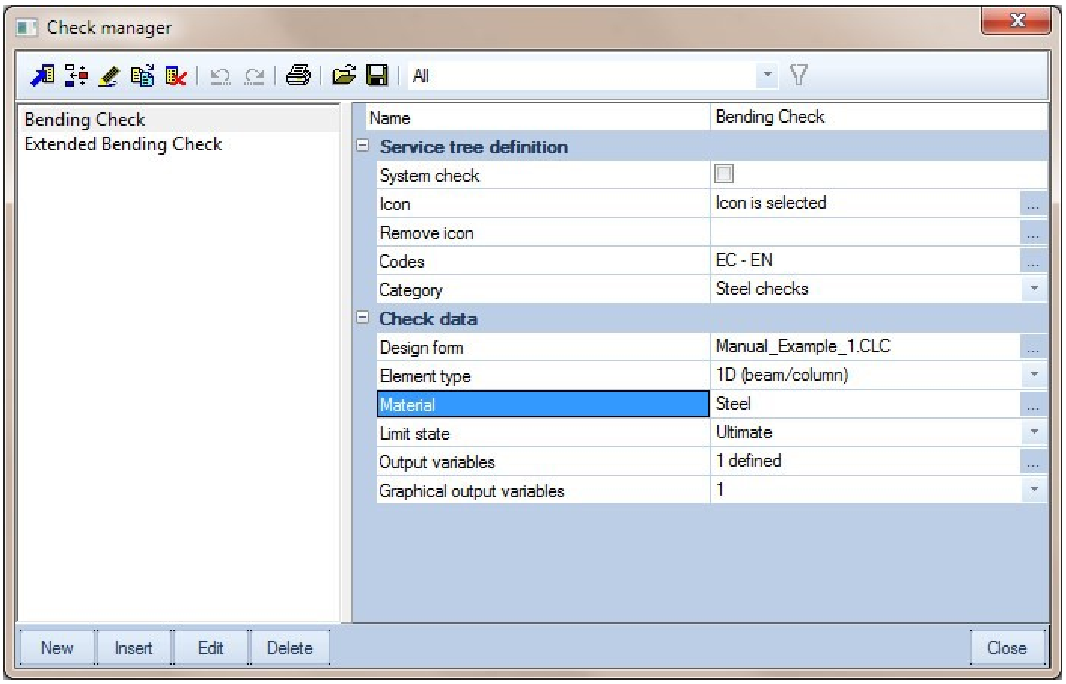
- Perform a Linear calculation to refresh the results.

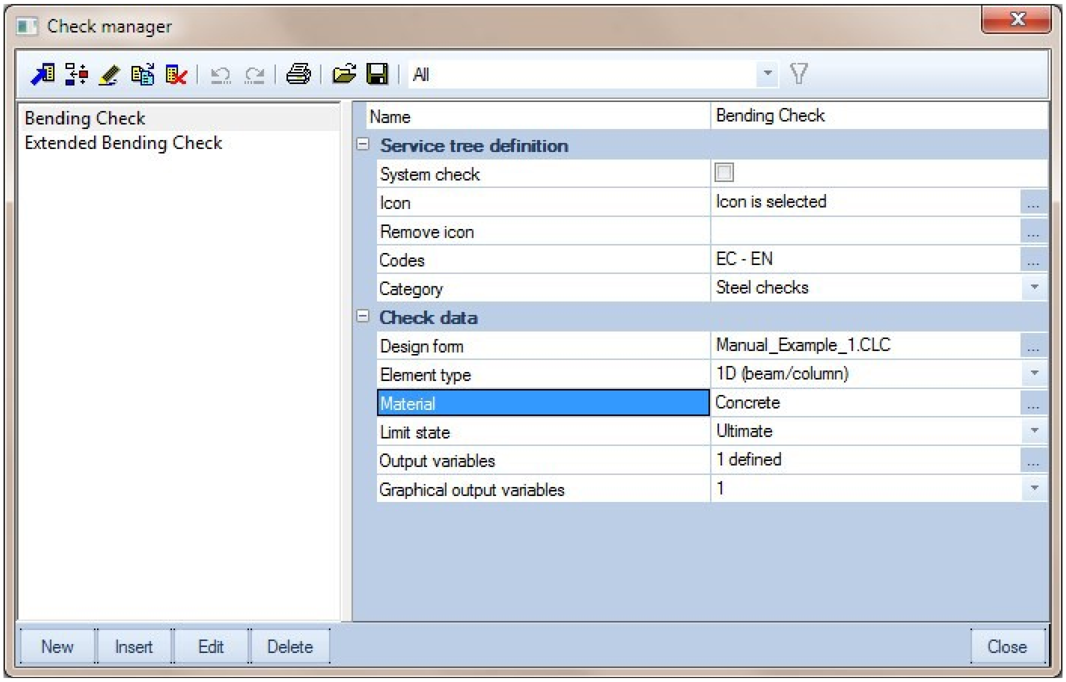
- Go to the Design Forms Checks service.

The project file in this case contains two Concrete members. Since the Checks defined in the previous examples require members made out of Steel these Checks will not work with these members.

- Therefore, open the **Check Manager**.

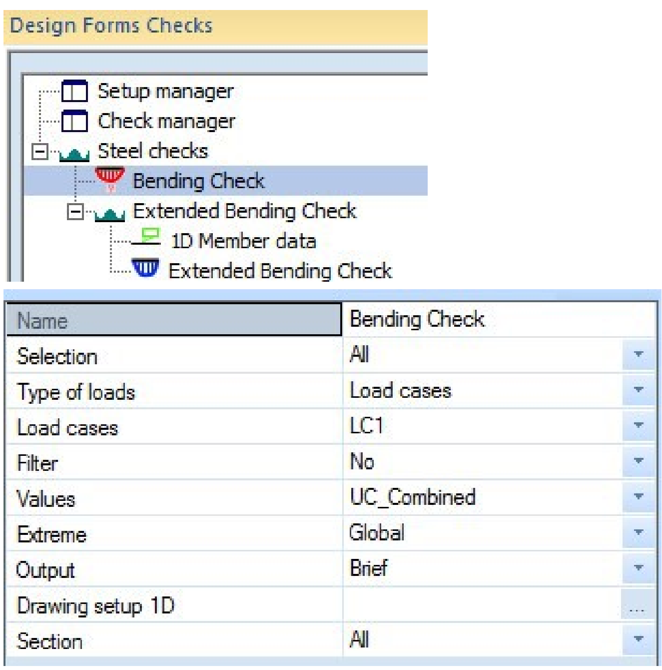
- Select the **Bending Check** and change its material from **Steel** to **Concrete**:





- Close the **Check Manager** with [Close]

- In the Service, select the Bending Check and set the following properties:



- Press the **Refresh** action button.

No results are shown, it seems as if nothing happened, but this is correct and caused by the usage of the **-OCEMD** switch. This switch generated the **.emd** files but does not run any check and thus results are neither generated nor shown.

- Using a file explorer like **Windows Explorer**, navigate to the **TEMP** folder of SCIA Engineer and locate the subfolder **ESA_Model_Data**

[Beam.1]	
[Beam.2]	
Setup	emd
SetupNA	emd
06404_Commands	txt
09180_Commands	txt

Now go to the subfolder Beam.1. This folder contains all input data (.emd) for this member.

1DMemberInfo	emd
Beam	emd
BucklingData	emd
ConcreteMemberDataNew	emd
CrossSection	emd
Deformations	emd
InternalForces	emd
Load	emd
Materials	emd
MemberInfo	emd
Sections	emd

When looking inside the **Materials.emd** file using a standard text editor like **NotePad** the data for the Material for this beam can be found:

```
(:Material :Type eConcrete :ID 318 :Name "C25/30" )
  (:EC :Ecm 3.15e+010 :G 1.3125e+010 :fck 2.5e+007 :fctm 2.6e+006
    (:Diagram :Type "ULS" )
      (:Point :sig 0 :eps 0 )
      (:Point :sig -1.66667e+007 :eps -0.00175 )
      (:Point :sig -1.66667e+007 :eps -0.0035 )
(:Material :Type eRsteel :ID 347 :Name "B 500B" )
  (:EC :Es 2e+011 :G 8.33333e+010 :fyk 5e+008 :eps_uk 0.05 :k 1.08
    (:Diagram :Type "ULS" )
      (:Point :sig -4.69565e+008 :eps -0.05 )
      (:Point :sig -4.6751e+008 :eps -0.045 )
      (:Point :sig -4.34783e+008 :eps -0.00217391 )
      (:Point :sig 0 :eps 0 )
      (:Point :sig 4.34783e+008 :eps 0.00217391 )
      (:Point :sig 4.6751e+008 :eps 0.045 )
      (:Point :sig 4.69565e+008 :eps 0.05 )
```

The file shows that there are two materials, **Concrete** and **Reinforcement Steel**. Each of these materials has its own **ID** by which it is referenced to in other parts of the DataCache.

Looking for example in the **CrossSection.emd** file it can be seen that the cross-section refers to the **Material ID** of the Concrete material.

```
(:CrossSection :ID 1 :MaterialID 318 :Name "CS1" :Description "Rectangle" )
  (:IsPhased 0 )
  (:Geometry :FormCode 2003 :B 0.3 :H 0.5 )
  (:Chars :A 0.15 :Ay 0.125 :Az 0.125 :AL 1.6 :AD 1.6 :cYUCS 0.15 :cZUCS 0.25
  (:Component :ID 0 :MaterialID 318 )
    (:Chars :A 0.15 :Ay 0 :Az 0 :AL 0 :AD 0 :cYUCS 0 :cZUCS 0 :IYLCS 0.003125
    (:Shape )
      (:Point :X 0.15 :Y -0.25 )
      (:Point :X 0.15 :Y 0.25 )
      (:Point :X -0.15 :Y 0.25 )
      (:Point :X -0.15 :Y -0.25 )
      (:Point :X 0.15 :Y -0.25 )
  (:Fibres )
    (:Fibre :X 0.15 :Y -0.25 :Flag 0 )
    (:Fibre :X 0.15 :Y 0 :Flag 0 )
    (:Fibre :X 0.15 :Y 0.25 :Flag 0 )
    (:Fibre :X 0 :Y 0.25 :Flag 0 )
    (:Fibre :X -0.15 :Y 0.25 :Flag 0 )
    (:Fibre :X -0.15 :Y 0 :Flag 0 )
    (:Fibre :X -0.15 :Y -0.25 :Flag 0 )
    (:Fibre :X 0 :Y -0.25 :Flag 0 )
```

In the same way the **Beam.emd** file shows the reinforcement layers which refer to the **Material ID** of the **Reinforcement Steel**.

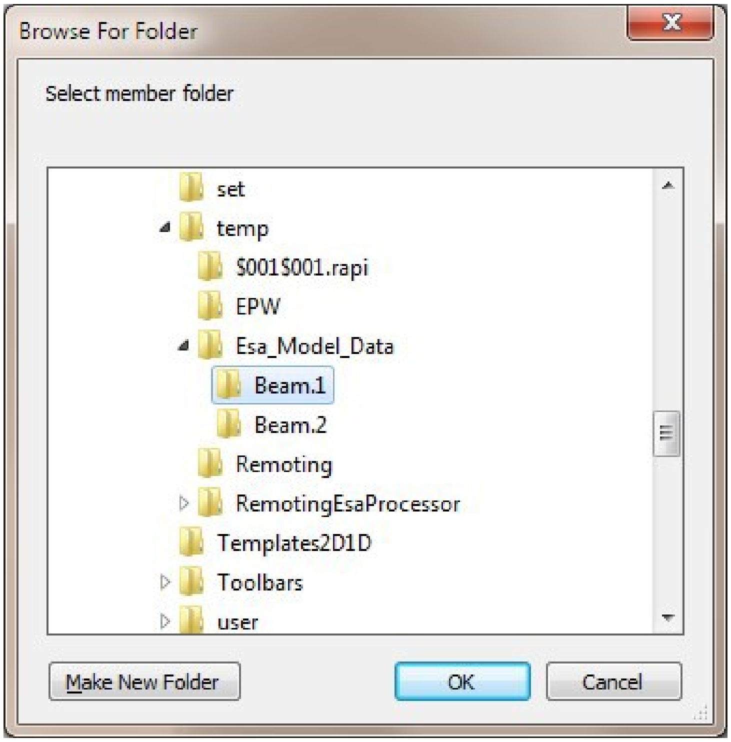
```
(:ReinfLayers )
  (:Lower :Cover 0.035 :Diameter 0.02 :MaterialID 347 )
  (:Upper :Cover 0.035 :Diameter 0.02 :MaterialID 347 )
```

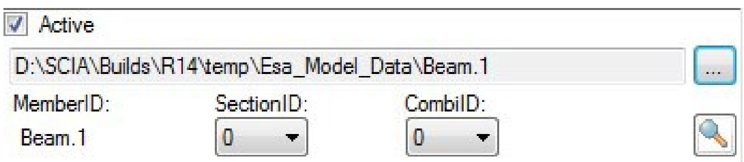
To illustrate exactly how the DataCache is seen by SCIA Design Forms the **EMD Loader** will be used.

- Open the SCIA Design Forms Builder and create a new empty Form.

- Go to the '**Dialog**' tab and add the EMD loader component to the Standalone dialog.

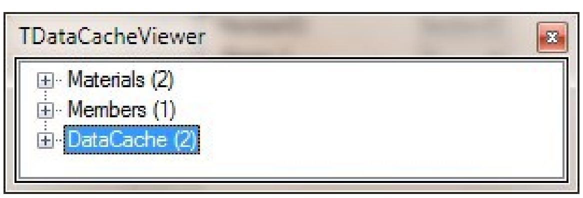
- In this component, use the ... button and navigate to the **TEMP** folder of SCIA Engineer and further down to the **ESA_ Model_Data** and **Beam.1** folder:



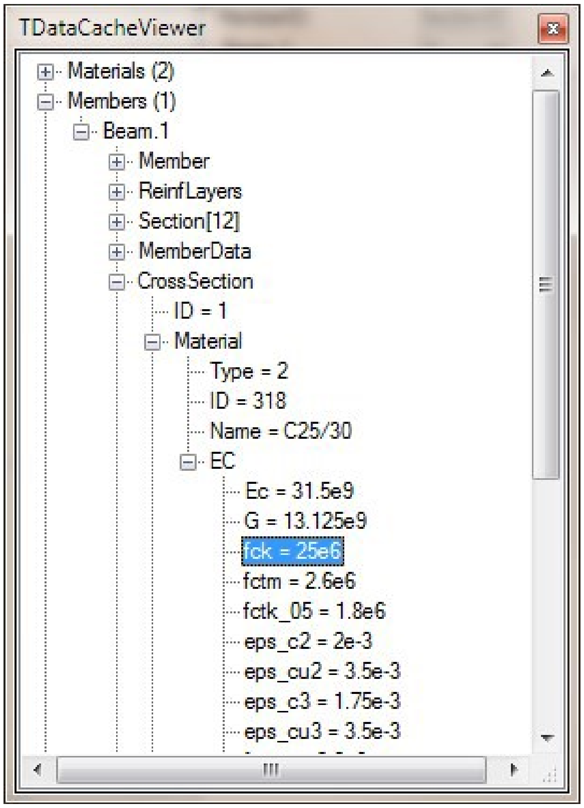


The folder on the screenshot will differ from your own folder and depends on the installation and version of SCIA Engineer.

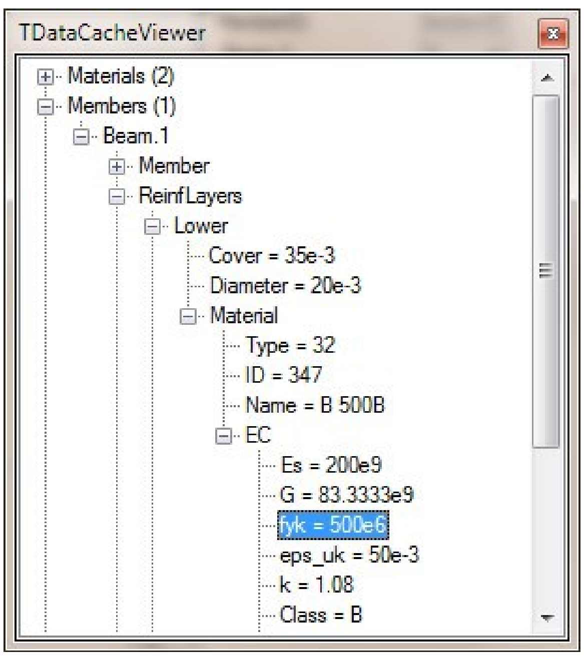
- Use the Spyglass button to see the contents of the DataCache



- The tree structure can now be used to navigate through the DataCache. For example the $\mathbf{f}_{ck}$ value for the (Concrete) **Material** of the **Cross-section** of the **Beam** can be found here:



- In the same way the $\mathbf{f}_{yk}$ value of the **Lower Reinforcement Layer** of the **Beam** can be found here:

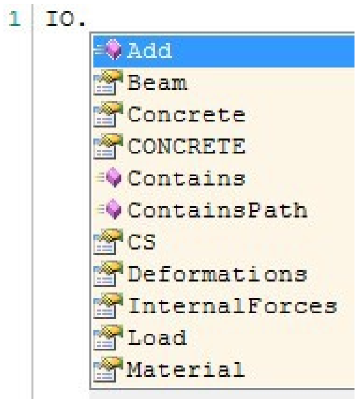


- Close the DataCacheViewer using the close  button.

- Now go to the '**Calculation**' tab.

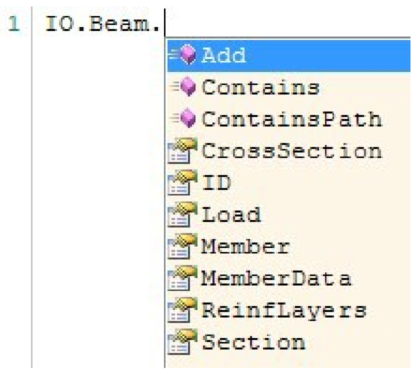
- In the editor, type **IO.** (Including the dot)

Immediately an Intellisense window will be shown which shows the content of the DataCache:

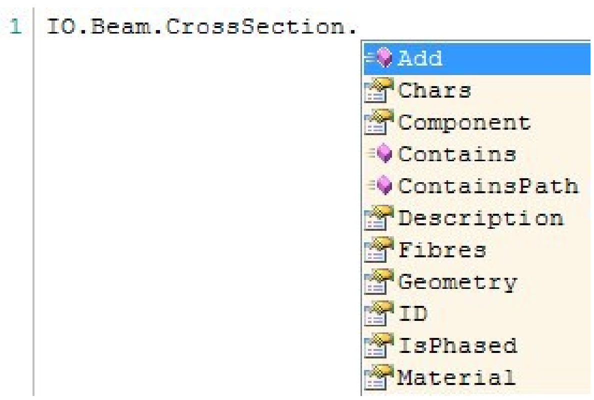


The **EMD Loader** has loaded the content of the **.emd** files into the IO Structure so it can be directly accessed here.

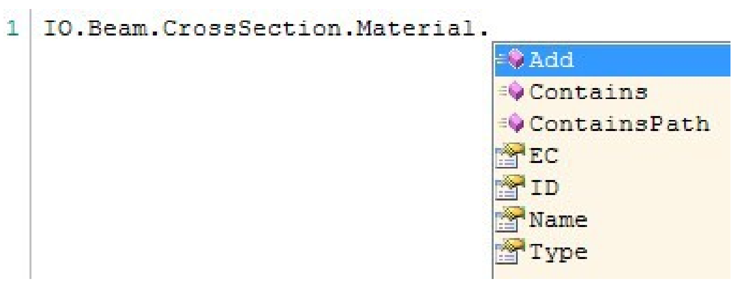
- Select **Beam** from the list and again type a dot. A new Intellisense will be shown:



We are now accessing the DataCache node of the Beam. As shown above we need to access the material of the cross-section thus select **CrossSection** from the list again followed by a dot.



The Intellisense shows the data within the **CrossSection** node. Now select **Material** again followed by a dot.



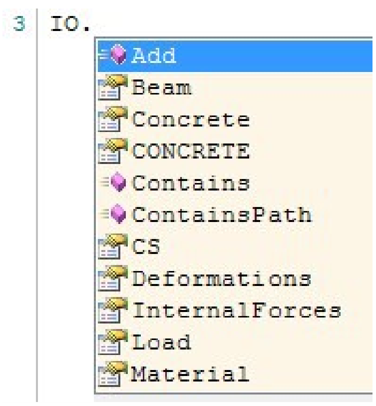
Now we are really accessing the material of the Cross-section i.e. the Concrete.

- Select **EC** followed by a dot.

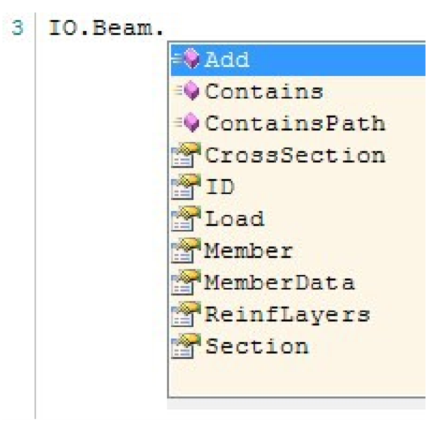
Here we now see all Concrete properties as shown in the **Materials.emd** file and the $\mathbf{f}_{ck}$ can be selected.

```
IO.Beam.CrossSection.Material.EC.fck
```

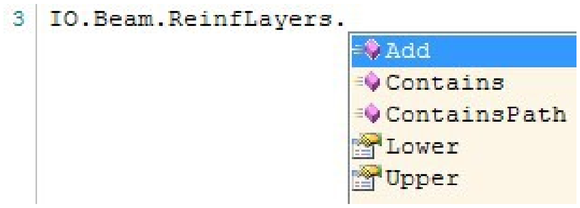
- In the same way the Reinforcement material can be accessed. On a new line type **IO** followed by a dot.



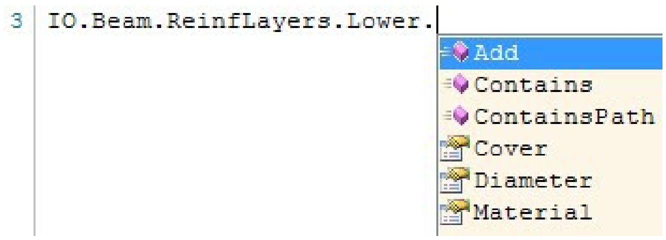
- Select **Beam** from the list and again type a dot.



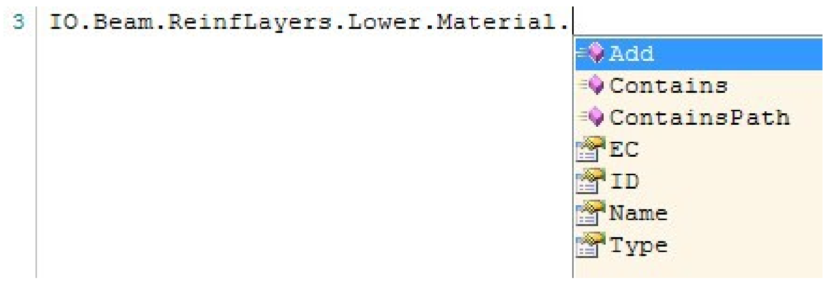
- Select **ReinfLayers** from the list and again type a dot.



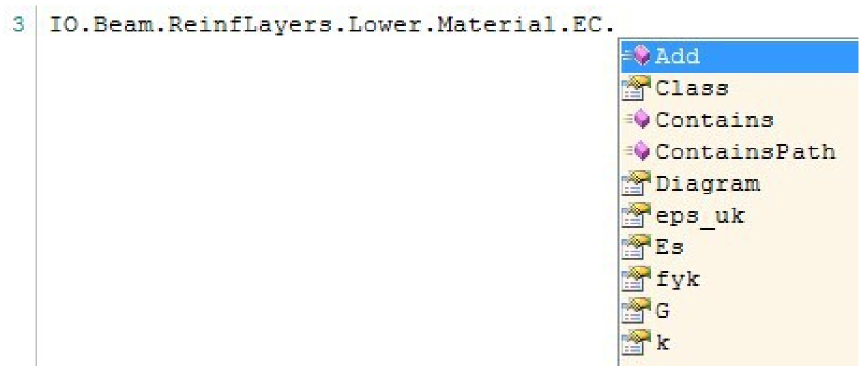
- Select **Lower** from the list and again type a dot.



- Now select **Material** again followed by a dot.



- Select **EC** followed by a dot.



Here we now see all **Reinforcement Steel** properties as shown in the **Materials.emd** file and the $\mathbf{f}_{yk}$ can be selected.

```
3  IO.Beam.ReinfLayers.Lower.Material.EC.fyk
```

This concludes the Intermezzo on the usage of the **EMD Loader**. It illustrated how to generate the **.emd** files and how to navigate through them using the **EMD Loader**.

Before continuing, please do the following:

- Within SCIA Engineer, in the **Check Manager**, change the material of the **Bending Check** back to **Steel**.

- Close SCIA Engineer.

- Within SCIA Design Forms, close the new Form containing the **EMD Loader**.

For the actual example, the Form contained the following two lines for reading the IO Structure during a standalone run:

```
426 fck = IO.CONCRETE.EC.fck;
427 fyk = IO.Reinforcement.EC.fyk;
```

The above Intermezzo has shown that, in order to read the data from the DataCache, the following references are needed:

f_{ck} = IO.Beam.CrossSection.Material.EC.fck

f_{yk} = IO.Beam.ReinfLayers.Lower.Material.EC.fyk

For both variables the IO Structure for a standalone run is different than for a check run from SCIA Engineer. When running standalone it uses the IO Structure as defined in the **Target IO node** while when running from SCIA Engineer it should be taken from the DataCache export.

It is thus required to add a test here to see which IO Structure is available and then read the appropriate data. Within the SCIA Design Form this is possible using the **IO.IsStandalone** parameter. This boolean indicates if a Form is run standalone or from within SCIA Engineer. For info on this parameter, reference is made to the Design Forms WebHelp.

The script is adapted as follows:

```
bool Test = IO.IsStandalone;

IF(Test)                    // Standalone
{
fck = IO.CONCRETE.EC.fck;
fyk = IO.Reinforcement.EC.fyk;
}
ELSE                        // Linked to Scia Engineer
{
fck = IO.Beam.CrossSection.Material.EC.fck;
fyk = IO.Beam.ReinfLayers.Lower.Material.EC.fyk;
}
```

The existing statements on **Lines 426 & 427** are thus replaced by the above code.

Using the **IO.IsStandalone** method on **Line 426** it is tested if the Form runs standalone or linked with SCIA Engineer. In case this boolean is True, both the $\mathbf{f}_{ck}$ and $\mathbf{f}_{yk}$ values are read from their respective Target IO nodes.

In case the boolean is False it means the Form is run from SCIA Engineer. In that casethe IO Structure is generated from the DataCache and thus a different structure needs to be used. The $\mathbf{f}_{ck}$ and $\mathbf{f}_{yk}$ values are in this case read from the **.emd** files in the DataCache.

After making the above changes to the Form, refresh the Form by pressing the button 

or the **F5** key.

The entire block of code from **Line 426 to 437** can be selected and its **Visibility** disabled so it is not shown on the output.

This concludes the first part of this step; all input variables for the Form which can be obtained from the DataCache are now properly linked to ESA.ID's. In addition the Form was modified to account for the difference in the IO Structure between a standalone run and a run when linked to SCIA Engineer.

Step 2.2: Set the ID for the Output (Result) data

In this example there is one result value which will be defined, the Reinforcement Area $\mathbf{A_s}$. This variable will thus be given the ID **Result.1**.

Double | String | Boolean | Structured

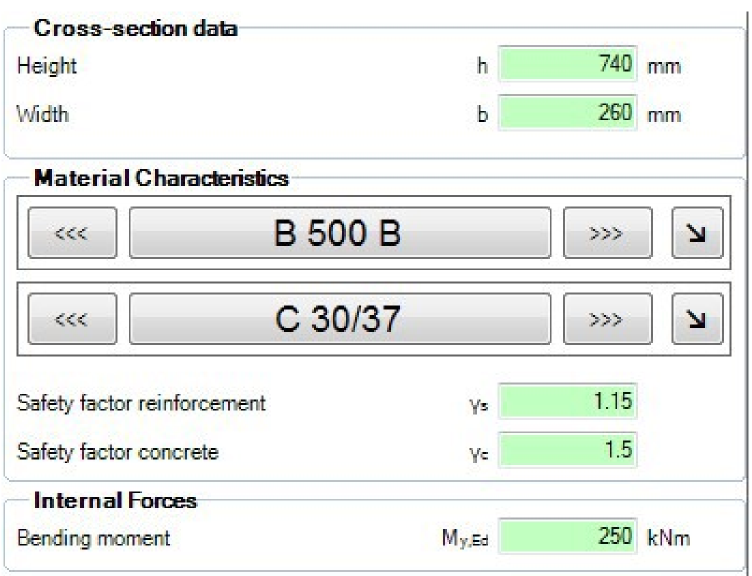
ID	Description	Symbol	Value	Unit
	Iteration step	i	0	
InternalForces.My	Bending moment	$M_{y,Ed}$	250	kNm
	Characteristic concrete compressive strength	f_{ck}	30	N/mm^2
	Characteristic reinforcement yield strength	f_{yk}	500	N/mm^2
	Safety factor concrete	γ_c	1.5	
	Safety factor reinforcement	γ_s	1.15	
	Design concrete compressive strength	f_{cd}	17	N/mm^2
	Design reinforcement yield strength	f_{yd}	435	N/mm^2
	Leverage arm	d	666	mm
CS.Geometry.B	Width	b	260	mm
	Reduced value of the bending moment	μ_d	0.128	
CS.Geometry.H	Height	h	740	mm
	Reinforcement percentage	ω_1	0.14	
Result.1	Reinforcement area	A_s	945	mm^2

In order to let the Form take into account the ID's defined in this Step refresh the form either by pressing the button ▶ Refresh or the **F5** key.

This concludes the definition of the in- and output variables.

Step 3: Optionally define Member data

Switch to the '**Dialog**' tab to review the current Dialog defined for a Standalone run of the Form:



The Standalone dialog of course contains all inputs since all of them are required for a Standalone run.

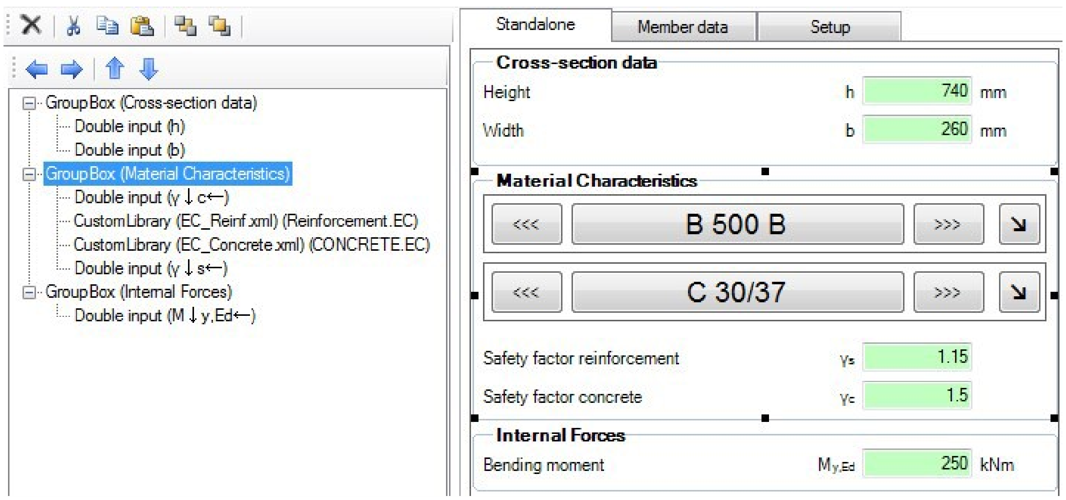
When running the Check from within SCIA Engineer, not all of these inputs should be visible. Therefore the Member data dialog is used: this dialog will be visible from within SCIA Engineer and should thus contain only the relevant information which is not set through the DataCache.

In this example, only the safety factors γ_S and γ_C should be visible in the Member data dialog.

Step 3.1 Define the Member data dialog

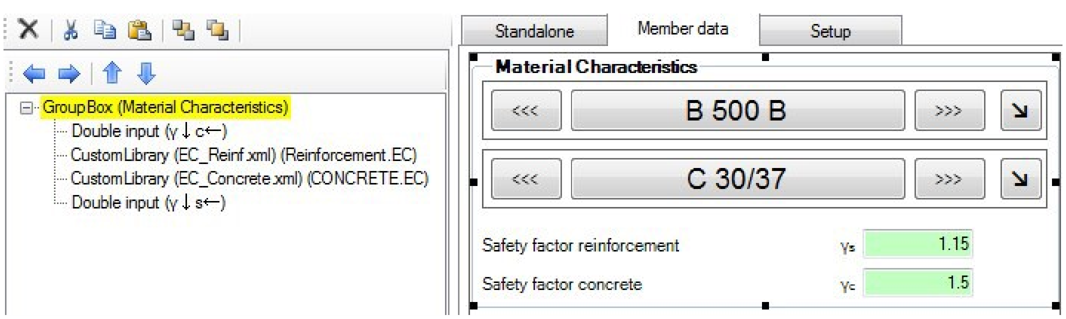
In the **Standalone** dialog, switch to the Dialog outline to see all components of the dialog.

Select the **GroupBox (Material Characteristics)**.



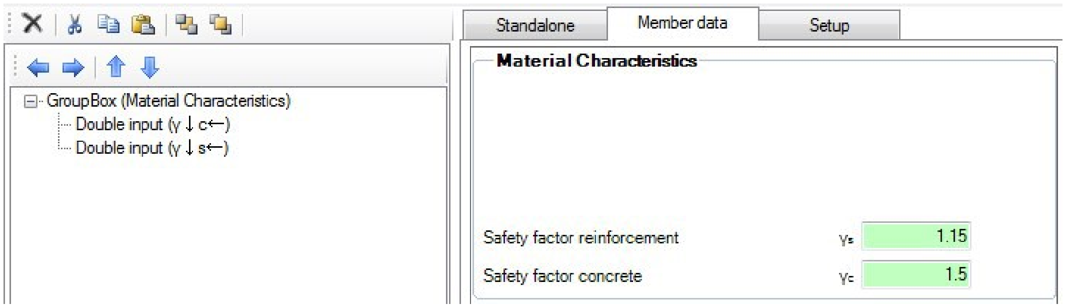
Use the Copy button to copy this **GroupBox** to the clipboard.

Now switch to the '**Member data**' tab, click anywhere in the empty tab to put the focus on that tab and use the Paste button to paste the component into this tab.



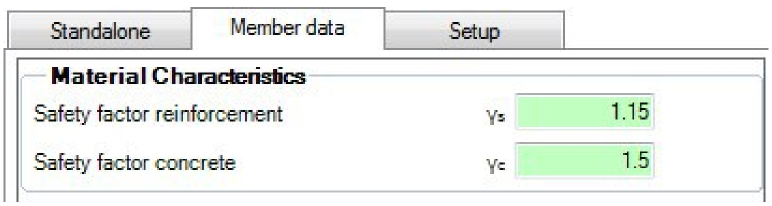
Press [Esc] to end the selection.

Now the libraries can be removed. Select the library components either in the dialog or in the Outline and press the Delete button to remove them.



Press [Esc] to end the selection.

The height of the **Material Characteristics GroupBox** can now be adjusted.



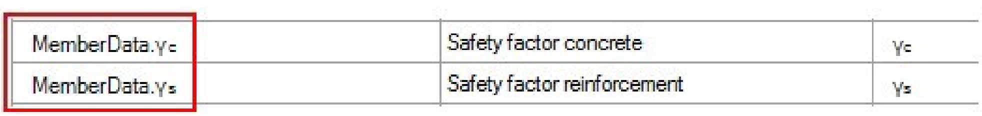
The member data dialog has now been defined. In the next part the properties within this dialog will be given ESA.ID's.

Step 3.2 Set the ID for the Member data Input data

First of all, switch to the  so the grid with input variables becomes visible again.

On the '**Double**' tab, set the ID for the safety factor γ_c to **MemberData.**γ_c.

In the same way, set the ID for the safety factor γ_s to **MemberData**.γ_s.



All properties defined on the Member data dialog now have a properly defined ID.

This concludes the definition of the Member data dialog and corresponding ID's.

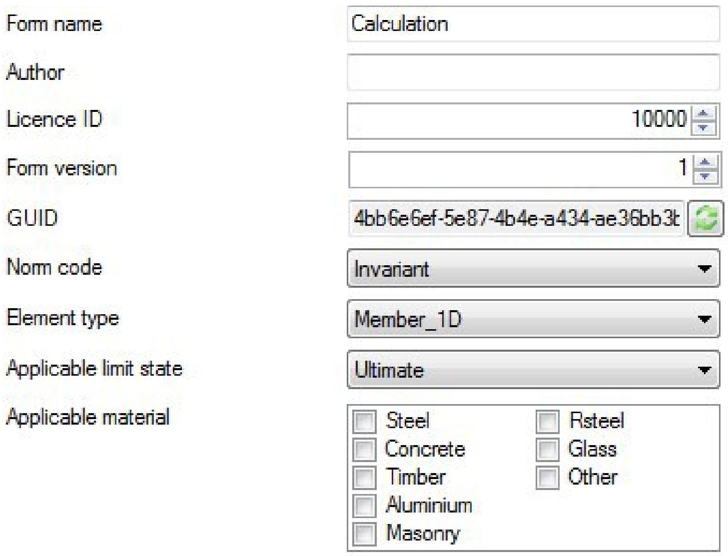
Step 4: Define the Check Header

After setting the ESA.ID's and inputting Member data the 'Check Header' can be defined.

Switch to the tab '**Header**' to define the Check Header.

Step 4.1 Define the Check Header Part 1

When initially opening the 'Header' tab the following data is shown by default:



For this example the **Form name** is set to **Concrete Reinforcement**.

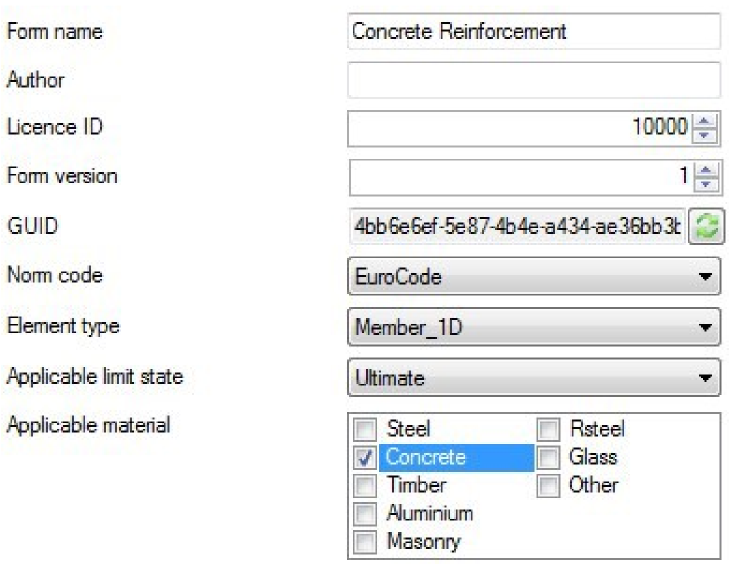
The **Norm code** is set to **EuroCode**.

The **Element type** is set to **Member 1D.**

Applicable limit state is set to **Ultimate**.

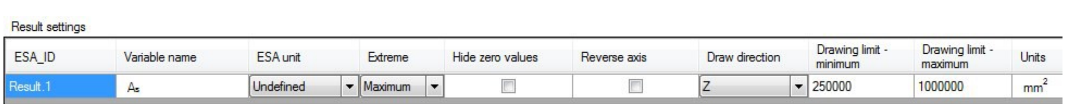
For the **Applicable material** the material **Concrete** is selected.

After the above modifications the Check Header definition looks as follows:



Step 4.2 Define the Check Header Part 2

The bottom part of the Check Header definition shows a grid which automatically lists all variables that were defined as output results in Step 1.2.



In this example the A_s variable was set as **Result.1** so this is the only variable shown here.

In this example, the A_s variable concerns a reinforcement area. The unit is therefore set to **Cross-section - Reinforcement area [m^2]**.

Result settings

ESA_ID	Variable name	ESA unit
Result.1	A_s	Cross-section - Reinforcement area [m^2]

For the drawing limits the following logic could be used: a 1φ6 rebar as minimum has an area of **28 mm²**. 10φ32 rebars as maximum have an area of **8042 mm²**.

These values can thus be set as sensible limits:

Drawing limit - minimum	Drawing limit - maximum	Units
28	8042	mm^2

The remaining settings are left on their defaults.

This concludes the definition of the Check Header; all required inputs have now been made.

Step 5: Export the Form to a .CLC

The final step within the Design Form concerns saving the changes made to the source file (.CLS) and exporting this into a .CLC file.

To this end, first save the Form using the Save button.

Next, export the file to a .CLC using the Export CLC button.

This .CLC file has the same name as the .CLS file and will be generated in the same folder as the .CLS file.

As specified in the chapter on Default folders, this file must be in the 'User folder' in order for it to be visible from within SCIA Engineer.

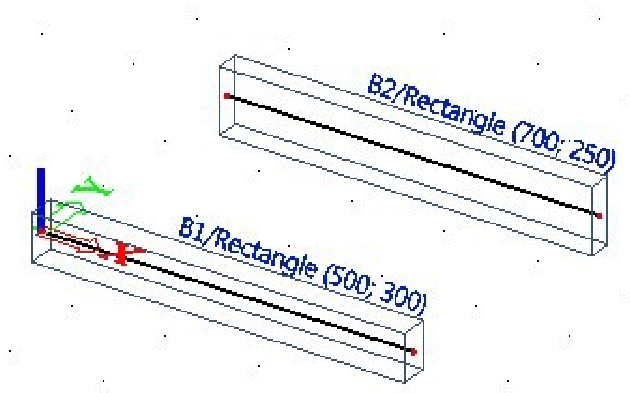
This concludes all steps needed within SCIA Design Forms. The SDF application can now be closed.

Step 6: Import the .CLC into the Check Manager

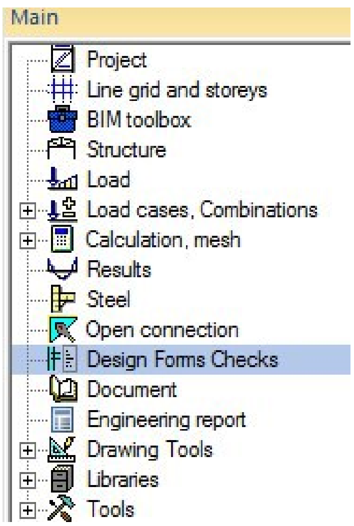
In this specific step, the .CLC file created in the previous Step will be imported within SCIA Engineer.

Launch SCIA Engineer and open the file **Manual_Example_3.esa**.

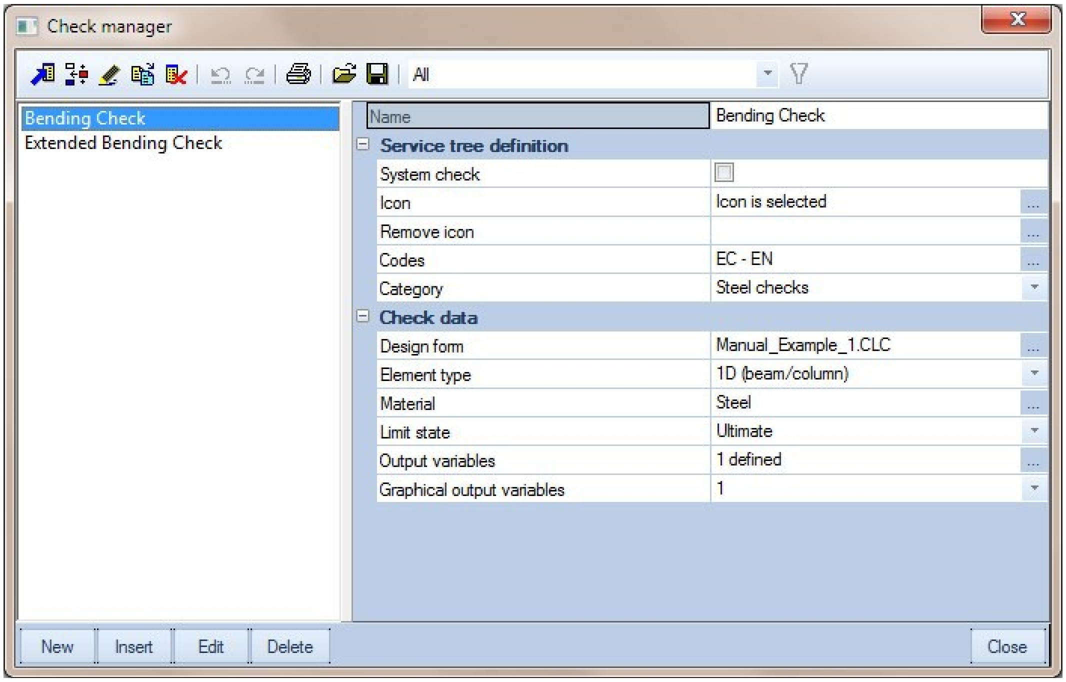
Within this example project, two concrete members with different cross-sections and materials are given:



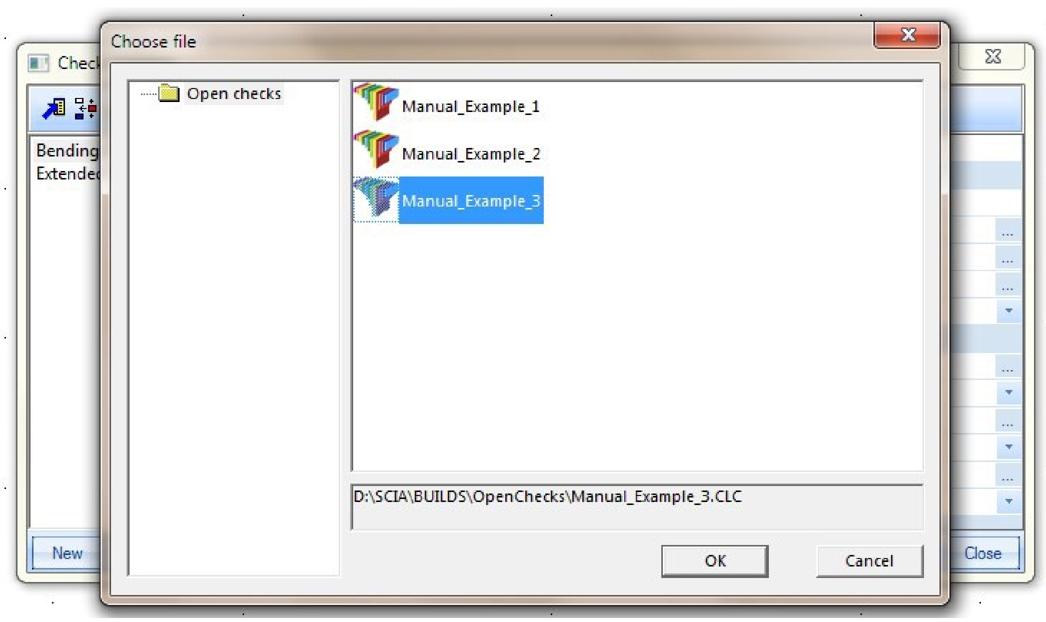
Go to the Design Forms Checks service.



In the Design Forms Checks service open the Check manager .

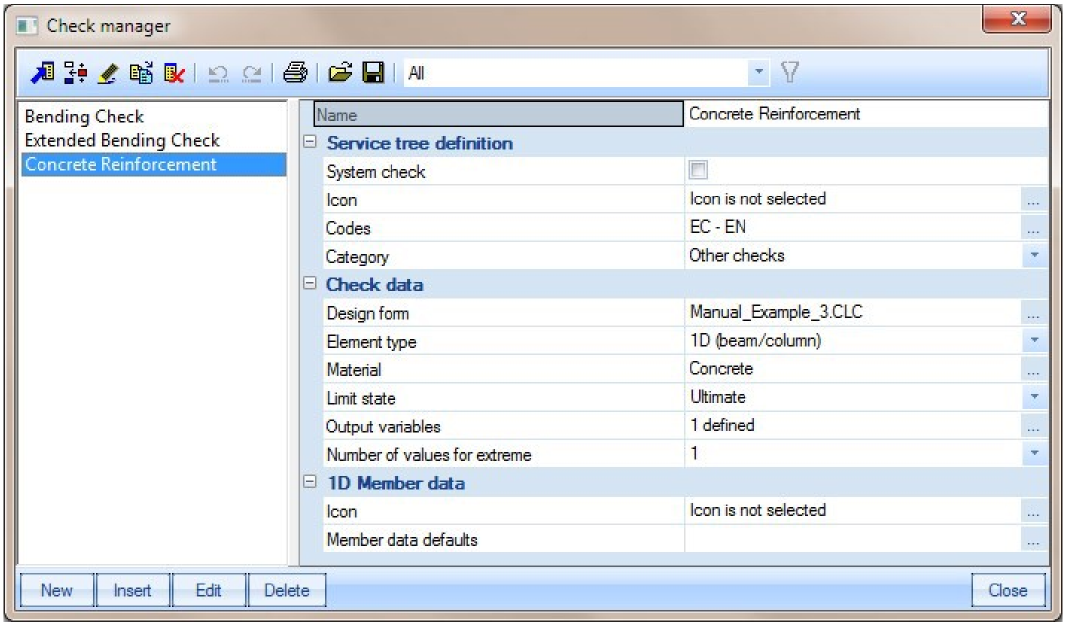


Use the New button  to define a new Check. The Choose file dialog appears showing all .CLC files in the OpenChecks folder:



Select the **Manual_Example_3.CLC** file and press [OK].

The **Check Manager** automatically imports the Check header information from the CLC:



The **Name, Codes, Element type, Material, Limit state** and **Output variables** are shown as defined in the .CLC in step 4.

In addition, since Member data was defined in this example an additional subgroup **1D Member data** is shown which contains the Member data dialog.

Step 7: Optionally edit the Check Header and provide icons

In essence, merely importing the .CLC is already sufficient since all required data has been defined.

It is however possible to add additional info to the definition of the check or edit the data which was defined in the Form.

Step 7.1: Service tree definition

The **Service tree definition** shows the info which will be used to display the check in the tree of SCIA Engineer.

For this example the Category will be set to **Concrete checks.**

Step 7.2: Check data

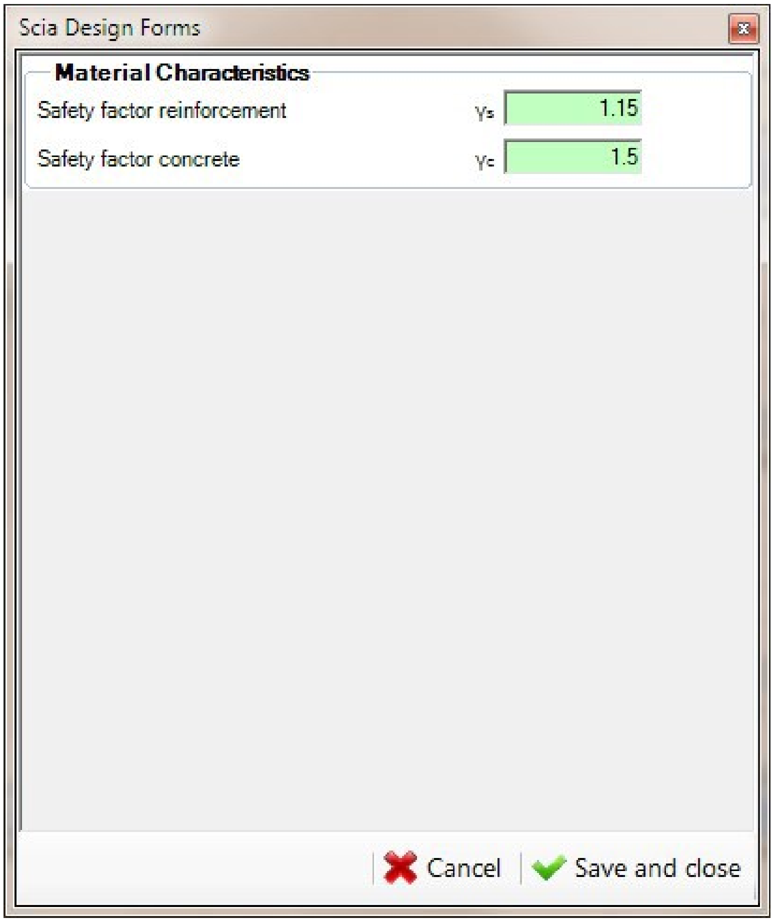
The **Check data** subgroup shows all data relevant to the check.

In this example no changes are made to the imported data.

Step 7.3 Member data

Since Member data was defined in the .CLC file a new subgroup **1D Member data** is shown.

The **Member data defaults** setting is used to display the dialog defined in the.CLC file in Step 3 directly within SCIA Engineer. Press the ... button to review the Member data dialog:



This dialog shows the defaults which are used by the Form. Any member which does not have Member data defined will get these values as a default.

Changing the defaults themselves is something which needs to be done within the .CLC file (see Step 2). The dialog shown here is merely a 'Viewer'.

Close the defaults dialog by either Cancel or Save and close since for this dialog both will just close the dialog. In Step 9 the actual input of Member data is discussed which allows editing this data for specific members.

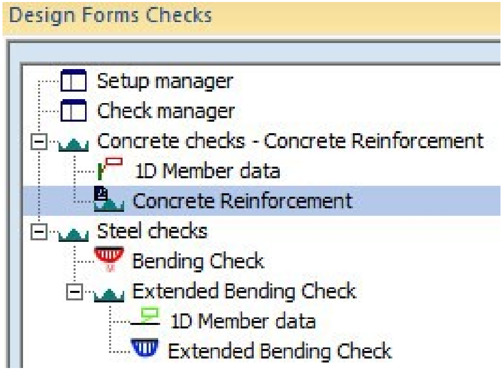
The Check Manager can now be closed by pressing [Close].

Step 8: Execute the Check

All preparations have been made; the Check is linked to SCIA Engineer and can now be executed.

Step 8.1: Display the Brief and Graphical output

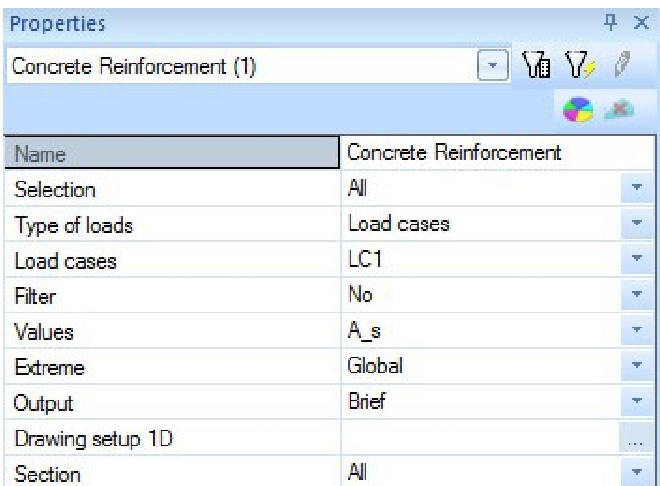
After closing the **Check Manager** the new check will be shown in the **Design Forms Checks** service:



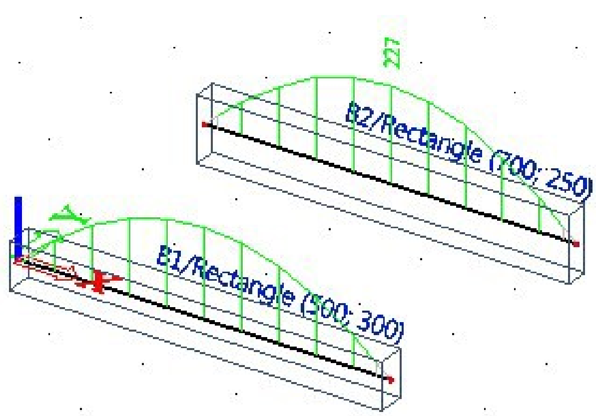
As with all check services in SCIA Engineer, checks are only visible in case the project is calculated. So if the new check is not shown, perform a linear calculation.

The tree shows the new subgroup **Concrete Checks** which contains the **Concrete Reinforcement**. Since in this case no icons were assigned in the Check Manager, default icons are being used.

Select the Concrete Reinforcement in the service. The Property window will show the properties of the check:



Press the **Refresh** action button to run the check.



Press the **Preview** action button to display the preview of the Brief check:

Concrete Reinforcement

Type Name	Name	Header	Concrete Reinforcement
Concrete Reinforcement	Concrete Reinforcement	Linear calculation, Extreme : Global Selection : All Load cases : LC1	Concrete Reinforcement

Type Name	Member	Position [m]	Case	A_{s2} [mm^2]
Concrete Reinforcement	B2	2,000	LC1	**227**

Use the **Table composer** as explained in Example 1 to change the layout of the output:

Concrete Reinforcement

Linear calculation, Extreme : Global
Selection : All
Load cases : LC1
Concrete Reinforcement

Member	Position [m]	Case	A_s [mm^2]
B2	2,000	LC1	**227**

Step 8.2: Display the Detailed output

In the Property window change the **Output** to **Detailed**.

Press the **Refresh** action button to run the check.

Concrete Reinforcement

Linear calculation, Extreme : Global
Selection : All
Load cases : LC1

Type Name	Detailed output Page: 000
Concrete Reinforcement	**Example 3: Concrete Reinforcement** **Section Properties** $b = 250$ mm $h = 700$ mm **Internal Forces** Bending moment $M_{y,Ed} = 60$ kNm **Material Characteristics** $f_{ck} = 40$ N/mm^2 $\gamma_c = 1,5$ $f_{yk} = 500$ N/mm^2 $\gamma_s = 1,15$ **Verification** $f_{cd} = \frac{0.85 \cdot f_{ck}}{\gamma_c} = \frac{0.85 \cdot 40 \cdot 10^6}{1.5} = 22,7$ N/mm^2 $f_{yd} = \frac{f_{yk}}{\gamma_s} = \frac{500 \cdot 10^6}{1,15} = 435$ N/mm^2 $d = 0.9 \cdot h = 0.9 \cdot 0,7 = 630$ mm $\mu_d = \frac{M_{y,Ed}}{b \cdot d^2 \cdot f_{cd}} = \frac{60000}{0,25 \cdot 0,63^2 \cdot 22,7 \cdot 10^6} = 0,0267$ $\Rightarrow \omega_1 = 0,028$ $A_s = \omega_1 \cdot \frac{f_{cd}}{f_{yd}} \cdot b \cdot d = 0,0277 \cdot \frac{22,7 \cdot 10^6}{435 \cdot 10^6} \cdot 0,25 \cdot 0,63 = 227$ mm^2

Use the **Table composer** as explained in Example 1 to change the layout of the output:

Concrete Reinforcement

Linear calculation, Extreme : Global
Selection : All
Load cases : LC1

Example 3: Concrete Reinforcement

Section Properties

b = 250 mm
h = 700 mm

Internal Forces

Bending moment $M_{y,Ed}$ = 60 kNm

Material Characteristics

f_{ck} = 40 N/mm^2

γ_c = 1,5

f_{yk} = 500 N/mm^2

γ_s = 1,15

Verification

$$f_{cd}=\frac{0.85 \cdot f_{ck}}{\gamma_c}=\frac{0.85 \cdot 40*10^6}{1,5}=22,7\ \text{N/mm}^2$$

$$f_{yd}=\frac{f_{yk}}{\gamma_s}=\frac{500*10^6}{1,15}=435\ \text{N/mm}^2$$

$$d=0.9 \cdot h=0.9 \cdot 0,7=630\ \text{mm}$$

$$\mu_d=\frac{M_{y,Ed}}{b \cdot d^2 \cdot f_{cd}}=\frac{60000}{0,25 \cdot 0,63^2 \cdot 22.7*10^6}=0,0267$$

$$=> \omega_1=0,028$$

$$A_s=\omega_1 \cdot \frac{f_{cd}}{f_{yd}} \cdot b \cdot d=0,0277 \cdot \frac{22.7*10^6}{435*10^6} \cdot 0,25 \cdot 0,63=227\ \text{mm}^2$$

Step 9: Evaluate the Results

With the link and output completed, the results can now be evaluated for different inputs as well as the Member data.

Set the **Selection** to **Current**, select member **B1** and press the **Refresh** action button:

Concrete Reinforcement

Linear calculation, Extreme : Global
Selection : B1
Load cases : LC1

Example 3: Concrete Reinforcement

Section Properties

b = 300 mm
h = 500 mm

Internal Forces

Bending moment $M_{y,Ed}$ = 40 kNm

Material Characteristics

$f_{ck} = 25\ N/mm^2$

$\gamma_c = 1{,}5$

$f_{yk} = 500\ N/mm^2$

$\gamma_s = 1{,}15$

Verification

$$f_{cd} = \frac{0.85 \cdot f_{ck}}{\gamma_c} = \frac{0.85 \cdot 25*10^6}{1{,}5} = 14{,}2\ N/mm^2$$

$$f_{yd} = \frac{f_{yk}}{\gamma_s} = \frac{500*10^6}{1{,}15} = 435\ N/mm^2$$

$$d = 0.9 \cdot h = 0.9 \cdot 0{,}5 = 450\ mm$$

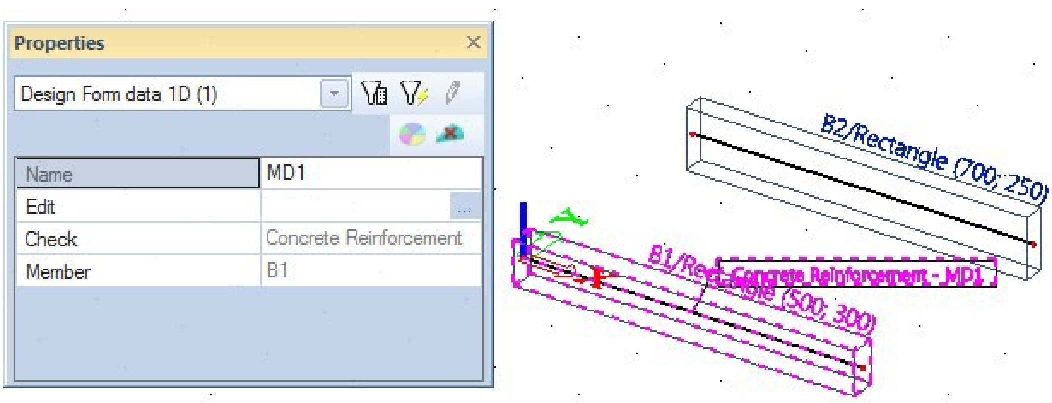
$$\mu_d = \frac{M_{y,Ed}}{b \cdot d^2 \cdot f_{cd}} = \frac{40000}{0{,}3 \cdot 0{,}45^2 \cdot 14.2*10^6} = 0{,}0465$$

$$=> \omega_1 = 0{,}048$$

$$A_s = \omega_1 \cdot \frac{f_{cd}}{f_{yd}} \cdot b \cdot d = 0{,}0485 \cdot \frac{14.2*10^6}{435*10^6} \cdot 0{,}3 \cdot 0{,}45 = 213\ mm^2$$

The output shows the section characteristics, internal forces and material properties of this member. Since no Member data was inputted the default values are used for γ_c and γ_s as defined in the .CLC file and shown in Step 7.3.

Now double click on the 1D Member data in the tree. Confirm the dialog which appears with [OK] and input this data on member **B1**.



The Property window shows the **Name** of the Member data as well as the **Check** to which it belongs.

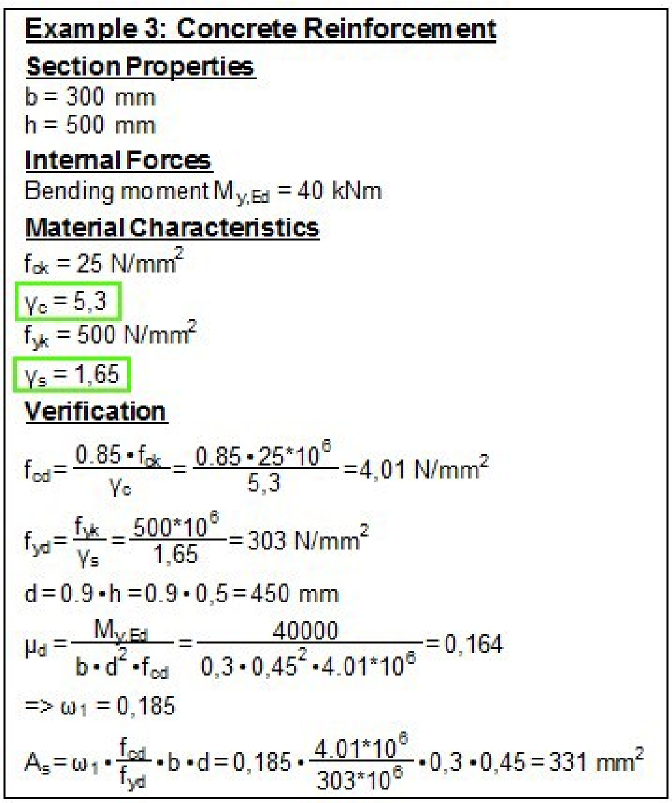
The viewflag shown on member **B1** automatically concatenates this information so it's visually always clear to which Check the data belongs.

Click on the ... button in the Property window to **Edit** the Member data and display the Member data dialog defined in the .CLC:

Change the safety factor γ_s for example to **1.65** and the γ_c to **5.3**.

After changing the values, click somewhere in the dialog outside of the input fields to confirm the data and close the dialog using the Save and close button.

Select the Concrete Reinforcement in the service and press the **Refresh** action button to execute the check once more:



Example 3: Concrete Reinforcement

Section Properties

b = 300 mm
h = 500 mm

Internal Forces

Bending moment $M_{y,Ed}$ = 40 kNm

Material Characteristics

f_{ck} = 25 N/mm^2
γ_c = 5,3
f_{yk} = 500 N/mm^2
γ_s = 1,65

Verification

$$f_{cd} = \frac{0.85 \cdot f_{ck}}{\gamma_c} = \frac{0.85 \cdot 25*10^6}{5,3} = 4,01 \text{ N/mm}^2$$

$$f_{yd} = \frac{f_{yk}}{\gamma_s} = \frac{500*10^6}{1,65} = 303 \text{ N/mm}^2$$

$$d = 0.9 \cdot h = 0.9 \cdot 0,5 = 450 \text{ mm}$$

$$\mu_d = \frac{M_{y,Ed}}{b \cdot d^2 \cdot f_{cd}} = \frac{40000}{0,3 \cdot 0,45^2 \cdot 4.01*10^6} = 0,164$$

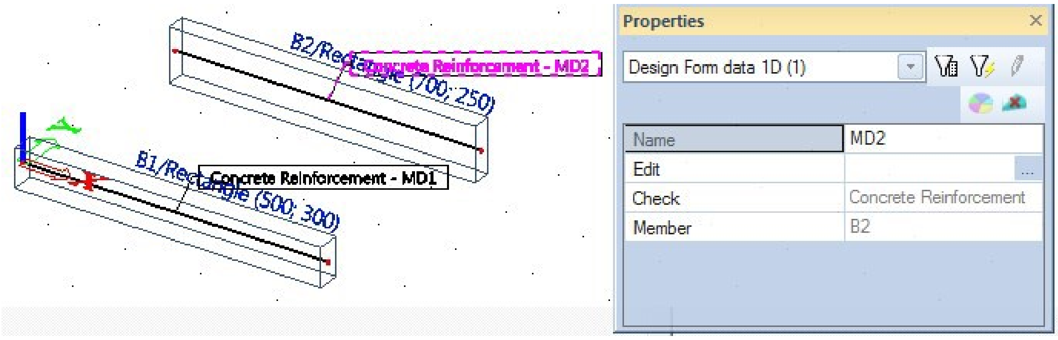
$$\Rightarrow \omega_1 = 0,185$$

$$A_s = \omega_1 \cdot \frac{f_{cd}}{f_{yd}} \cdot b \cdot d = 0,185 \cdot \frac{4.01*10^6}{303*10^6} \cdot 0,3 \cdot 0,45 = 331 \text{ mm}^2$$

The output shows the modified values as defined in the Member data.

Press [Esc] to end the selection.

Now double click on the 1D Member data in the tree. Confirm the dialog which appears with [OK] and input this data on member **B2**.



Click on the ... button in the Property window to **Edit** the Member data and display the Member data dialog defined in the .CLC:

Change the safety factor γ_S for example to **6.2** and the γ_C to **1.4**.

After changing the values, click somewhere in the dialog outside of the input fields to confirm the data and close the dialog using the Save and close button.

Press [Esc] to end the selection.

Select member **B2**.

Select the Concrete Reinforcement in the service and press the **Refresh** action button to execute the check once more:

Example 3: Concrete Reinforcement

Section Properties

b = 250 mm
h = 700 mm

Internal Forces

Bending moment $M_{y,Ed}$ = 60 kNm

Material Characteristics

f_{ck} = 40 N/mm^2
γ_c = 1,4
f_{yk} = 500 N/mm^2
γ_s = 6,2

Verification

$$f_{cd} = \frac{0.85 \cdot f_{ck}}{\gamma_c} = \frac{0.85 \cdot 40*10^6}{1,4} = 24,3\ N/mm^2$$

$$f_{yd} = \frac{f_{yk}}{\gamma_s} = \frac{500*10^6}{6,2} = 80,6\ N/mm^2$$

$$d = 0.9 \cdot h = 0.9 \cdot 0,7 = 630\ mm$$

$$\mu_d = \frac{M_{y,Ed}}{b \cdot d^2 \cdot f_{cd}} = \frac{60000}{0,25 \cdot 0,63^2 \cdot 24.3*10^6} = 0,0249$$

=> ω_1 = 0,026

$$A_s = \omega_1 \cdot \frac{f_{cd}}{f_{yd}} \cdot b \cdot d = 0,0259 \cdot \frac{24.3*10^6}{80.6*10^6} \cdot 0,25 \cdot 0,63 = 1228\ mm^2$$

The output shows the section characteristics, internal forces and material properties of this member as well as modified values of γ_C and γ_S which were set through the Member data.

This quick overview of the results illustrates how Member data can be used to modify the input data for each member.

Step 10: Optionally review the DataCache and Trace file

The final step of this example illustrates the internal transfer of data between SCIA Engineer and SCIA Design Forms.

Before going through this step please read the information contained within Annex A which explains the Command line switches and generation of the **.trace** file.

- First of all, close SCIA Engineer. The changes to the project file can be **saved** so the inputted member data is also stored within the project.

- Next, launch SCIA Engineer using the Command line switch **-OCFILES**.

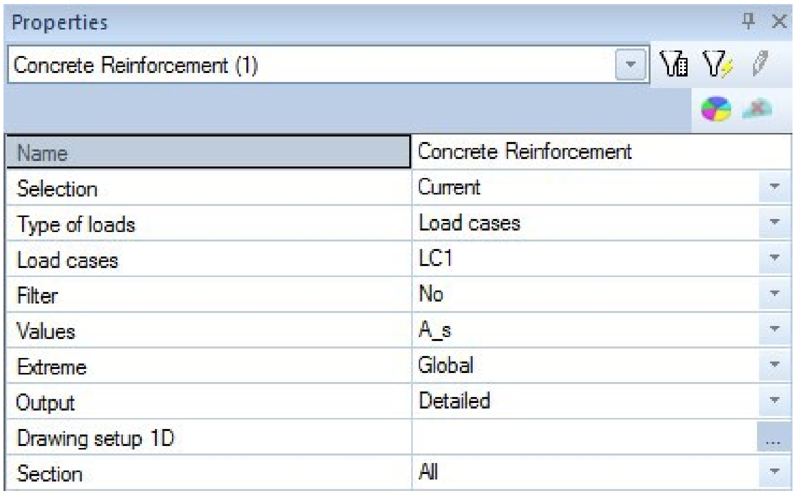
- Open the file **Manual_Example_3.esa**.

- Perform a Linear calculation to refresh the results.

- Go to the Design Forms Checks service.

- Select the Concrete Reinforcement in the service.

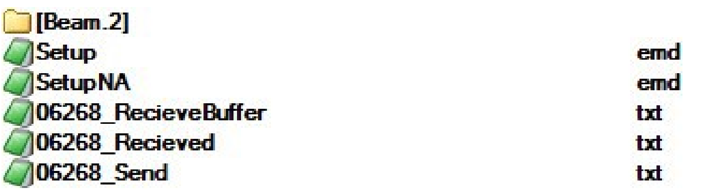
- Set the following Properties for the Check:



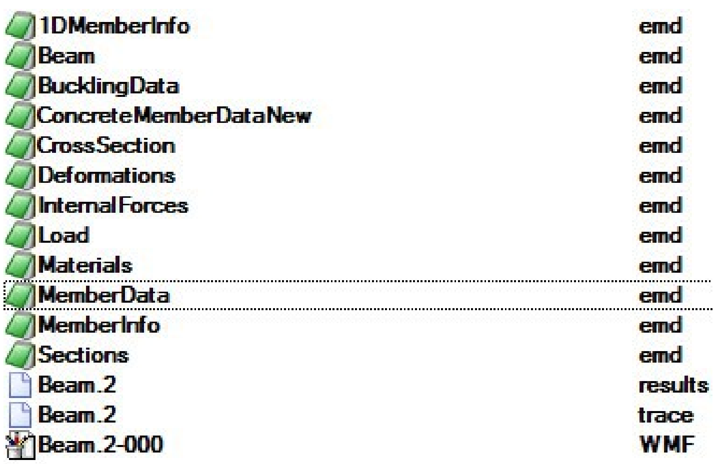
- Select member **B2**

- Press the **Refresh** action button to run the check

- Using a file explorer like **Windows Explorer**, navigate to the **TEMP** folder of SCIA Engineer and locate the subfolder **ESA_Model_Data**



Now go to the subfolder **Beam.2**. This folder contains all input data (**.emd**), output data (**.results**) and the **.trace** file.



When looking inside the **MemberData.emd** file using a standard text editor like **NotePad** the settings for the Member data for this beam can be found:

```
(:MemberData :γ↓s← 6.2 :γ↓c← 1.4 )
```

This data is thus transferred to the DataCache using the ID's as defined in Step 3.2.

The **CrossSection.emd** file for example contains the geometry of the section:

```
(:CrossSection :ID 1 :MaterialID 321 :Name "CS2" :Description "Rectangle" )
  (:IsPhased 0 )
  (:Geometry :FormCode 2003 :B 0.25 :H 0.7 )
  (:Chars :A 0.175 :Ay 0.145833 :Az 0.145833 :AL 1.9 :AD 1.9 :cYUCS 0.125 :cZUCS
```

The following picture shows the top part of the **.trace** file content which shows the data read from the IO structure:

```
--------------------------------------------------------------------------------
MEMBER = Beam.2; SECTION = 5; COMBI = 0; LAYOUT = 0; LANGUAGE = 0; PAGING = True
--------------------------------------------------------------------------------
Reading values from IO: {
  M↓y,Ed← = 60000
  γ↓c← = 1,4
  γ↓s← = 6,2
  b = 0,25
  h = 0,7
}
```

The bottom part of the **.trace** file shows the different parts of the check including the run of the **Get_ω1** function.

```
TEXT(Example 3: Concrete Reinforcement)
TEXT(Section Properties)
TEXT(b = 250 mm)
TEXT(h = 700 mm)
TEXT(Internal Forces)
TEXT(Bending moment M↓y,Ed← = 60 kNm)
Test = False
IF (False) {
  f↓ck← = 40*10↑6←
  f↓yk← = 500*10↑6←
}
TEXT(Material Characteristics)
TEXT(f↓ck← = 40 N/mm↑2←)
TEXT(γ↓c← = 1,4)
TEXT(f↓yk← = 500 N/mm↑2←)
TEXT(γ↓s← = 6,2)
TEXT(Verification)
f↓cd← = 24.3*10↑6←
f↓yd← = 80.6*10↑6←
d = 0,63
μ↓d← = 0,0249
Get_ω↓1←(0,0249) {
  i = 0
  WHILE () {
    WHILE CYCLE #0
    IF (False) {
    }
    IF (False) {
    }
    IF (False) {
    }
```

Close SCIA Engineer and discard any changes made.

This concludes the third example which illustrated the use of Libraries and showed how the DataCache and EMD Loader can be used to walk through the IO structure. In addition the example illustrated how a script can be modified to distinguish between the IO structure for a standalone run and a run from within SCIA Engineer.

Annex A: Data Transfer and Command line switches

This Annex gives background information on the way how data is transferred between SCIA Engineer and SCIA Design Forms as well as additional Command line switches which can be used with SCIA Engineer to get more in depth info and perform troubleshooting.

Data Transfer

When executing a Design Forms Check from within SCIA Engineer, all data is transferred to the **DesignForms_CalcExe.exe** process which in turn launches the selected .CLC file.

The communication between SCIA Engineer and the **DesignForms_CalcExe.exe** process is done through TCP/IP.

For this communication, SCIA Engineer searches for a free port in the range **8000-9000**.

The main advantage of the TCP/IP protocol is speed; compared to a communication through a file system this protocol gives a much higher speed when executing checks.

It does imply of course that **DesignForms_CalcExe.exe** should not be blocked by any firewall in order to make sure the communication is not prohibited.

Command line Switches

To obtain more information on the data transfer and do in-depth troubleshooting a set of Command line switches has been foreseen for the SCIA Engineer process (**esa.exe**).

The following switches are available:

esa.exe -OCLOG

esa.exe -OCFILES

esa.exe -OCEMD

esa.exe -OCRESULTS

These switches can either be used from a Command line interface or added to the SCIA Engineer shortcut.

-OCEMD

This switch generates only the **.emd** files. No calculations are processed, but a batch of commands is logged into the file "Commands.txt". This file can be loaded by **DesignForms_CalcExe.exe's** internal client and used for a manual run of the check to generate **.results** files.

-OCFILES

Using the **-OCFILES** switch, instead of TCP/IP the communication is done through files. This allows a detailed overview of all information which is sent to and from the Design Form and also provides a detailed trace file to see exactly what happens in the Form. This transfer is of course slower than TCP/IP.

In addition, the same log files as generated during **-OCLOG** are generated.

After executing the check, a new subfolder **ESA_Model_Data** will be created in the SCIA Engineer **TEMP** folder.

This folder contains a subfolder for each member which was checked:

[Beam.1]	
[Beam.2]	
[Beam.3]	
[Beam.4]	
Setup	emd
07408_RecieveBuffer	txt
07408_Recieved	txt
07408_Send	txt
09372_RecieveBuffer	txt
09372_Recieved	txt
09372_Send	txt
09468_RecieveBuffer	txt
09468_Recieved	txt
09468_Send	txt
09924_RecieveBuffer	txt
09924_Recieved	txt
09924_Send	txt

The numbers used here to distinguish the members are the ID's of the members, not the beam names.

Each subfolder contains a set of **.emd** files which contains the actual data for the transfer to SCIA Design Forms:

Beam	emd
BucklingData	emd
CrossSection	emd
Deformations	emd
InternalForces	emd
Load	emd
Materials	emd
MemberDataNew	emd
Sections	emd
Beam.1	results
Beam.1	trace

All files in this folder can be viewed using a standard text editor like **NotePad**.

For example, the Materials.emd file contains all relevant material data:

```
(:Material :Type eSteel :ID 152 )
  (:EC :E 2.1e+011 :G 8.07692e+010 :fy 2.35e+008 :fu 3.6e+008 )
(:Material :Type eRsteel :ID 241 )
  (:EC :E 2e+011 :G 8.33333e+010 :fy 0 :fu 0 :fyk 4e+008 :eps_uk 0.025 )
    (:Diagram :Type ULS )
      (:Point :sig -3.65217e+008 :eps -0.025 )
      (:Point :sig -3.64649e+008 :eps -0.0225 )
      (:Point :sig -3.47826e+008 :eps -0.00173913 )
      (:Point :sig 0 :eps 0 )
      (:Point :sig 3.47826e+008 :eps 0.00173913 )
      (:Point :sig 3.64649e+008 :eps 0.0225 )
      (:Point :sig 3.65217e+008 :eps 0.025 )
```

This file also shows the Input/Output structure of the DataCache. For example, to read the yield strength fy of the steel material the following ID is used: **Material.EC.fy**

```
(:Material :Type eSteel :ID 152 )
  (:EC :E 2.1e+011 :G 8.07692e+010 :fy 2.35e+008 :fu 3.6e+008 )
```

These files can be used to see all the data which is available to transfer between SCIA Engineer and SCIA Design Forms.

Annex B gives an overview of the basic ESA ID's. A complete overview can be found in the SCIA Design Forms WebHelp.

In addition to the **.emd** files the folder also contains a **.results** file and a **.trace** file.

The **.results** file shows the result values for each section for the given load case / combination.

```
(:Res :Section 0 :Combi 0 :Values 175.9203e-3; )
(:Res :Section 1 :Combi 0 :Values 118.0346e-3; )
(:Res :Section 2 :Combi 0 :Values 60.14902e-3; )
(:Res :Section 3 :Combi 0 :Values 2.263314e-3; )
(:Res :Section 4 :Combi 0 :Values 55.62224e-3; )
(:Res :Section 5 :Combi 0 :Values 84.56404e-3; )
(:Res :Section 6 :Combi 0 :Values 84.56617e-3; )
(:Res :Section 7 :Combi 0 :Values 113.508e-3; )
(:Res :Section 8 :Combi 0 :Values 171.3935e-3; )
(:Res :Section 9 :Combi 0 :Values 229.2788e-3; )
(:Res :Section 10 :Combi 0 :Values 287.1652e-3; )
(:Res :Section 11 :Combi 0 :Values 345.0515e-3; )
```

The **.trace** file gives a full trace of the Design Form when it's run from SCIA Engineer. It shows the data which is read from the DataCache, provides a rundown of all steps followed in the Form and also indicates possible errors during the analysis.

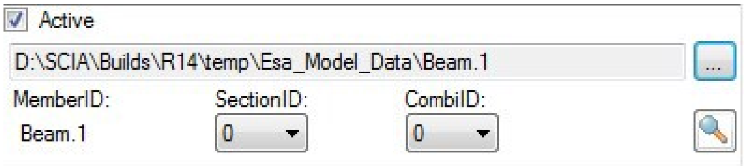
```
-------------------------------------------
MEMBER = Beam.1; SECTION = 0; COMBI = 0
-------------------------------------------
Reading values from IO: {
  A = 0,0117
  Av = 5.01*10↑-3←
  W↓el,y← = 2.19*10↑-3←
  x = 0
  N↓Ed← = -90164
  V↓z,Ed← = -53697
  M↓y,Ed← = 90661
  WARNING: Unable to find 'Material.General.fy': '.Material' does not contain property 'General' !
  γ↓M0← = 1,25
  Ignore_Shear = False
  Bolt = 0
}
TEXT(Elastic verification according to EN 1993-1-1 article 6.2.9.2 Formula (6.42).)
TEXT(Section properties)
TEXT(A = 11700 mm↑2←)
TEXT(Av = 5013,08 mm↑2←)
TEXT(Wel = 2193000 mm↑3←)
TEXT(Internal Forces at section position x = 0 m)
TEXT(Axial force N↓Ed← = -90,1637 kN)
TEXT(Axial force V↓z,Ed← = -53,697 kN)
TEXT(Axial force M↓y,Ed← = 90,6614 kNm)
TEXT(Material Characteristics)
TEXT(f↓y← = 235 N/mm↑2←)
TEXT(γ↓M0← = 1,25)
TEXT(Settings)
IF (False) {
  TEXT(Ignore Shear? No)
}
SWITCH() {
  Bolt_name = M12
}
TEXT(Selected bolt diameter: M12)
TEXT(Verification)
ρ = 0
IF (True) {
  TEXT(Shear reduction)
  V↓pl,Rd← = 544129
  UC↓Shear← = 0,0987
  IF (False) {
    IF (True) {
      TEXT(=> No reduction for shear is required)
```

In case the check does not provide the expected results, the **.trace** file will indicate where the issue is. The above picture for example indicates that a certain ID could not be found in the IO structure.

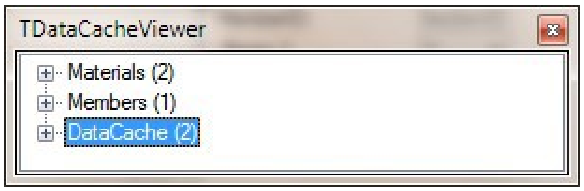
EMD Loader

Within SCIA Design Forms the **EMD Loader** component can be added into the Stand alone dialog.

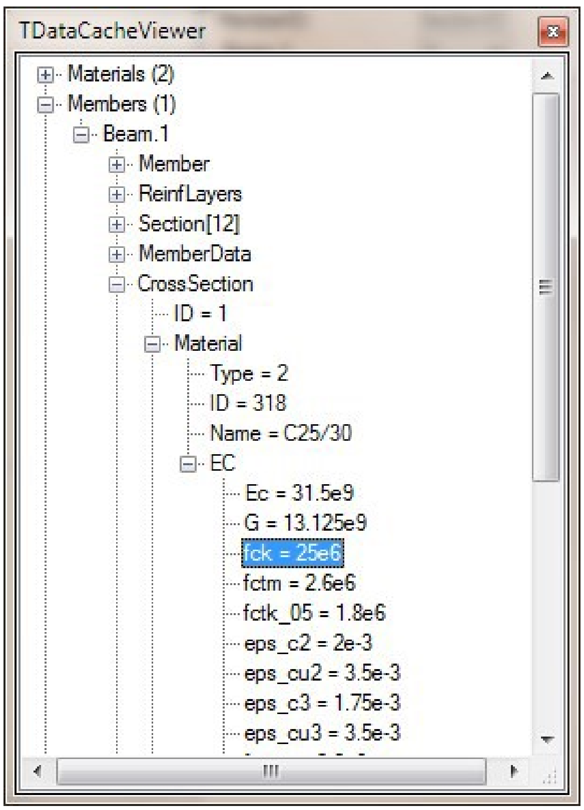
The **EMD Loader** is used to navigate directly through the content of the DataCache provided by the **OCFILES** switch.



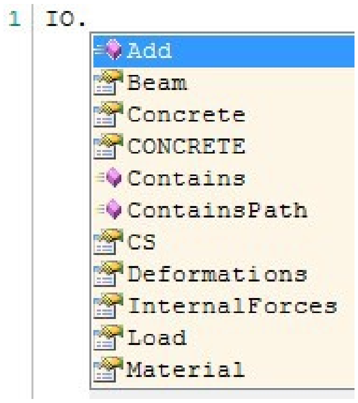
After defining the path of the DataCache, as outlined above, the Spyglass button can be used  to see the contents of the DataCache:

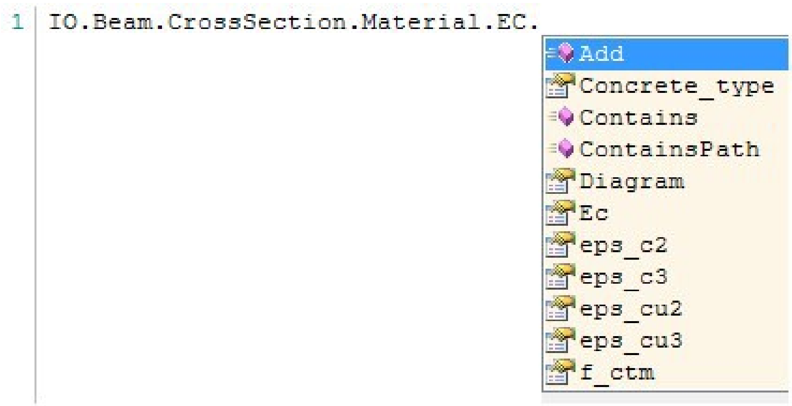


- The tree structure can now be used to navigate through the DataCache:



- In the editor, when typing **IO.** (Including the dot) an Intellisense window will be shown which shows the content of the DataCache:





The **EMD Loader** has loaded the content of the **.emd** files into the IO Structure so it can be directly accessed here.

The **EMD Loader** thus makes it possible to walk through the content of the DataCache to see exactly what is available and 'visible' from within SCIA Design Forms.

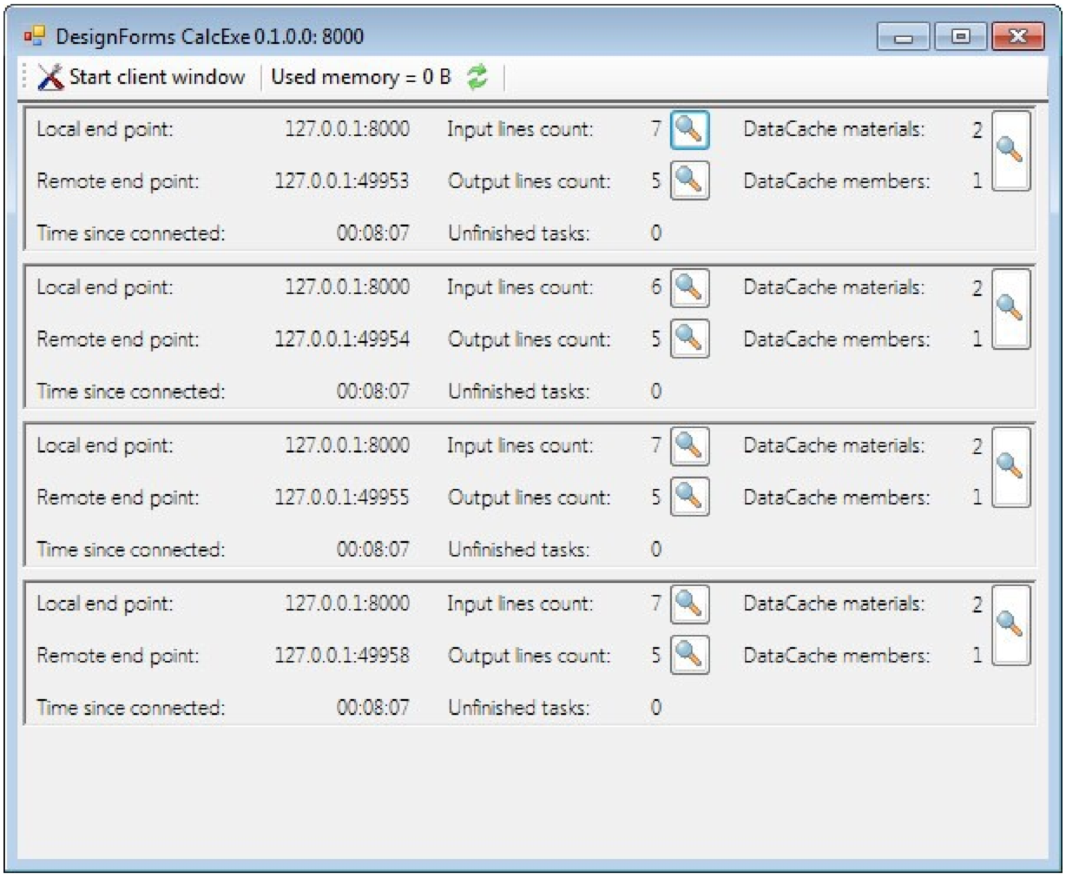
-OCLOG

Using this switch, when executing the check for each thread log files ("Send.txt", "Recieved.txt", "RecieveBuffer.txt") are generated showing the commands and data sent to and received from **DesignForms_CalcExe.exe**. These files all start with a number equal to their thread number which functions as a unique ID.

The Buffer file contains data which is not parsed, typically after the results were obtained (like 'end of calculation').

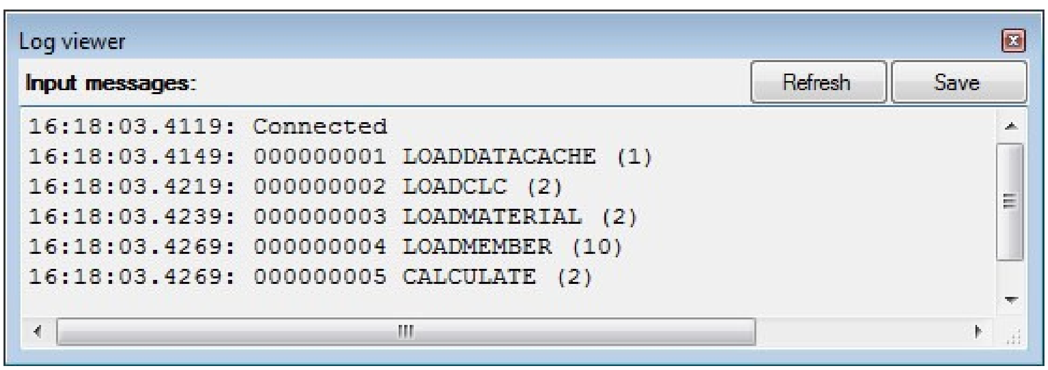
The log files are generated in a new subfolder **ESA_Model_Data** in the SCIA Engineer **TEMP** folder.

This switch keeps **DesignForms_CalcExe.exe** visible and open after the Check is finished to allow the user to work with it.

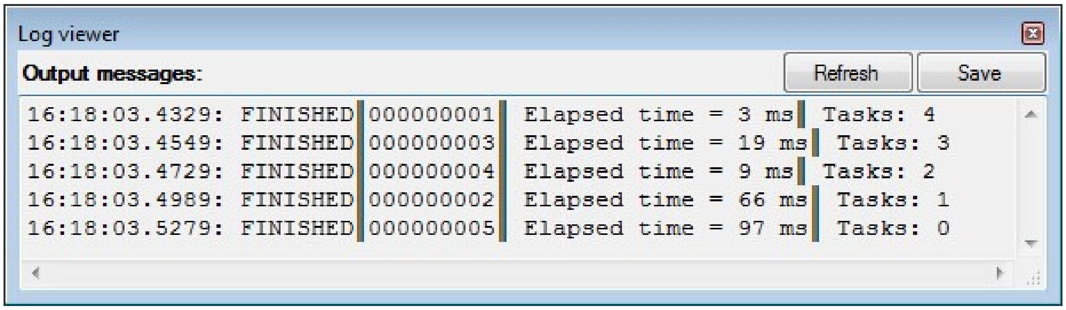


Using the spyglass  buttons the data from the log files can be visualized.

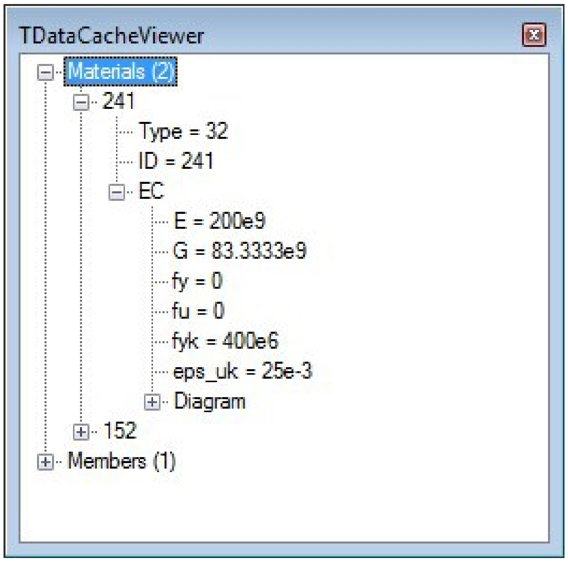
The following picture shows an example of the input commands sent through TCP/IP to Calcexe:



The following picture shows an example of the output messages sent back from Calcexe:



This dialog can also be used to see what exactly is present in the DataCache i.e. the actual data transferred between SCIA Engineer and SCIA Design Forms:



More information about the DataCache is given in the **-OCFILES** switch explanation.

-OCRESULTS

This switch uses **.results** files only for the import of check results into SCIA Engineer. No calculations or **.emd** file generation is done, but a batch of requested **.results** files is logged into the file "RequestedResults.txt".

Annex B: Basic ESA ID's and Shortcuts

This Annex provides a basic set of ESA ID's which are used to link data between SCIA Engineer and SCIA Design Forms. The second part of this Annex gives some more info on the use of shortcuts to refer to ESA ID's.

Basic ESA ID's

To obtain an overview of everything which is available in the DataCache the **-OCFILES** Command line switch can be used as outlined in Annex A. Reference is made also to the SCIA Design Forms WebHelp for a complete list of ID's.

Cross-Section Properties	ESA ID
Area	CS.Chars.A
Shear Area in principal y-direction	CS.Chars.Ay
Shear Area in principal z-direction	CS.Chars.Az
Circumference per unit length	CS.Chars.AL
Drying Surface per unit length	CS.Chars.AD
Centroid coordinate in Y-direction of Input axis system	CS.Chars.cYUCS
Centroid coordinate in Z-direction of Input axis system	CS.Chars.cZUCS
Second moment of area about the YLCS axis	CS.Chars.IYLCS
Second moment of area about the ZLCS axis	CS.Chars.IZLCS
Product moment of area in the LCS system	CS.Chars.IYZLCS
Rotation Angle of the principal axis system	CS.Chars.alpha
Second moment of area about the principal y-axis	CS.Chars.Iy
Second moment of area about the principal z-axis	CS.Chars.Iz
Radius of gyration about the principal y-axis	CS.Chars.iy
Radius of gyration about the principal z-axis	CS.Chars.iz
Elastic section modulus about the principal y-axis	CS.Chars.Wely
Elastic section modulus about the principal z-axis	CS.Chars.Welz
Plastic section modulus about the principal y-axis	CS.Chars.Wply
Plastic section modulus about the principal z-axis	CS.Chars.Wplz
Plastic moment about the principal y-axis for a positive My moment	CS.Chars.Mplyp
Plastic moment about the principal y-axis for a negative My moment	CS.Chars.Mplyn
Plastic moment about the principal z-axis for a positive Mz moment	CS.Chars.Mplzp
Plastic moment about the principal z-axis for a negative Mz moment	CS.Chars.Mplzn
Shear center coordinate in principal y-axis measured from the centroid	CS.Chars.dy
Shear center coordinate in principal z-axis measured from the centroid	CS.Chars.dz
Torsional constant	CS.Chars.It
Warping constant	CS.Chars.Iw
Mono-symmetry constant about the principal y-axis	CS.Chars.beta_y
Mono-symmetry constant about the principal z-axis	CS.Chars.beta_z

Cross-Section Dimensions	ESA ID
Formcode 1: I-section	
Height	CS.Geometry.h
Flange width	CS.Geometry.b
Flange thickness	CS.Geometry.t
Web thickness	CS.Geometry.s
Radius at flange root	CS.Geometry.r
Radius at flange toe	CS.Geometry.r1
Flange slope	CS.Geometry.a
Internal bolt distance	CS.Geometry.W
Unit warping at flange toe	CS.Geometry.wm
Formcode 2003: Rectangle	
Rectangle width	CS.Geometry.B
Rectangle height	CS.Geometry.H

Internal Forces	ESA ID
N	InternalForces.N
Vy	InternalForces.Vy
Vz	InternalForces.Vz
Mx	InternalForces.Mx
My	InternalForces.My
Mz	InternalForces.Mz

Material	ESA ID
Base Yield Strength fy	Material.EC.fy
Base Ultimate Strength fu	Material.EC.fu
mat. char. E	Material.EC.E
mat. char. G	Material.EC.G
mat. char.- Concrete - E	Material.EC.Ecm
mat. char.- Concrete - G	Material.EC.G
mat. char.- Concrete - fck	Material.EC.fck
mat. char.- Concrete - fctm	Material.EC.fctm
mat. char.- Concrete - fctk_05	Material.EC.fctk_05
mat. char.- Concrete - eps_cu2	Material.EC.eps_cu2
mat. char.- Concrete - epc_cu3	Material.EC.eps_cu3
mat. char.- Concrete - eps_c2	Material.EC.eps_c2
mat. char.- Concrete - eps_c3	Material.EC.eps_c3

The above ID's are given for Eurocode EN which uses the code identification "EC". For other codes similar tags are used, for example NBR, IBC, DIN, BS, ... which all correspond to the National Codes within SCIA Engineer.

Buckling Data	ESA ID
H	Section.BucklingData.H
Ly	Section.BucklingData.Ly
Lz	Section.BucklingData.Lz
beta_y	Section.BucklingData.beta_y
beta_z	Section.BucklingData.beta_z
braced_y	Section.BucklingData.braced_y
braced_z	Section.BucklingData.braced_z
my	Section.BucklingData.my
mz	Section.BucklingData.mz

Shortcuts

In order to facilitate the use of the DataCache the following shortcuts are foreseen:

Beam

The shortcut **Beam** refers to the same target node as **Member**.

CS

The shortcut **CS** automatically refers to either **Section.CrossSection** or **Member.CrossSection**.

For prismatic members, the cross-section is defined on the level of the member. It only has to be exported to the DataCache once since it's the same for all sections. For a non-prismatic member the cross-section is exported for each section since it differs for each section.

Using the **CS** shortcut thus automatically takes care of this so the Designer of the Form does not have to worry about different locations of the Cross-section data.

In the above tables the use of the **CS** shortcut can be seen, for example the cross-section area is referred to using **CS.Chars.A**.

For a prismatic beam this could have been written as follows using the full syntax:

Beam.CrossSection.Chars.A

Note the use of the "Beam" shortcut here.

Material

The shortcut **Material** automatically refers to **CS.Material**.

The material is linked to a cross-section which in turn can come from a section level or a member level.

The yield strength for example can thus either be accessed through the shortcut **Material.EC.fy** or for a prismatic member through **Beam.CrossSection.Material.EC.fy**.

InternalForces

The shortcut **InternalForces** automatically links to the respective section and Combination ID.

For example when a 1D Member check is run, it automatically runs over the different sections of the member. This shortcut automatically accounts for those section references.

The same applies for Deformations and EndForces used in the BucklingData.

Annex C: Troubleshooting

This Annex provides solutions to common issues which might occur during the run of a SCIA Design Forms Check from SCIA Engineer.

Q: The Check is not visible in the tree but it is in the Check Manager

A:

When the check is visible in the Check Manager but not visible in the service tree it can have two possible causes:

1. Checks in SCIA Engineer are shown only in case the project is calculated, so make sure the calculation is done.

2. Within the Check Header the codes for which this check applies have been set. If the actual project is using a different code than those listed in the Check Header the check will not be displayed.

Q: When pressing Refresh the check is not executed, nothing happens

A:

In case nothing happens when executing the Check this can have the following possible causes:

1. The members in the current selection have a different material than the material defined in the Check Header.

2. In case the material is correct, something might be incorrect in the data transfer. Use the OCFILES switch as indicated in Annex A to see the contents of the DataCache and evaluate what happens in the Check using the .trace file.

In addition the OCLOG switch can be used to evaluate the data transfer through TCP and to see if any error messages are returned by the Calcexe process.

Q: Member data doesn't seem to be accounted for

A:

Use the OCFILES switch as indicated in Annex A to see the contents of the DataCache. The MemberData.emd file in each Beam.x folder shows the contents of the member data used for that beam. Evaluate what happens in the Check using the .trace file, especially the data read from the IO structure.

Q: The Firewall asks to give permission to DesignForms_CalcExe.exe

A:

As specified in Annex A the communication between SCIA Engineer and the DesignForms_CalcExe.exe process is done through TCP/IP.

For this communication, SCIA Engineer searches for a free port in the range 8000-9000.

In case the Firewall asks for permission for this process make sure to allow it since else the check will not run.

Q: The Member data dialog shows a note for a missing XML file

Could not find file
'C:\Users\Public\Documents\DesignForms_0.1\CustomLibrary\Special_Library.xml'.

A:

Libraries within SCIA Design Forms are stored in .xml files. When adding a Library component to a dialog the DataFile property is used to specify which .xml file is linked to this Library component.

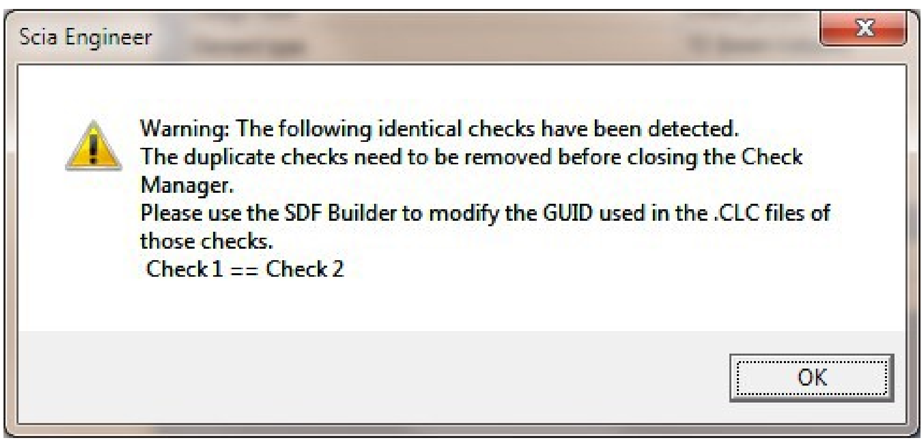
The .xml files for SCIA Design Forms are by default located in the following folder:

"C:\Users\Public\Documents\DesignForms_X.X\CustomLibrary\" where "X.X" indicates the version.

In case an .xml file which is referenced by a Library component cannot be found, the respective dialog using this component will show a note to indicate the file cannot be found. The same occurs when showing the Member data dialog from within SCIA Engineer.

This is especially important when using custom, self-defined Libraries: make sure to supply the .xml file together with your .CLC file when sending this check to another user. Each file must then be put in its respective folder.

Q: When closing the Check Manager a message is given for identical Checks

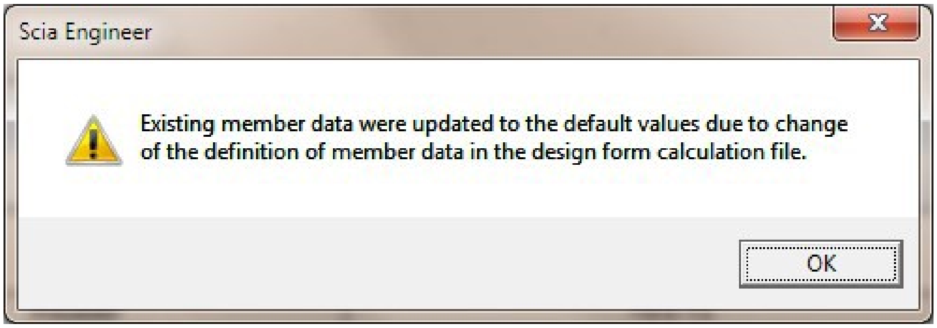


A:

As specified in Step 4 of the Examples in this manual, each .CLC file contains a unique reference number (GUID). No two Forms can have the same GUID.

When this message appears, open one of the Forms listed by the message in the SCIA Design Forms Builder and on the 'Header' tab use the button to generate a new GUID. Save the file and export it again to a .CLC file. It can then be re-imported into the Check Manager and will not give a conflict with another .CLC file anymore.

Q: When closing the Check Manager a message is given regarding updating of existing Member data

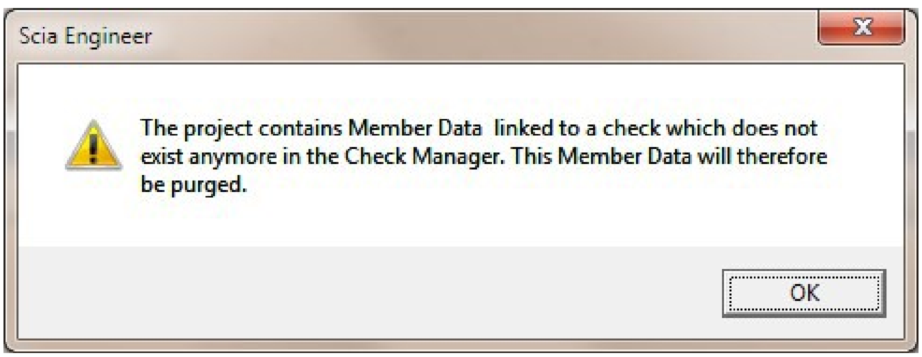


A:

In case the definition of the Member data dialog within a .CLC file is modified it means already inputted Member data are invalid (items can be added/removed from the dialog, new defaults can be set etc).

Upon Opening/Closing the Check Manager automatically verifies if already inputted Member data are still valid. If not, then these data are updated with the new Member data dialog definition and all input data is reset to the new defaults set in the .CLC file.

Q: When closing the Check Manager a message is given regarding purging of existing Member data



A:

In case a project contains Member data which is linked to a check that is no longer defined in the Check Manager this data will be purged upon Opening/Closing the Check Manager.

If this is not the intention (for example the respective Check Manager db4 was not yet updated) then close the project without saving, replace the db4 database or import the .CLC in a new project and then re-open the project in question.

Q: The output is not shown in the Document / Engineering Report

A:

In case the table with Check results is not displayed in the Document / Engineering Report, edit the respective OTX / TLX to make sure the required table components are added.

In most cases the table will contain for example the component for the Brief output and not for the Detailed output and thus the Detailed is empty. Make sure both components are properly added to the table template.