Tekla Structures 2017
Custom components

March 2017

©2017 Trimble Solutions Corporation
# Contents

1  What is a custom component................................................................. 5  
  1.1  Custom parts....................................................................................... 7  
  1.2  Custom connections........................................................................... 9  
  1.3  Custom details.................................................................................. 11  
  1.4  Custom seams.................................................................................. 13  

2  Create a custom component............................................................... 16  
  2.1  Explode a component........................................................................ 19  
  2.2  Create a nested custom component.................................................. 19  
  2.3  Create a thumbnail image of a custom component............................. 22  
  2.4  Example: Create a custom end plate component............................... 23  
  2.5  Example: Create a nested connection with stiffeners......................... 26  

3  Modify a custom component............................................................... 31  
  3.1  Protect a custom component with a password.................................. 33  

4  Add variables to a custom component.............................................. 34  
  4.1  Bind component objects to a plane................................................... 35  
    Bind objects automatically................................................................. 35  
    Bind objects manually....................................................................... 37  
    Test a binding................................................................................... 40  
    Delete a binding.............................................................................. 40  
    Example: Bind an end plate to a plane............................................. 41  
  4.2  Bind component objects using magnetic construction planes or lines... 43  
    Bind handles using a magnetic construction plane............................. 43  
    Bind handles using a magnetic construction line............................... 45  
  4.3  Add a distance between component objects.................................... 45  
  4.4  Copy object properties using parametric variables............................ 48  
  4.5  Copy properties and property references from another object............ 52  
  4.6  Create a variable formula................................................................. 53  
    Functions in variable formulas......................................................... 55  
      Arithmetic operators........................................................................ 55  
      Logical and comparison operators.................................................. 56  
      Reference functions........................................................................ 56  
      ASCII file as a reference function.................................................. 58  
      Mathematical functions................................................................... 59  
      Statistical functions........................................................................ 60  
      Data type conversion functions..................................................... 61  
      String operations............................................................................ 62  
      Trigonometric functions.................................................................. 63
What is a custom component

You can create customized connections, parts, seams, and details for your project. These are called custom components. You can use custom components in the same way as any Tekla Structures system component. You can even create intelligent custom components that automatically adjust to changes in the model.

When to use

Create a custom component if you cannot find a predefined system component that meets all your needs. Especially if you need to create a large number of complex model objects and copy them across several projects.

Benefits

Once you store a custom component in the Applications & components catalog, you can easily access it from the catalog and use it in another location in the same model. If you must modify the custom component, you only need to make the changes once. When you save the changes, they will be automatically applied to all copies of that custom component in the model. You can also import and export custom components as .uel files between models and share the custom components with your colleagues.

Custom component types

You can create four types of custom components:
<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom part (page 7)</td>
<td>Creates a group of objects that may contain connections and details.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Unlike other custom components, custom parts are <strong>not</strong> marked with a component symbol in the model. Custom parts have the same position properties as beams have.</td>
<td></td>
</tr>
<tr>
<td>Custom connection (page 9)</td>
<td>Creates connection objects and connects the secondary parts to the main part. The main part may be continuous at the connection point.</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Custom detail (page 11)</td>
<td>Creates detail objects and connects them to a single part at the location you picked.</td>
<td><img src="image" alt="Custom detail" /></td>
</tr>
<tr>
<td>Custom seam (page 13)</td>
<td>Creates seam objects and connects the parts along a line that you create by picking with two points. The parts are usually parallel.</td>
<td><img src="image" alt="Custom seam" /></td>
</tr>
</tbody>
</table>

### 1.1 Custom parts

Custom parts may consist of a single part or a group of parts, and they often have a complex composition. The following images show some examples of custom parts:
<table>
<thead>
<tr>
<th>Steel</th>
<th>Company standard bracing plates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Castellated beam and cell beam</td>
<td></td>
</tr>
<tr>
<td>Built-up beams/ columns</td>
<td></td>
</tr>
<tr>
<td>Built-up beams</td>
<td></td>
</tr>
<tr>
<td>Standard glazing fixings</td>
<td></td>
</tr>
</tbody>
</table>
1.2 Custom connections

Custom connections can be used to connect a main part to up to 30 secondary parts. The connection is made between the main part and the ends of the
secondary parts. The following images show some examples of custom connections:

<table>
<thead>
<tr>
<th>Steel</th>
<th>Built-up plate seat</th>
<th><img src="image1" alt="Built-up plate seat" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Shear plate</td>
<td><img src="image2" alt="Shear plate" /></td>
<td></td>
</tr>
<tr>
<td>Typical japanese post connections</td>
<td><img src="image3" alt="Typical japanese post connections" /></td>
<td></td>
</tr>
<tr>
<td>Precast concrete</td>
<td>Base detail</td>
<td><img src="image4" alt="Base detail" /></td>
</tr>
<tr>
<td></td>
<td>Double tee to L profile</td>
<td><img src="image5" alt="Double tee to L profile" /></td>
</tr>
</tbody>
</table>
1.3 Custom details

Custom details can be used to add more information to a single part, such as extra plates or cut-outs. The following images show some examples of custom details:

See also

Custom parts (page 7)
Custom details (page 11)
Custom seams (page 13)
<table>
<thead>
<tr>
<th>What is a custom component</th>
<th>Custom details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cast base</td>
<td></td>
</tr>
<tr>
<td>Timber base</td>
<td></td>
</tr>
<tr>
<td>Out rigger (stiffeners) and out rigger plate (stiffeners)</td>
<td></td>
</tr>
<tr>
<td>Precast concrete</td>
<td>Door and window</td>
</tr>
<tr>
<td>Column patterns</td>
<td></td>
</tr>
</tbody>
</table>
1.4 Custom seams

Custom seams can be used to connect a main part to up to 30 secondary parts. They can also be used on one main part only. The seam is made along
the length of the part. The following images show some examples of custom seams:

<table>
<thead>
<tr>
<th>Steel</th>
<th>Steel stair step</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Steel" /></td>
<td><img src="image2.png" alt="Steel stair step" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Turnbuckles</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3.png" alt="Turnbuckles" /></td>
</tr>
</tbody>
</table>

What is a custom component

Custom seams
<table>
<thead>
<tr>
<th>Handrail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precast concrete</td>
</tr>
<tr>
<td>Panel to panel grout tube connection</td>
</tr>
</tbody>
</table>

**See also**
- Custom parts (page 7)
- Custom connections (page 9)
- Custom details (page 11)
2 Create a custom component

You can create customized components that have all the details you need. Start by creating a simple custom component which you can modify later. In the following example, we will create a simple custom connection.

1. In the model, create a sample component that contains all the necessary component objects, such as parts, cuts, fittings, and bolts.
   To do this quickly, **explode and modify (page 19)** a similar existing component.

2. Click the **Applications & components** button in the side pane to open the **Applications & components** catalog.

3. Click the **Access advanced features** button and select **Define custom component**.
   The **Custom Component Wizard** dialog box opens.
4. In the **Type** list, select the component type (page 5): connection, detail, seam, or part.

5. In the **Name** box, enter a unique name for the component.

6. Modify the other properties (page 124) on the **Type/Notes** tab, **Position** tab, and **Advanced** tab, and then click **Next**.

7. In the model, select the objects you want to include in the custom component.

You can use area selection to select multiple objects simultaneously. The main and secondary parts and the grids are ignored when you select objects for the custom component.

**NOTE** If you cannot select desired objects in the model, check the selection switches and the selection filter settings.

8. Click **Next**.

9. Select the main part for the component.
10. Click **Next**.

11. Select the secondary parts for the component.

To select multiple secondary parts, hold down **Shift** when you select them. The maximum number of secondary parts in a custom component is 30.

**NOTE** Pay attention to the order in which you select secondary parts. Tekla Structures will use the same picking order when you use the custom component in a model.

12. Define any other properties required for this custom component, such as detail or seam position. The properties depend on the component type that you selected in step 4.
13. If you want to adjust any of the settings at this stage, click **Back** to return to the previous page of the **Custom Component Wizard**.

14. When you are happy with the settings, click **Finish** to create the custom component.
   
The custom component is added in the model and in the **Applications & components** catalog.

15. If you want to change these settings afterwards:
   
   a. On the **custom component editor toolbar (page 31)**, click the **Modify custom component settings** button.
   
   b. Change the settings.
   
   c. Click **OK**.

**See also**

- Create a nested custom component (page 19)
- Example: Create a custom end plate component (page 23)
- Tips for creating custom components (page 141)

### 2.1 Explode a component

Exploding means that you ungroup the objects of an existing component. This can be useful when you want to create custom components more quickly.

Once the objects are detached, you can modify them to suit your needs and then create new custom components using these objects.

1. Select the component you want to explode.
2. Right-click and select **Explode Component**.

   Tekla Structures separates the component objects. You can modify the objects and use them to create new custom components (page 16).

### 2.2 Create a nested custom component

You can create more complex custom components by joining two or more components together as a nested component. The original components become sub-components in the nested component.

1. In the model, create the components and other model objects that you want to include in the nested component.
2. Click the **Applications & components** button in the side pane to open the **Applications & components** catalog.

3. Click the **Access advanced features** button and select **Define custom component**. The **Custom Component Wizard** dialog box opens.

4. In the **Type** list, select the type of the nested custom component.

5. In the **Name** box, enter a unique name for the nested component.

6. Modify the other properties on the **Type/Notes** tab, **Position** tab, and **Advanced** tab, and then click **Next**.

7. Select the components and any other objects you want to include in the nested component, and then click **Next**.

8. Follow the instructions in the **Custom Component Wizard** to continue. You will be asked to select the main and secondary parts for the nested component. Depending on the component type that you selected in step 3, you may also have to define other properties, such as detail or seam position.

9. When you are happy with the settings, click **Finish** to create the nested component.
The component is added in the model and in the **Applications & components** catalog. The sub-components are shown in the (page 31), together with the other component objects:

10. If you want to change the settings afterwards:
   a. **In the custom component editor (page 31), click the** **Modify custom component settings** button.
   b. Change the settings.
   c. Click **OK**.

**WARNING** If you use a component of the type plug-in as a sub-component of a nested component, and change the sub-component’s properties in the custom component editor, note that those changes may be lost when you save the nested component and use it in a model.

To prevent losing any properties, link a variable to each plug-in property that you want to keep. You can also use component attribute files to do this. For more information, see Examples of parametric variables and variable formulas (page 68).
2.3 Create a thumbnail image of a custom component

Create a thumbnail image for each custom component to make it easier to find a suitable component when modeling.

1. In the model, select the custom component.
2. Right-click and select Edit Custom Component.
3. Adjust the view and hide unnecessary objects to have a clear view of the custom component.
4. Take a screenshot of the custom component.
   a. On the View tab, click Screenshot --> Screenshot to open the Screenshot dialog box.
   b. Click Pick view and select the view you want to take a screenshot of.
   c. Click Options to open the Screenshot Options dialog box.
   d. Select Print to file.
   e. Select White background and click OK.
   f. In the Screenshot dialog box, click Capture.
   g. Click Close to close the dialog box.
   h. On the File menu, click Open the model folder.
   i. Browse to the screenshots folder under the model folder.
   j. Open the screenshot file in a graphics editor.
   k. Crop the image, if needed.
5. Save the thumbnail image in the screenshots folder under the model folder.
6. Click the Applications & components button in the side pane to open the Applications & components catalog.
7. Right-click the custom component and select Thumbnails.
   The Thumbnails dialog box opens.
8. Click Add thumbnail.
9. Browse to the screenshots folder under the model folder.
10. Select the thumbnail image and click Open.
11. In the Thumbnails dialog box, select the check box next to the image you want to use and clear the other check boxes.
12. Click Close.
2.4 Example: Create a custom end plate component

In this example, you will create a simple custom component based on an existing end plate component.

1. Click the **Applications & components** button in the side pane to open the **Applications & components** catalog.

2. Click the **Access advanced features** button and select **Explode component**.

3. Select the end plate component in the model.

Tekla Structures separates the objects in the component.
4. Click the **Access advanced features** button and select **Define custom component**.

5. In the **Type** list, select **Connection**.

6. In the **Name** box, enter a unique name for the custom component.

![Custom Component Wizard](image)

7. Click **Next**.

8. Select the objects you want to use in the custom component, and then click **Next**.

Example: Create a custom end plate component
You can use area selection (left to right) to select the objects. Tekla Structures ignores the main part and secondary parts and the grids when you are selecting objects to include in the custom component.

9. Select the column as the main part, and then click **Next**.

   The main part supports the secondary part.

10. Select the beam as the secondary part.

    The secondary part is supported by the main part.

    **NOTE** When you select multiple secondary parts, pay attention to the order of selection. The custom component will use the same selection order when you add the component in a model. The maximum number of secondary parts in a custom component is 30.

11. Click **Finish**.
Tekla Structures displays a component symbol for the new component.

You have now defined a simple custom component, which you can use in locations similar to where it was originally created. This component is not intelligent and Tekla Structures does not adjust dimensions to suit any changes in the model. To make the custom component intelligent, you need to modify (page 31) it in the custom component editor.

2.5 Example: Create a nested connection with stiffeners

In this example, you will create a nested custom connection that consists of an end plate, a bolt group, welds, and two Stiffeners (1003) components. The stiffeners are optional, which means that you can choose whether or not to create them when using the component in a model.

1. Add an End plate (144) component.

Create a custom component 26 Example: Create a nested connection with stiffeners
2. **Explode (page 19)** the end plate component.

3. Add two **Stiffeners (1003)** components.

4. Create a nested custom component that contains the stiffeners and the end plate objects.

   a. Click the **Applications & components** button in the side pane to open the **Applications & components** catalog.

   b. Click the **Access advanced features** button and select **Define custom component**.

   c. In the **Type** list, select **Connection**.

   d. In the **Name** box, enter **End plate with stiffeners**.

   e. Click **Next**.

   f. Make an area selection (from right to left) to include the following objects in the nested component: the column, the beam, the stiffener
components, and all the end plate objects.

g. Click **Next**.

h. Choose the column as the main part of the nested component, and then click **Next**.

i. Choose the beam as the secondary part of the nested component, and then click **Finish**. Tekla Structures creates the nested component.

5. Select the nested component you just created.
6. Right-click and select **Edit Custom Component**.

7. In the custom component editor, click the **Display variables** button. The **Variables** dialog box opens.
8. Create the following parametric variables:
   a. Click Add to create a new parametric variable P1.
   b. In the Value type list, select Yes/No.
   c. In the Label in dialog box box, enter Create Stiffener 1.
   d. Click Add to create a new parametric variable P2.
   e. In the Value type list, select Yes/No.
   f. In the Label in dialog box box, enter Create Stiffener 2.

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula</th>
<th>Value</th>
<th>Value type</th>
<th>Variable type</th>
<th>Visibility</th>
<th>Label in dialog box</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>0</td>
<td>0</td>
<td>Yes/No</td>
<td>Parameter</td>
<td>Show</td>
<td>Create Stiffener 1</td>
</tr>
<tr>
<td>P2</td>
<td>0</td>
<td>0</td>
<td>Yes/No</td>
<td>Parameter</td>
<td>Show</td>
<td>Create Stiffener 2</td>
</tr>
</tbody>
</table>

9. Link the variables to the Creation property of the two stiffeners:
   a. In the Custom component browser, browse for the uppermost Connection.
   b. Right-click Creation and select Add Equation.
   c. Enter P1 after the equal sign, and then press Enter.
   d. Browse for the second Connection.
   e. Right-click Creation and select Add Equation.
   f. Enter P2 after the equal sign, and then press Enter.

10. Save and close (page 88) the nested component.
You now have the following options in the nested component's dialog box:

- Create Stiffener 1
- Create Stiffener 2

Example: Create a nested connection with stiffeners
3 Modify a custom component

Use the custom component editor to fine-tune existing custom components. When you modify a custom component, Tekla Structures updates all instances of that component throughout the model with the changes you have made.

1. In the model, select the custom component by clicking the green component symbol.

   **NOTE** Custom parts do not have a component symbol in the model. To select custom parts, ensure that the `Select components` selection switch is active.

2. Right-click and select **Edit Custom Component**.
   The custom component editor opens. It consists of the following parts:
   - The **Custom component browser**
• The **Custom component editor** toolbar

![Custom component editor toolbar](image)

• Four different **views** of the custom component

![Custom component views](image)

3. **Modify the custom component.** You can, for example:
   • **Add or remove component objects**  
     For example, add extra bolts or stiffeners to the component. Only component objects, not the main or secondary parts, can be modified in the custom component editor.
   • **Bind component objects to a plane** (page 35)
   • **Add a distance between component objects** (page 45)
   • **Set object properties using parametric variables** (page 48)

4. **Save the custom component** (page 88). Click **Yes** when prompted to replace all occurrences of the custom component in the model. All instances of the custom component are now updated with the changes you made.

**See also**

*Protect a custom component with a password* (page 32)
3.1 Protect a custom component with a password

You can set a password to prevent others from modifying a custom component. Password-protected custom components can still be added to models as usual.

1. In the model, select a custom component.
2. Right-click the custom component symbol and select Edit Custom Component.
3. In the custom component editor, click the Display variables button. The Variables dialog box opens.
4. Click Add to create a new variable.
5. In the Name box, enter Password.
6. In the Formula box, enter the desired password.
7. Save the custom component. (page 88)

The next time someone tries to edit this custom component, they will be asked for the password.
Variables are the properties of a custom component. You can create variables in the custom component editor and use them to adapt custom components to changes in the model. Some of the variables appear in the custom component's dialog box, while others are hidden and are only used in calculations.

Variable types
There are two types of variables:

- **Distance variable:** The distance between two planes, or between a point and a plane. A distance variable binds parts together, or works as a reference distance.

- **Parametric variable:** Controls all the other properties in a custom component, such as name, material grade, and bolt size. Parametric variables are also used in calculations.

Distance variables
Use distance variables to bind custom component objects to a plane, so that the component objects stay at a fixed distance even if the surrounding objects change. You can create distance variables manually or automatically.

You can bind the following objects to a plane:

- construction planes
- reference points of parts (only custom component objects)
- reference points of bolt groups
- chamfers
- part and polygon cut handles
- line cuts
- reference points of reinforcing bars
• reference points of reinforcement meshes and strands
• fittings

You can decide which distance variables are shown in the custom component's dialog box. Show the variables if you want to edit their values in the dialog box. Hide the variables if you only use them to bind objects to a plane.

**Parametric variables**

Use parametric variables to set properties for any object the custom component creates (page 48). After creating the variable, you will be able to change the value directly in the custom component's dialog box.

You can also create formulas to calculate values. For example, you can calculate the position of a stiffener relative to the beam length.

You can decide which parametric variables are shown in the custom component's dialog box. Show the variables if you want to edit their values in the dialog box. Hide the variables if you only use them in calculations.

### 4.1 Bind component objects to a plane

Use *distance variables* to bind component objects to a plane. Binding keeps the custom component at a fixed distance from the plane even if the surrounding objects change. Distance variables automatically get the prefix **D** (distance), which is shown in the **Variables** dialog box.

**Bind objects automatically**

You can bind objects automatically to the main and secondary parts of a connection or detail. The selected objects, or their handles, are bound to existing planes if the objects (or handles) are located exactly on the plane.

**NOTE** You cannot bind custom parts (page 7) automatically, because they do not have a main part.

1. In the custom component editor, click the Create distances variables automatically button.
2. Select an object that has handles.
3. Click the middle mouse button to bind the object. Tekla Structures binds the object from a maximum of three directions to the existing planes.

Tekla Structures displays a distance symbol for each binding. Select the object to see the bindings.
The corresponding distance variables are shown in the (page 136) dialog box:

![Variables dialog box](image)

## Bind objects manually
Create the bindings manually if you want to bind a custom component from specific handles only. You can bind an object to a maximum of three planes.

1. Ensure that **Direct modification** is switched off.
   
The selection of handles is easier when **Direct Modification** is off.

2. Ensure that you are using a model view that shows object faces.
   
   On the **View** tab, click **Rendering**, and use one of the following options:
   - **Parts grayscale** (Ctrl+3)
   - **Parts rendered** (Ctrl+4)

3. In a custom component view, select the custom component to see its handles.
4. Select the handle that you want to bind to a plane.

5. In the custom component editor, click the Add fixed distance button. You can also right-click and select Bind to Plane.

6. Move the mouse pointer in a custom component view to highlight the plane that you want to bind with the handles.
For example:

**NOTE** If you cannot highlight the correct plane, change the plane type (page 132) on the Custom component editor toolbar. Boundary and component planes work for most profile types, so try to use them whenever you can.

7. Click the plane to create the binding.
   Tekla Structures displays a distance symbol for the binding.

The corresponding distance variable is shown in the **Variables** dialog box:
Test a binding
Test all bindings to see that they work correctly.

1. Double-click the binding symbol in a custom component view.
   The **Distance Properties** dialog box opens.

2. In the **Value** box, enter a new value.
3. Click **Modify**.
   You should see the binding change in the model.

**TIP** Alternatively, you can test the binding in the (page 136) dialog box:
   a. Enter a new value in the **Formula** box.
   b. Press **Enter**.
      You should see the binding change in the model.

Delete a binding
Bindings cannot be modified, but you can delete the existing bindings and then create new ones to rebind the objects.

1. Select the binding in a custom component view.
2. Press **Delete**.
   
   You can also select the binding in the (page 136) dialog box and then click the **Delete** button.

**Example: Bind an end plate to a plane**

In this example, you will bind the end plate top to the upper side of the beam.

1. Ensure that **Direct modification** is switched off. The selection of end plate handles is easier when **Direct Modification** is off.
2. In a custom component view, select the end plate to see the end plate handles.

3. Select the top handle of the end plate.
4. Right-click the top handle and select **Bind to Plane**.
5. Move the pointer over the upper side of the beam flange to highlight it.
Here we are using the boundary plane type. If the part profile changes, the boundary plane is always found.

**NOTE** If you cannot highlight the desired plane, change the plane type (page 132) on the Custom component editor toolbar.

6. Click the upper side of the beam flange.
   A distance symbol appears in the custom component views.

7. Give a descriptive name for the binding you created:
   a. In the custom component editor, click the **Display variables** button
      ![](image)
      The **Variables** dialog box opens.
   b. In the **Label in dialog box** box, enter **Plate Top to Flange Top** as the name of the new binding.
4.2 Bind component objects using magnetic construction planes or lines

Instead of binding each component object handle to a plane separately, you can use magnetic construction planes and lines. The objects that are directly on a magnetic construction plane (or line) will move with the plane (or line), which means you only need to create one distance variable instead of 8, for example.

Bind handles using a magnetic construction plane

1. In the custom component editor, click Add construction plane.
2. Pick four points to define the shape of the construction plane.
   For example, create a plane that goes through all the handles and chamfers of the custom component.
3. Click the middle mouse button.
   Tekla Structures creates a construction plane. For example:

5. Enter a name for the plane.
6. Select the Magnetic check box.
7. Click Modify.
Now when you move the construction plane, all handles that are on the plane are moved as well:

8. Bind the construction plane to a part face:
   a. Select the construction plane, right-click and select **Bind to plane**.
   b. Select a suitable part face.
      For example, the inner flange of the column:

      ![](image)

      Tekla Structures displays a distance symbol for the binding. Now if you move the part face, the handles on the magnetic construction plane will follow.

      **NOTE** Only the objects whose reference points are directly on the magnetic construction plane are affected. By default, the magnetic distance is 0.2 mm. To change this setting, use the advanced option `XS_MAGNETIC_PLANE_OFFSET`.  

Add variables to a custom component | 44 | Bind component objects using magnetic construction planes or lines
Bind handles using a magnetic construction line

1. In the custom component editor, click \[ Add construction line. \]
2. Pick the starting point of the construction line.
3. Pick the end point of the construction line.
   Tekla Structures creates a construction line.
4. Double-click the line. The **Construction Line Properties** dialog box opens.
5. Enter a name for the line.
6. Select the **Magnetic** check box.
7. Click **Modify**.
   Now when you move the construction line, all handles that are on the line are moved as well.
8. Bind the construction line to a part face:
   a. Select the construction line, right-click and select **Bind to plane**.
   b. Select a suitable part face.
      Tekla Structures displays a distance symbol for the binding. Now if you move the part face, the handles on the magnetic construction line will follow.

4.3 Add a distance between component objects

Use **reference distance variables** to add a distance between two points or a point and a plane. The reference distance changes as you move the objects it refers to. You can use reference distances in calculations, for example, to determine the spacing of rungs on a ladder. Reference distance variables automatically get the prefix D (distance), which is shown in the **Variables** dialog box.

1. In a custom component view, select a handle.
This is the starting point for your measurement.

2. In the custom component editor, click the Add reference distance button.

3. Move the mouse pointer in the view to highlight a plane.
   This will be the end point for your measurement. If you cannot highlight the correct plane, change the plane type (page 132) on the Custom component editor toolbar.

4. Click the plane to select it.
Tekla Structures displays the distance.

The corresponding reference distance variable is shown in the Variables dialog box:

Note that the Add reference distance command remains active. You can click more planes if you want to measure other distances.

5. To stop measuring, press **Esc**.
6. To check that the reference distance works correctly, move the handle.
The distance changes accordingly. For example:

4.4 **Set object properties using parametric variables**

Use *parametric variables* to set basic properties (such as name, material, profile, position number, and so on) for any object the custom component creates. Parametric variables automatically get the prefix `P` (parameter), which is shown in the *Variables* dialog box.

In the following example, we will create a variable that sets all welds in a custom component to a given size. After creating the variable, we will be able to change the weld size directly in the custom component's dialog box.

1. In the custom component editor, click the **Display variables** button. The *Variables* dialog box opens.
2. Click **Add** to create a new parametric variable.
3. In the **Name** box, enter a name for the variable. You can also use the default name, such as `P1`. In our example, we will enter `Weldsize` as the name of the variable.
4. In the **Value type** list, select a suitable value type ([page 136](#)). The type determines what kind of values can be used with this variable. In our example, we will select **Length**, which is suitable for lengths and distances.
5. In the **Formula** box, enter a value or variable formula. In our example, we will leave this box empty.
6. In the **Label in dialog box** box, enter a descriptive name for the parametric variable.

This label will be shown in the custom component's dialog box. In our example, we will enter **Weld size** as the label.

7. In the **Visibility** list, define whether the variable will be visible in the custom component's dialog box.

Hide the variable if you only use it in calculations. Show the variable if you want to be able to edit the value in the custom component's dialog box. In our example, we will select **Show**.

8. Click **Close**.

In our example, we have now created a parametric variable with the following settings:

![Variables dialog box](image)

9. In the **Custom component browser**, link the variable to the desired object property.

   a. Select the property.

   In our example, we will select the **Size above line** property of the uppermost weld.

   ![Component objects](image)

   b. Right-click the property and select **Add Equation**.
c. After the equal sign, enter the name of the parametric variable. In our example, we will enter **Weldsize** here.

You can now modify the **Size above line** property by using the **Weld size** box in the custom component's dialog box.

10. Repeat step 9 for any other property of the same type, if needed.
In our example, we will repeat the procedure for the other welds as well, so that they will all be linked to the **Weld size** box in the custom component's dialog box.

11. **Save the custom component.** *(page 88)*

   The variable is now displayed in the custom component's dialog box, unless you set the visibility of the variable to **Hide** in step 7.

   ![Custom Component Dialog Box](image)

   If we change the weld size value now, the size of all welds within the custom component will change accordingly.

**See also**

*Copy properties and property references from another object* *(page 51)*
4.5 **Copy properties and property references from another object**

You can copy properties, such as names and values, from other objects and use them to determine the properties of a custom component. You can also copy property *references*. The link is dynamic, so when the property changes, the reference reflects the change. For example, you can use a beam length reference in variable formulas. Even if the length changes, the correct value is always used in calculations.

1. In the **Custom component browser**, browse for the object property you want to copy.
   
   To find the required component object more easily, select it in a custom component view. Tekla Structures highlights the selected object in the **Custom component browser**.

2. Right-click the property and select one of the following:
   
   - **Copy Name**
     
     Copies the name of the object. For example, **Material**.
   
   - **Copy Value**
     
     Copies the value the object currently has. For example, **S235JR**.
   
   - **Copy Reference**
     
     Copies the link to the property. For example,
     
     `fP(Material, "ID57720EEE-0000-000E-3134-363730393237")`.
3. Right-click where you want to insert the object property, and then select Paste.

For example, you can paste a reference to the Formula box in the dialog box to use it in a calculation.

See also
Examples of parametric variables and variable formulas (page 68)

4.6 Create a variable formula

Use variable formulas to add more intelligence to your custom components. Variable formulas always begin with the equal sign (=). At its simplest, a formula can be a simple dependency between two variables, stating that P2 equals half of P1 (P2 = P1/2), for example. To create more complex calculations, you can use functions and operators inside the formula. For example, you can
add mathematical expressions, **if** statements, references to object properties, and so on.

In the following example, we will create a formula that sets the weld size to half the thickness of the secondary part flange. When the component is used in a model, Tekla Structures will use the thickness of the secondary part flange to calculate the size of the weld.

1. In the custom component editor, click the **Display variables** button.
   The **Variables** dialog box opens.
2. Click **Add** to create a new parametric variable.
3. In the **Name** box, enter a name for the variable.
   In our example, we will enter \( w \) as the name of the variable.
4. In the **Custom component browser**, go to **Input objects --> Secondary parts --> Part --> Profile properties**.
5. Right-click **Flange thickness 1** and select **Copy reference**.
6. In the **Formula** box, type \( = \), right-click, and select **Paste**.
   Tekla Structures pastes the reference to flange thickness from the clipboard.
7. After the flange thickness formula, enter \( \times 0.5 \).
   The formula should now read:
   \( =\text{fP(Flange thickness 1,"GUID")}\times 0.5 \)
8. Set the other values as follows:
   a. In the **Value type** list, select **Length**.
b. In the **Visibility** list, select **Hide**.

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula</th>
<th>Value</th>
<th>Value type</th>
<th>Variable type</th>
<th>Visibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>=&quot;P1(range thickness 1,&quot;ID648C323F9-24EC-442E-9382-AA20EAE3B2AA&quot;)*0.5&quot;</td>
<td>4.00</td>
<td>Length</td>
<td>Parameter</td>
<td>Hide</td>
</tr>
</tbody>
</table>

9. In the **Custom component browser**, go to **Component objects --> Weld --> General properties**.

10. Right-click **Size above line**, select **Add equation** and type \( = w \).

### Functions in variable formulas

You can use functions to calculate values for parametric variables. Variable formulas always begin with the equal sign (=).

For more information, see Set object properties using parametric variables (page 48).

### Arithmetic operators

Use arithmetic operators to combine expressions that return numeric values. You can use the following arithmetic operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>addition</td>
<td>Use also to create strings of parameters.</td>
</tr>
<tr>
<td>-</td>
<td>subtraction</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>multiplication</td>
<td>Multiplication is faster than division.</td>
</tr>
<tr>
<td>/</td>
<td>division</td>
<td></td>
</tr>
</tbody>
</table>
**Logical and comparison operators**

Use logical and comparison operators inside `if` statements. You can use `if-then-else` statements to test a condition and to set the value according to the result.

For example:

```
=if (D1>200) then 20 else 10 endif
```

You can use the following operators inside `if` statements:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>both sides are equal</td>
<td></td>
</tr>
<tr>
<td>!=</td>
<td>sides are not equal</td>
<td></td>
</tr>
<tr>
<td>&lt;</td>
<td>left side is smaller</td>
<td></td>
</tr>
<tr>
<td>&lt;=</td>
<td>left side is smaller or equal</td>
<td></td>
</tr>
<tr>
<td>&gt;</td>
<td>right side is smaller</td>
<td></td>
</tr>
<tr>
<td>&gt;=</td>
<td>right side is smaller or equal</td>
<td></td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>logical AND</td>
<td><code>=if (D1==200 &amp;&amp; D2&lt;40) then 6 else 0 endif</code></td>
</tr>
<tr>
<td></td>
<td>both conditions must be true</td>
<td>If D1 is 200 and D2 smaller than 40, the result is 6, otherwise 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>only one condition must be true</td>
<td>If D1 is 200 or D2 is smaller than 40, the result is 6, otherwise 0.</td>
</tr>
</tbody>
</table>

**Reference functions**

Use reference functions to refer to the property of another object, such as the plate thickness of a secondary part. Tekla Structures refers to the object on the system level, so if the object property changes, so does the reference function value.

You can use the following reference functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>fTpl(&quot;template attribute&quot;, &quot;object GUID&quot;)</td>
<td>Returns the template attribute value of an object that has a given object GUID.</td>
<td>=fTpl(&quot;WEIGHT&quot;,&quot;ID50B8559A-0000-010B-3133-353432373038&quot;) returns the weight of an object whose GUID is</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>fP(&quot;user-defined attribute&quot;, &quot;object GUID&quot;)</td>
<td>Returns the user-defined attribute value of an object that has a given object GUID.</td>
<td>fP(&quot;comment&quot;, &quot;ID50B8559A-0000-010B-3133-353432373038&quot;) returns the user-defined attribute <strong>comment</strong> of an object whose GUID is ID50B8559A-0000-010B-3133-353432373038.</td>
</tr>
<tr>
<td>fValueOf(&quot;parameter&quot;)</td>
<td>Returns the value of the parameter.</td>
<td>If the equation is ( P2<em>P3 ), the result is ( P2</em>P3 ). With ( =fValueOf(&quot;P2&quot;) +&quot;<em>&quot;+fValueOf(&quot;P3&quot;) ), where ( P2=780 ) and ( P3=480 ), the result is ( 780</em>480 ).</td>
</tr>
<tr>
<td>fRebarCatalogValue(BarGrade, BarSize, Usage, FieldName)</td>
<td>Returns the reinforcing bar catalog value of an object. Usage can be either 2 (&quot;Tie&quot;) or 1 (&quot;Main&quot;). FieldName must be one of the following: • 0 NominalDiameter • 1 ActualDiameter • 2 Weight • 3 MinRadius • 4 Hook1Radius • 5 Hook1Angle • 6 Hook1Length • 7 HookRadius • 8 Hook2Angle</td>
<td>fRebarCatalogValue(&quot;A500HW&quot;, &quot;10&quot;, 1, 2) returns the size, usage, and weight of an object whose reinforcing bar grade is A500HW.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>• 9</td>
<td>Hook2Length</td>
<td></td>
</tr>
<tr>
<td>• 10</td>
<td>Hook3Radius</td>
<td></td>
</tr>
<tr>
<td>• 11</td>
<td>Hook3Angl</td>
<td></td>
</tr>
<tr>
<td>• 12</td>
<td>Hook3Length</td>
<td></td>
</tr>
<tr>
<td>• 13</td>
<td>Area</td>
<td></td>
</tr>
</tbody>
</table>

**ASCII file as a reference function**

You can refer to ASCII files to get data. Tekla Structures searches for the files in the following order:

1. model
2. ..\TeklaStructuresModels\<model>\CustomComponentDialogFiles\  
3. project (set with advanced option XS_PROJECT)  
4. firm (set with advanced option XS_FIRM)  
5. system (set with advanced option XS_SYSTEM)  

The format for reading files is the following:

```fVF("filename", "key_value_of_row", column_number)`

• Key value of row is a unique text value.  
• Column number is an index starting from 1.

**NOTE** Enter a space at the end of each row in the ASCII file. Otherwise the information is not read correctly.
Example

The `fVF("Overlap.dat", "MET-202Z25", 5)` function is in the Formula box in the Variables dialog box. The function gets the value 16.0 for the profile MET-202Z25, from the Overlap.dat file.

1. Key value of the row (MET-202Z25)
2. Column number (5)

Mathematical functions

Use mathematical functions to create more complex mathematical expressions. You can use the following functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fabs(parameter)</code></td>
<td>Returns the absolute value of the parameter</td>
<td><code>=fabs(D1)</code> returns 15 if D1 = -15</td>
</tr>
<tr>
<td><code>exp(power)</code></td>
<td>Returns e raised to the power e is Euler's number.</td>
<td><code>=exp(D1)</code> returns 7.39 if D1 = 2</td>
</tr>
<tr>
<td><code>ln(parameter)</code></td>
<td>Returns the natural logarithm of the parameter (base number e)</td>
<td><code>=ln(P2)</code> returns 2.71 if P2 = 15</td>
</tr>
<tr>
<td><code>log(parameter)</code></td>
<td>Returns the logarithm of the parameter (base number 10)</td>
<td><code>=log(D1)</code> returns 2 if D1=100</td>
</tr>
<tr>
<td><code>sqrt(parameter)</code></td>
<td>Returns the square root of the parameter</td>
<td><code>=sqrt(D1)</code> returns 4 if D1 = 16</td>
</tr>
<tr>
<td><code>mod(dividend, divider)</code></td>
<td>Returns the modulo of the division</td>
<td><code>=mod(D1, 5)</code> returns 1 if D1 = 16</td>
</tr>
<tr>
<td><code>pow(base number, power)</code></td>
<td>Returns the base number raised to the specified power</td>
<td><code>=pow(D1, D2)</code> returns 9 if D1 = 3 and D2 = 2</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example (P1 = 1.4, P2 = 2.3)</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------</td>
<td>-------------------------------</td>
</tr>
</tbody>
</table>
| hypot(side1, side2) | Returns the hypotenuse          | ![Diagram](image)  
 1. side1
  2. hypotenuse
  3. side2  
  
  =hypot(D1, D2) returns 5 if D1 = 3 and D2 = 4 |
| n!(parameter)    | Returns the factorial of the parameter | =n!(P2) returns 24  
  if P2 = 4  
  1*2*3*4 |
| round(parameter, accuracy) | Returns the parameter rounded off to the given accuracy | =round(P1, 0.1) returns 10.600  
  if P1 = 10.567 |
| PI               | Returns the value of pi to 31 decimal places | =PI returns 3.1415926535897932384626433832795 |

**Statistical functions**

Use statistical functions to calculate sums and averages, and to round values. You can use the following statistical functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example (P1 = 1.4, P2 = 2.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ceil()</td>
<td>Returns the smallest whole number greater than or equal to the parameter</td>
<td>=ceil(P1) returns 2</td>
</tr>
<tr>
<td>floor()</td>
<td>Returns the largest whole number less than or equal to the parameter</td>
<td>=floor(P1) returns 1</td>
</tr>
<tr>
<td>min()</td>
<td>Returns the smallest parameter</td>
<td>=min(P1, P2) returns 1.4</td>
</tr>
<tr>
<td>max()</td>
<td>Returns the largest parameter</td>
<td>=max(P1, P2) returns 2.3</td>
</tr>
<tr>
<td>sum()</td>
<td>Sum of the parameters</td>
<td>=sum(P1, P2) returns 3.7</td>
</tr>
<tr>
<td>sqsum()</td>
<td>Sum of the squared parameters: (parameter1)^2 + (parameter2)^2</td>
<td>=sqsum(P1, P2) returns 7.25</td>
</tr>
<tr>
<td>ave()</td>
<td>Average of the parameters</td>
<td>=ave(P1, P2) returns 1.85</td>
</tr>
</tbody>
</table>
### Function Description

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example (P1 = 1.4 P2 = 2.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqave()</td>
<td>Average of the squared parameters</td>
<td>=sqave(P1, P2) returns 3.625</td>
</tr>
</tbody>
</table>

#### Example: Ceil and floor statistical functions

In this example, you have the following parametric variables:

- Beam length: P1 = 3500
- Post spacing: P2 = 450

\[
P1 / P2 = 7.7778
\]

You can use the `ceil` and `floor` statistical functions to round the value and then use the rounded value as the number of beam posts:

- \( =\text{ceil}(P1/P2) \) returns 8
- \( =\text{floor}(P1/P2) \) returns 7

### Data type conversion functions

Use data type conversion functions to convert values into another data type. You can use the following data type conversion functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| int()    | Converts data to integer | Useful especially for calculating profile dimensions:  
=\text{int}(100.0132222000) returns 100, if decimals are set to 0 in the Options dialog box |
| double() | Converts data to a double | |
| string() | Converts data to string | |
| imp()    | Converts imperial units  
Use this function in calculations instead of imperial units. You cannot use imperial units directly in calculations. | For the following examples, length unit is set to mm and decimals are set to 2 in the Options dialog box.  
=\text{imp}(1,1,1,2) meaning 1 foot 1 1/2 inch returns 342.90 mm  
=\text{imp}(1,1,2) meaning 1 1/2 inches returns 38.10 mm  
=\text{imp}(1,2) meaning 1/2 inches returns 12.70 mm  
=\text{imp}(1) meaning 1 inch returns 25.40 mm  
=3’/3” is not possible, but  
=\text{imp}(36)/\text{imp}(3) is ok |
### Function Description

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>vwu(value, unit)</td>
<td>Converts the length values and angle values. The available units are:</td>
<td>=vwu(4.0,&quot;in&quot;) returns 101.60 mm, if length unit is set to mm and decimals are set to 2 in the Options dialog box</td>
</tr>
<tr>
<td></td>
<td>• &quot;ft&quot; (&quot;feet&quot;, &quot;foot&quot;)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;in&quot; (&quot;inch&quot;, &quot;inches&quot;)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;m&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;cm&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;mm&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;rad&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;deg&quot;</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE** The units depend on the settings in **File menu --> Settings --> Options --> Units and decimals**.

### String operations

Use string operations to manipulate character strings. Strings must be inside quotation marks in variable formulas.

You can use the following string operations:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
<th>Example (P1 = &quot;PL100*10&quot;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>match(parameter1, parameter2)</td>
<td>Returns 1 if parameters are equal and 0 if different. You can also use wildcards *, ?, and [ ] with the match function.</td>
<td>=match(P1, &quot;PL100*10&quot;) returns 1</td>
</tr>
<tr>
<td>length(parameter)</td>
<td>Returns the number of characters in the parameter.</td>
<td>=length(P1) returns 8</td>
</tr>
<tr>
<td>find(parameter, string)</td>
<td>Returns the order number (starting at zero) of the specified string and -1 if the specified string is not</td>
<td>=find(P1, &quot;*&quot;) returns 5</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
<td>Example (P1 = &quot;PL100*10&quot;)</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>getat(parametr, n)</td>
<td>Returns the n:th (starting at zero) character from the parameter.</td>
<td>=getat(P1, 1) returns &quot;L&quot;</td>
</tr>
<tr>
<td>setat(parametr, n, character)</td>
<td>Sets the n:th (starting at zero) character to the specified character in the parameter.</td>
<td>=setat(P1, 0, &quot;B&quot;) returns &quot;BL100*10&quot;</td>
</tr>
<tr>
<td>mid(string, n, x)</td>
<td>Returns x characters from the string starting from n:th (starting at zero) character. If you leave out the last argument (x), returns the last part of the string.</td>
<td>=mid(P1,2,3) returns &quot;100&quot;</td>
</tr>
<tr>
<td>reverse(string)</td>
<td>Reverses the given string.</td>
<td>=reverse(P1) returns &quot;01*001LP&quot;</td>
</tr>
</tbody>
</table>

**Example 1**

To define profile size PL100*10 with two variables P2 = 100 and P3 = 10, enter the formula as follows:

="PL"+P2+"*"+P3

**Example 2**

Tekla Structures handles bolt spacings as strings. To define bolt spacing, set **Value type** to **Distance list** and enter the formula as follows:

=P1+" "+P2

This results in 100 200, if P1 = 100 (**length**) and P2 = 200 (**length**).

**Trigonometric functions**

Use trigonometric functions to calculate angles. You can use the following trigonometric functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>sin()</td>
<td>Returns the sine value</td>
<td>=sin(d45) returns 0.71</td>
</tr>
<tr>
<td>cos()</td>
<td>Returns the cosine value</td>
<td>=cos(d45) returns 0.71</td>
</tr>
<tr>
<td>tan()</td>
<td>Returns the tangent value</td>
<td>=tan(d45) returns 1.00</td>
</tr>
<tr>
<td>asin()</td>
<td>Inverse function of sin(), return value in radians</td>
<td>=asin(1) returns 1.571 rad</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>acos()</td>
<td>Inverse function of cos(), return value in radians</td>
<td>=acos(1) returns 0 rad</td>
</tr>
<tr>
<td>atan()</td>
<td>Inverse function of tan(), return value in radians</td>
<td>=atan(1) returns 0.785 rad</td>
</tr>
<tr>
<td>sinh()</td>
<td>Return value of hyperbolical sine</td>
<td>=sinh(d45) returns 0.87</td>
</tr>
<tr>
<td>cosh()</td>
<td>Return value of hyperbolical cosine</td>
<td>=cosh(d45) returns 1.32</td>
</tr>
<tr>
<td>tanh()</td>
<td>Return value of hyperbolical tangent</td>
<td>=tanh(d45) returns 0.66</td>
</tr>
<tr>
<td>atan2()</td>
<td>Return value of angle whose tangent is the quotient of two numbers. Return value in radians</td>
<td>=atan2(1,3) returns 0.32</td>
</tr>
</tbody>
</table>

**NOTE** When you use trigonometric functions in variable formulas, you need to include a prefix to define the unit. If you do not include a prefix, Tekla Structures uses radians as the default unit.

- d is degree. For example, \( \sin(d180) \)
- r is radians (default). For example, \( \sin(r3.14) \) or \( \sin(3.14) \)

**Market size function**

Use the market size function in a custom component to select a suitable plate dimension (usually plate thickness) from the available market sizes. For example, a plate's thickness should match the web of a beam.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>fMarketSize(material, thickness, extrastep)</td>
<td>Returns the next available market size for the material from the marketsize.dat file, based on the thickness you specify. The file must be in the .. \environments \your_environment \profile folder or the system folder. For extrastep enter a number to define the increment to the next size (default is 0).</td>
<td>=fMarketSize(&quot;S235JR &quot;, 10, 0)</td>
</tr>
</tbody>
</table>
Example

In this example, you have the following data in `marketsize.dat`:

- **S235JR**: 6, 9, 12, 16, 19, 22
- **SS400**: 1.6, 2.3, 3.2, 4.5, 6, 9, 12, 16, 19, 22, 25, 28, 32, 38
- **DEFAULT**: 6, 9, 12, 16, 19, 22, 25, 28, 32, 38

The first item in a row is a material grade followed by available plate thicknesses in millimeters. The DEFAULT line lists the thicknesses available in all other material grades.

With the above data, the function `=fMarketSize("S235JR",10,0)` would return 12, and `=fMarketSize("S235JR",10,1)` would return 16 (one size up).

Framing condition functions

Use the framing condition functions to return the skew, slope, and cant angle of the secondary beam relative to the main part (column or beam). You can use the following framing condition functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fAD(&quot;skew&quot;, GUID)</code></td>
<td>Returns the skewed angle of the secondary part whose GUID is given.</td>
<td><code>=fAD(&quot;skew&quot;,&quot;ID50B8559A-0000-010B-3133-353432373038&quot;)</code> returns 45. ID50B8559A-0000-010B-3133-353432373038 is the GUID of the secondary part, which is at a 45 degree angle to the main part.</td>
</tr>
<tr>
<td><code>fAD(&quot;slope&quot;, GUID)</code></td>
<td>Returns the sloped angle of the secondary part whose GUID is given.</td>
<td><code>=fAD(&quot;slope&quot;,&quot;ID50B8559A-0000-010B-3133-353432373038&quot;)</code></td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>fAD(&quot;cant&quot;, GUID)</td>
<td>Returns the cant angle of rotated secondary part whose GUID is given.</td>
<td>=fAD(&quot;cant&quot;,&quot;ID50B8559A-0000-010B-3133-353432373038&quot;)</td>
</tr>
</tbody>
</table>

**NOTE**

- These functions do not return positive and negative slope and skew values. It is not possible to determine up or down slope and left or right skew with these functions.
- The maximum skew angle to return is 45 degrees.
- Tekla Structures calculates the angles in 2D so that slope and skew are isolated from each other. For example, the skew angle is not taken into consideration when calculating the slope angle, which means that the slope angle value stays the same regardless of the secondary part's rotation around the primary part.

To find out the true 3D slope with the skew included, you can use the following mathematical formula:

\[
\text{TRUE\_SLOPE} = \tan^{-1}(\tan(\text{SLOPE}) \times \cos(\text{SKEW}))
\]

**Example 1**

The slope and skew are relative to a beam framing into a column.

1. Column
2. Beam

Add variables to a custom component 66 Create a variable formula
3. **Slope**  
4. **Skew**  

**Example 2**  
With two beams, the **slope** is actually the horizontal skew of the beam framing into the other beam, and the vertical slope of the beam relative to the main is actually the **skew** angle.

**Side view**  
**Top view**

1. **Skew**  
2. **Slope**

**How to avoid cyclic dependencies in formulas**  
Be careful not to create cyclic dependencies between variables, or else the custom component will not work correctly. A cyclic dependency chain contains formulas that make a variable eventually dependent on itself.

In the following example, variable P1 becomes dependent on itself, through variables P2 and P3:

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>=P2</td>
</tr>
<tr>
<td>P2</td>
<td>=P3/4</td>
</tr>
<tr>
<td>P3</td>
<td>=P1*2</td>
</tr>
</tbody>
</table>

Cyclic dependencies may also occur when binding handles to other objects or when using magnetic construction planes. When you create new formulas, bindings, or magnetic construction planes, Tekla Structures checks if they create cyclic dependency chains in a custom component. If that happens, a warning message "Caution. Operation created cycle." is displayed.
4.7 Examples of parametric variables and variable formulas

Here you will find some examples that demonstrate how to use parametric variables and variable formulas to create intelligent custom components that adapt to changes in the model.

The examples are independent from each other.

• **Example: Set the end plate material (page 69)**
  In this example, you will link a parametric variable to the end plate material of a component object.

• **Example: Create new component objects (page 70)**
  In this example, you will create a parametric variable that adds bolts to the custom component.

• **Example: Replace sub-components (page 71)**
  In this example, you will create a parametric variable that replaces sub-components with other sub-components.

• **Example: Modify a sub-component by using a component attribute file (page 73)**
  In this example, you will create a parametric variable that modifies a sub-component on the basis of a component attribute file.

• **Example: Define the stiffener position using construction planes (page 74)**
  In this example, you will use construction planes for determining the position of the stiffeners. You will position the stiffeners so that they divide the beam into three equally long sections.
• Example: Determine the bolt size and bolt standard (page 77)
In this example, you will create two parametric variables that determine the bolt size and bolt standard.

• Example: Calculate the bolt group distance (page 78)
In this example, you will create a variable formula that calculates the bolt group distance from the beam flange.

• Example: Calculate the number of bolt rows (page 80)
In this example, you will create a variable formula that calculates the number of bolt rows based on the beam height. You will use if statements in the calculations.

• Example: Link variables to user-defined attributes (page 82)
In this example, you will link parametric variables to the user-defined attributes of panels. You can then use the user-defined attributes in view filters to show or hide the panels.

• Example: Calculate the number of handrail posts using a template attribute (page 84)
In this example, you will create a variable formula that calculates the number of handrail posts based on the length template attribute of the beam. The handrail posts were created at both ends of the beam and one of them was copied with the Array of objects (29) component.

• Example: Link an Excel spreadsheet to a custom component (page 86)
In this example, you will link a parametric variable to an Excel spreadsheet. For example, you can use Excel spreadsheets to check connections.

Example: Set the end plate material
In this example, you will link a parametric variable to the end plate material of a component object.

1. In the custom component editor, click the Display variables button.
The Variables dialog box opens.

2. Click the Add button.
A new parametric variable appears.

3. In the Value type list, change the variable's value type to Material.

4. In the Label in dialog box box, enter End Plate Material.
5. In the **Custom component browser**, browse for the end plate material.

6. Right-click **Material** and select **Add Equation**.
7. Enter $P1$ after the equal sign, and then press **Enter**.
8. Save the custom component.
9. Close the custom component editor.

You can now change the end plate material in the custom component's dialog box.

---

**Example: Create new component objects**

In this example, you will create a parametric variable that adds bolts to the custom component.

1. In the custom component editor, click the **Display variables** button.
   The **Variables** dialog box opens.
2. Click **Add** to create a new parametric variable.
3. Modify the variable as follows:
   a. In the Value type list, select Yes/No.
   b. In the Label in dialog box box, enter Create bolts.

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula</th>
<th>Value</th>
<th>Value type</th>
<th>Variable type</th>
<th>Visibility</th>
<th>Label in dialog box</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>0</td>
<td>0</td>
<td>Yes/No</td>
<td>Parameter</td>
<td>Hide</td>
<td>Create bolts</td>
</tr>
</tbody>
</table>

4. Select the bolt group in a custom component view to highlight it in the **Custom component browser**.
5. In the **Custom component browser**, browse for Bolt.
6. Right-click Creation and select **Add Equation**.
7. Enter $P_1$ after the equal sign, and then press **Enter**.

8. Save the custom component.
9. Close the custom component editor.

You now have the following option in the custom component's dialog box:

```
<table>
<thead>
<tr>
<th>Parameters 1</th>
<th>General</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create bolts</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

### Example: Replace sub-components
In this example, you will create a parametric variable that replaces sub-components with other sub-components.
1. In the custom component editor, click the **Display variables** button. The **Variables** dialog box opens.

2. Click **Add** to create a new parametric variable.

3. Modify the variable as follows:
   a. In the **Value type** list, select **Component name**.
      Tekla Structures automatically adds the suffix `_name` in the variable name. Do not delete the suffix.
   b. In the **Formula** box, enter the name of the sub-component.
   c. In the **Label in dialog box** box, enter **Cast-in plate**.

4. Link the variable to the **Name** property of both sub-components:
   a. In the **Custom component browser**, browse for the **Name** attribute of the first sub-component.
   b. Right-click **Name** and select **Add Equation**.
   c. Enter `P1_name` after the equals sign.
   d. Repeat steps 4b–4c for the other sub-component.
5. Save the custom component.
6. Close the custom component editor.

You can now change the sub-components by using the Cast-in-plate option in the custom component's dialog box.

Example: Modify a sub-component by using a component attribute file
In this example, you will create a parametric variable that modifies a sub-component on the basis of a component attribute file.

1. In the custom component editor, click the Display variables button. The Variables dialog box opens.
2. Click Add to create a new parametric variable.
3. In the Value type list, select Component attribute file.
   Tekla Structures automatically adds the suffix _attrfile in the variable name. Do not delete the suffix.
4. In the Formula box, enter the name of the component attribute file.
5. In the Name box, ensure that the variable has the same prefix as the variable that is linked to the component name.
   In this example, the prefix is P1.

   **NOTE** The component name and the component attribute file variables must always have the same prefix, otherwise they do not work.

6. In the Label in dialog box box, enter Properties file.
7. In the **Custom component browser**, browse for the component attribute file property of the sub-component.

8. Right-click **Attribute file** and select **Add Equation**.

9. Enter `P1_attrfile` after the equal sign, and then press **Enter**.

10. Save the custom component.

11. Close the custom component editor.

You can now modify the sub-component by using the **Properties file** option in the custom component's dialog box.

### Example: Define the stiffener position using construction planes

In this example, you will use construction planes for determining the position of the stiffeners. You will position the stiffeners so that they divide the beam into three equally long sections.
1. Ensure that **Direct Modification** is switched off. The selection of handles is easier when **Direct Modification** is off.

2. In the custom component editor, click the **Display variables** button. The **Variables** dialog box opens.

3. Click **Add** to create a new parametric variable.

4. Get the GUID of the beam.
   a. On the ribbon, click **Inquire objects**.
   b. Select the beam.
   c. In the **Inquire Object** dialog box, check the GUID of the beam.

5. Modify the variable as follows:
   a. In the **Formula** box, enter 
      $$=fTpl("\text{LENGTH}",\text{ID4C8B5E24-0000-017D-3132-383432313432})$$
      
      **ID4C8B5E24-0000-017D-3132-383432313432** is the GUID of the beam.
      
      The value of the variable is now the same as the beam length. If you change the beam length, also the value changes.
   b. In the **Label in dialog box** box, enter **Beam Length**.

6. Click **Add** to create another parametric variable.

7. Modify the new variable as follows:
   a. In the **Formula** box, enter **=P1/3**.
   b. In the **Label in dialog box** box, enter **3rd Points**.

8. Create a construction plane:
   a. In the custom component editor, click the **Add construction plane** button.
   b. Pick the required points and then click the middle mouse button to create a construction plane in the center of a stiffener at one end.
9. Bind the stiffener to the construction plane:
   a. Select the stiffener.
   b. Hold down Alt and use area selection (from left to right) to select all stiffener handles.
   c. Right-click and select **Bind to plane**.
   d. Bind the stiffener handles to the construction plane.

10. Bind the construction plane to the beam end:
    a. Select the construction plane.
    b. Right-click and select **Bind to plane**.
    c. Bind the construction plane to the beam end.
11. Repeat steps 9–11 for the stiffener at the other end.
12. In the Formula box, enter \( P_2 \) for the two distance variables that bind the construction planes to the beam ends.
13. Save the custom component.
14. Close the custom component editor.

If you now change the beam length, the position of the stiffeners changes so that the stiffeners divide the beam into three equally long sections.

**Example: Determine the bolt size and bolt standard**

In this example, you will create two parametric variables that determine the bolt size and bolt standard.

1. In the custom component editor, click the **Display variables** button.
   The Variables dialog box opens.
2. Click **Add** twice to create two new parametric variables.
3. Modify the first variable as follows:
   • In the **Value type** list, select **Bolt size**.
     Tekla Structures automatically adds the suffix _diameter to the name of the variables. Do not delete the suffix.
   • In the **Label in dialog box** box, enter **Bolt Size**.
4. Modify the second variable as follows:
   a. In the **Value type** list, select **Bolt standard**.
     Tekla Structures automatically adds the suffix _screw to the name of the variable. Do not delete the suffix.
b. In the **Name** box, change the prefix of the second variable so that the prefixes for the two variables are the same.

In this example, the prefix is P1.

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula</th>
<th>Value</th>
<th>Value type</th>
<th>Variable type</th>
<th>Visibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1_diameter</td>
<td>0.00</td>
<td>0.00</td>
<td>Bolt size</td>
<td>Parameter</td>
<td>Show</td>
</tr>
<tr>
<td>P1_screwdin</td>
<td>0.00</td>
<td>0.00</td>
<td>Bolt standard</td>
<td>Parameter</td>
<td>Show</td>
</tr>
</tbody>
</table>

**NOTE** The bolt size and bolt standard variables must always have the same prefix, otherwise they do not work.

c. In the **Label in dialog box** box, enter **Bolt Standard**.

5. Link the parametric variables to the bolt group properties:

a. In the **Custom component browser**, browse for the size property of the component object.

b. Right-click **Size** and select **Add Equation**.

c. Enter **P1_diameter** after the equal sign, and then press **Enter**.

d. Right-click **Bolt standard** and select **Add Equation**.

e. Enter **P1_screwdin** after the equal sign, and then press **Enter**.

6. Save the custom component.

7. Close the custom component editor.

You can now determine the bolt size and bolt standard for the custom component in the custom component's dialog box.

**Example: Calculate the bolt group distance**

In this example, you will create a variable formula that calculates the bolt group distance from the beam flange.
1. Modify the bolt group properties as follows:
   a. In the custom component editor, double-click the bolt group. The **Bolt Properties** dialog box opens.
   b. Clear all values that are under the **Offset from** area.
   c. Click **Modify**. The bolt group moves to the same level with the start point handle of the bolt group.

2. Bind the bolt group to the beam flange:
   a. In the custom component editor, select the bolt group.
   b. Select the yellow top handle.
   c. Right-click the handle and select **Bind to plane**.
   d. Select the top flange of the beam.
A new distance variable appears in the Variables dialog box.

3. In the custom component editor, click the Display variables button. The Variables dialog box opens.

4. Click Add to create a new parametric variable.

5. Modify the variable as follows:
   a. In the Formula box, enter a distance value.
   b. In the Label in dialog box box, enter Vertical distance to bolt.

6. In the Formula box, enter \(-P1\) for the distance variable.

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula</th>
<th>Value</th>
<th>Value type</th>
<th>Variable type</th>
<th>Visibility</th>
<th>Label in dialog box</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>(+P1)</td>
<td>-75.00</td>
<td>Length</td>
<td>Distance</td>
<td>Hide</td>
<td>D1.BOLT.BEAM</td>
</tr>
<tr>
<td>P1</td>
<td>75.00</td>
<td>75.00</td>
<td>Length</td>
<td>Parameter</td>
<td>Show</td>
<td>Vertical distance to bolt</td>
</tr>
</tbody>
</table>

7. Save the custom component.

8. Close the custom component editor.

You can now determine the bolt group distance from the beam flange by changing the Vertical distance to bolt value in the custom component’s dialog box.

Example: Calculate the number of bolt rows
In this example, you will create a variable formula that calculates the number of bolt rows based on the beam height. You will use if statements in the calculations.
1. In the custom component editor, click the **Display variables** button. The **Variables** dialog box opens.

2. Click **Add** to create a new parametric variable.

3. In the **Value type** list, select **Number**.

4. In the **Custom component browser**, browse for height property of the beam.

5. Right-click **Height** and select **Copy Reference**.

6. In the **Formula** box, enter the following **if** statement for the parametric variable:

   \[
   =\text{if (fP(Height,"ID50B8559A-0000-00FD-3133-353432363133")< 301) then 2  
   \text{else (if (fP(Height,"ID50B8559A-0000-00FD-3133-353432363133")>501) then 4  
   \text{else 3 endif) endif}}
   \]

   In the formula, 
   \(\text{fP(Height,"ID50B8559A-0000-00FD-3133-353432363133")}\) is the beam height reference copied from the **Custom component browser**. The variable gets its value in the following way:

   - If the beam height is under 301 mm, the value is 2.
   - If the beam height is over 501 mm, the value is 4.
   - If the beam height is between 300 and 500 mm, the value is 3.

7. Click **Add** to create another parametric variable.

8. In the **Value type** list, select **Distance list** for the new variable.

9. In the **Formula** box, enter \(=\text{P1+"*"+100}\) for the new variable.
In the formula, 100 is the bolt spacing and the \( P_1 \) value is the number of bolt rows.

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula</th>
<th>Value</th>
<th>Value type</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_1 )</td>
<td>( = \text{if} (\text{fP(Height, }&quot;ID50B8550A-0000 \text{ }) \leq 2) )</td>
<td>2</td>
<td>Number</td>
</tr>
<tr>
<td>( P_2 )</td>
<td>( = P_1 + &quot;x&quot; + 100 )</td>
<td>2(^*)100.00</td>
<td>Distance list</td>
</tr>
</tbody>
</table>

10. In the **Custom component browser**, browse for **Bolt group distance x**.
11. Right-click **Bolt group distance x** and select **Add Equation**.
12. Enter \( P_2 \) after the equal sign, and then press **Enter**.
13. Save the custom component.
14. Close the custom component editor.

When you now change the beam height, the number of bolt rows changes as well.

**Example: Link variables to user-defined attributes**

In this example, you will link parametric variables to the user-defined attributes of panels. You can then use the user-defined attributes in view filters to show or hide the panels in the model.

1. In the custom component editor, click the **Display variables** button.
   The **Variables** dialog box opens.
2. Click **Add** to create a new parametric variable.
3. Modify the variable as follows:
   a. In the **Value type** list, select **Text**.
b. In the **Formula** box, enter Type1.

c. In the **Label in dialog box** box, enter Panel1.

4. In the **Custom component browser**, browse for the user-defined attributes of the first panel.

   You will link the P1 variable to the USER_FIELD_1 attribute. However, the attribute is not visible in the **Custom component browser**.

5. Make the user-defined attribute visible in the **Custom component browser**:

   a. Double-click the first panel.
      
      The panel properties dialog box opens.

   b. Click **User-defined attributes**.
      
      The dialog box for user-defined attributes opens.

   c. Go to the **Parameters** tab.

   d. Enter text in the **User field 1** box.

   e. Click **Modify**.

6. In the **Custom component browser**, click **Refresh**.

   USER_FIELD_1 appears under **User-defined attributes** in the **Custom component browser**.

7. Link P1 to **USER_FIELD_1**.

   a. Right-click **USER_FIELD_1** and select **Add Equation**.

   b. Enter P1 after the equal sign, and then press **Enter**.

8. Create two new parametric variables and link them to the user-defined attributes of the other two panels.

9. Save the custom component.

10. Close the custom component editor.

    You can now create a view filter in the model to hide or show panels using the **User field 1** attribute and the **Formula** values you entered for the parametric variables in the filter.
Example: Calculate the number of handrail posts using a template attribute

In this example, you will create a variable formula that calculates the number of handrail posts based on the length template attribute of the beam. The handrail posts were created at both ends of the beam and one of them was copied with the Array of objects (29) component.

1. In the custom component editor, click the Display variables button.
   The Variables dialog box opens.
2. Create three new parametric variables by clicking Add.
3. Modify the variable P1 as follows:
   • In the Formula box, enter 250.
   • In the Label in dialog box box, enter End Distance.
4. Modify the variable P2 as follows:
   • In the Formula box, enter 900.
   • In the Label in dialog box box, enter Spacing.
5. Modify the variable P3 as follows:
   • In the Value type box, select Number.
   • In the Label in dialog box box, enter Number of Posts.
6. Inquire the GUID of the beam:
   a. On the ribbon, click Inquire objects.
   b. Select the beam.
c. Check the GUID of the beam in the **Inquire Object** dialog box.

7. In the **Formula** box of the P3 variable, enter

\[ \frac{fTpl("LENGTH","ID50B8559A-0000-010B-3133-353432373038") - (P1*2)}{P2}. \]

\( fTpl("LENGTH","ID50B8559A-0000-010B-3133-353432373038") \) is the length template attribute of the beam and ID50B8559A-0000-010B-3133-353432373038 is the GUID of the beam.

The number of the posts is calculated as follows: First the end distances are subtracted from the beam length, and then the result is divided by the post spacing.

8. In the **Custom component browser**, link the variables P2 and P3 to the properties of **Array of objects (29)**.
   a. Right-click **dist_between_elem** and select **Add Equation**.
   b. Enter P2 after the equal sign, and then press Enter.
   c. Right-click **number_of_arrays** and select **Add Equation**.
   d. Enter P3 after the equal sign, and then press Enter.

9. Bind the first post to the beam end.
   a. Select the post in the custom component view.
   b. Hold down Alt and use area selection (from left to right) to select the post handles.
   c. Right-click and select **Bind to Plane**.
10. Bind the last post to the other beam end by following the instructions in step 9.

11. Modify the distance variables as follows:
   a. In the Formula box, enter \( P1 \).
   b. In the Visibility list, select Hide.

12. Save the custom component.

13. Close the custom component editor.

You can now change the spacing and the end distance of the handrail posts in the custom component dialog box. Tekla Structures calculates the number of posts based on the spacing, end distance, and length of the beam.
Example: Link an Excel spreadsheet to a custom component

In this example, you will link a parametric variable to an Excel spreadsheet. For example, you can use Excel spreadsheets to check connections.

1. Create an Excel spreadsheet.
   The name of the spreadsheet file must be `component_"component_name".xls`. For example, `component_stiffener.xls` for a custom component whose name is `stiffener`.

2. Save the Excel spreadsheet in the model folder: `..\<model>\exceldesign\`. Alternatively, you can save the spreadsheet in the folder defined with the `XS_EXTERNAL_EXCEL_DESIGN_PATH` advanced option.

3. In the custom component editor, click the Display variables button. The Variables dialog box opens.

4. Click Add to create a new parametric variable.

5. Modify the variable as follows:
   a. In the Value type list, select Yes/No.
   b. In the Name box, enter `use_externaldesign`.
   c. In the Label in dialog box box, enter Use external design.

6. Save the custom component.

7. Close the custom component editor.

The custom component dialog box now contains the Use external design option.
After modifying a custom component, save the changes.

<table>
<thead>
<tr>
<th>To</th>
<th>Do this</th>
</tr>
</thead>
</table>
| Save changes to all copies of the custom component | 1. In the custom component editor, click the **Save component** button.  
2. In the **Save confirmation** dialog box, click **Yes**.  
Tekla Structures saves the changes and applies them to all copies of the custom component in the model. |
| Save the component with a new name | 1. In the custom component editor, click the **Save with new name** button.  
2. Enter a new name for the component.                                                                                                        |
| Save and close the component       | 1. In the custom component editor, click the **Close** button.  
2. In the **Close custom component editor** message, click **Yes**.  
If you click **No**, the custom component editor closes without saving the changes.                                                            |

**See also**

*Tips for sharing custom components* (page 142)
6 Modify the dialog box of a custom component

Tekla Structures automatically creates a dialog box for each custom component you define. You can customize the dialog box by using the Custom Component Dialog Editor tool.

To open the dialog editor, select a custom component in the model, right-click, and select Edit Custom Component Dialog Box.

<table>
<thead>
<tr>
<th>To</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>View and edit object properties</td>
<td>1. Select a dialog box element. For example, a text box.</td>
</tr>
<tr>
<td></td>
<td>2. Click Modify --&gt; Properties.</td>
</tr>
<tr>
<td></td>
<td>Now you can view and edit the current properties of the dialog box element. For example, you</td>
</tr>
</tbody>
</table>
To | Do this
can check that you have the correct text box under each label in the dialog box. Alternatively, you can double-click the dialog box element. If the dialog box element will not open for viewing and editing, try double-clicking the space right underneath the check box:

**Add a dialog box element**
Click **Insert** and select a suitable element from the list. The options are:
- **Tab Page**: add a new tab
- **Label**: add a label for a text box or list
- **Parameter**: add a text box
- **Attribute**: add a list
- **Part**: add some basic part properties
- **Profile**: add some basic profile properties
- **Picture**: add an illustrative image of the custom component

**Add an image**
1. Click **Insert --> Picture** to show the contents of the **Image Folder** set in **Tools --> Options**.
2. Select an image.
   The image must be in the bitmap (.bmp) format.
3. Click **Open**.
4. Drag the image to the desired location.

**Add a tab**
1. Click **Insert --> Tab Page**.
2. Double-click the new tab.
3. Enter a new name, and then press **Enter**.
<table>
<thead>
<tr>
<th>To</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NOTE</strong> Each tab may contain up to 25 fields. If you have more than 25 fields visible, Tekla Structures automatically creates another tab.</td>
<td></td>
</tr>
<tr>
<td><strong>Show or hide the pixel grid</strong></td>
<td>Click 🌟. Tekla Structures displays a pixel grid that makes it easier to align elements in the dialog box.</td>
</tr>
<tr>
<td><strong>Move a dialog box element</strong></td>
<td>Drag the dialog box element to a new location. You can also use the keyboard shortcuts <code>Ctrl+X</code> (cut), <code>Ctrl+C</code> (copy), and <code>Ctrl+V</code> (paste). For example, to move a dialog box element to another tab: select the dialog box element, press <code>Ctrl+X</code>, go to another tab, and press <code>Ctrl+V</code>.</td>
</tr>
<tr>
<td><strong>Select multiple dialog box elements</strong></td>
<td>Hold down the <code>Ctrl</code> key and click the dialog box elements, or use area selection.</td>
</tr>
</tbody>
</table>
| **Rename a tab or text box label** | 1. Double-click the tab or text box label.  
2. Type a new name.  
3. Press **Enter**. |
| **Remove a dialog box element** | 1. Select the dialog box element you want to remove.  
2. Press **Delete**. |
| **Remove a tab** | 1. Select the tab.  
2. Right-click and select **Delete**. |
| **Add images to a list** | 1. Select the list element.  
2. Click **Modify --> Properties**.  
3. Click **Edit Values**.  
4. Click **Browse Add**.  
5. Select the image you want to use and click **Open**. |
To | Do this
---|---
6. | Repeat steps 4–5 for any other images you want to use.
7. | Click OK to save the changes.

Save the changes | Click File --> Save.

**NOTE** If you are an advanced user, you may also modify the dialog box input (.inp) files manually in a text editor. Be careful when modifying an input file, as errors may cause the dialog box to disappear.

See also
- Dialog editor settings (page 116)
- Example: Modify the dialog box of a stiffener detail (page 93)

### 6.1 Custom component input files

Each custom component has an input file that defines the contents of the custom component's dialog box.

When you create a new custom component, Tekla Structures automatically creates an input file for the component. The input file is located in the CustomComponentDialogFiles folder under the model folder. The input file has the same name as the custom component, and the file name extension is .inp.

When you modify a custom component (page 31), you will lose any changes you have made to the input file. However, when you modify the custom component, Tekla Structures automatically creates a backup copy of the input file. The backup copy has the file name extension .inp_bak, and it is located in the CustomComponentDialogFiles folder under the model folder. Tekla Structures displays a notification when the backup file is created.

See also
- Lock or unlock the custom component input file (page 92)

### 6.2 Lock or unlock the custom component input file

You can lock the custom component's input file to prevent accidental modifications. If the file is unlocked, and someone else updates the custom component in the custom component editor, all your modifications to the dialog box will be lost.
1. In the model, select the custom component whose input file (page 92) you want to lock or unlock.
2. Right-click and select **Edit Custom Component Dialog Box**.
3. In the dialog editor, click the **Lock/Unlock** button.

If someone modifies the custom component in the custom component editor when the .inp file is locked, the .inp file will not be updated. You can still modify the dialog box in the **Custom Component Dialog Editor** even if the .inp file is locked.

### 6.3 Example: Modify the dialog box of a stiffener detail

In this example, we will edit the dialog box of a custom stiffener detail to make it easier to adjust the settings later on.

When we begin, the dialog box looks like the following:

<table>
<thead>
<tr>
<th>Stiffener set back</th>
<th>10.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plates created</td>
<td>0</td>
</tr>
<tr>
<td>Left plate class</td>
<td>[4]</td>
</tr>
<tr>
<td>Right plate class</td>
<td>[5]</td>
</tr>
</tbody>
</table>

**Do this**

1. Create a custom stiffener detail with all the needed variables that control the creation of stiffener plates.
2. Add a list with images.
3. Arrange text boxes and labels.
4. Dim unavailable options.
Example: Create a custom stiffener detail with variables
In this example, we will create a stiffener detail with variables that control the shape and position of the stiffeners.

Create a basic stiffener detail
In this example, we will create a basic stiffener detail.

1. Create a beam with two stiffeners.

   ![Image of a beam with two stiffeners]

   **TIP** To create the stiffeners, you can use the **Stiffeners (1003)** component and then explode the component.

2. Click the **Applications & components** button in the side pane to open the **Applications & components** catalog.

3. Click the **Access advanced features** button and select **Define custom component**.
   The **Custom Component Wizard** dialog box opens.

4. In the **Type** list, select **Detail**.

5. In the **Name** box, type **Stiffeners**.
6. Click **Next**.
7. Select the stiffeners and the beam as the objects that form the custom component.

![Image of custom component wizard]

8. Click **Next**.
9. Select the beam as the main part.
10. Click **Next**.
11. Select the middle point of the beam as the reference point.

**TIP** Switch to the plane view to select the middle point more easily.
12. Click **Finish** to finish creating the stiffener detail.

Tekla Structures displays a component symbol for the new custom component and adds the stiffener detail to the component catalog.

---

**Create bindings to control the stiffener shape**

In this example, we will bind custom component handles to a plane to control the shape of the stiffeners.

1. Open the stiffener detail in the custom component editor.
   a. Right-click the custom component in the model.
   b. Select **Edit Custom Component**.

   The custom component editor opens showing the custom component editor toolbar, the component browser, and four views of the custom component.
2. On the **View** tab, click **Rendering --> Parts rendered**.
   Part surfaces and available planes can be selected only when they are rendered.

3. On the custom component editor toolbar, select **Outline planes** from the list.

4. In the custom component editor, select the stiffener on the right.

5. Bind the two inside handles of the stiffener to the beam web.
   a. Select the two handles next to the beam web.

   ![Stiffener handles selected next to beam web](image)

   b. Right-click and select **Bind to Plane**.
   c. Move the pointer over the face of the web to highlight it.

   ![Stiffener handles bound to beam web](image)

   d. Click the web to bind the handles.

6. Bind the two outside handles of the stiffener to the face of the top flange.
Use the same method as in step 5.

7. Bind the two bottom handles of the stiffener to the inside face of the bottom flange.
   Use the same method as in step 5.

8. Bind the two top handles of the stiffener to the inside face of the top flange.
   Use the same method as in step 5.

9. Repeat steps 4–11 for the stiffener on the left.
10. In the custom component editor, click the **Display variables** button.

The **Variables** dialog box opens.

11. Click **Add** to create a new parametric variable P1.

12. Modify the variable P1 as follows:
   a. In the **Formula** box, enter 10.
   b. In the **Label in dialog box** box, enter **Stiffener set back**.

13. In the **Formula** box, enter $=P1$ for all variables that got values during the binding of the handles.

   For example:

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula</th>
<th>Value</th>
<th>Value type</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>0.00</td>
<td>0.00</td>
<td>Length</td>
</tr>
<tr>
<td>D2</td>
<td>0.00</td>
<td>0.00</td>
<td>Length</td>
</tr>
<tr>
<td>D3</td>
<td>10.00</td>
<td>10.00</td>
<td>Length</td>
</tr>
<tr>
<td>D4</td>
<td>10.00</td>
<td>10.00</td>
<td>Length</td>
</tr>
</tbody>
</table>

The variable P1 now controls the distances of these variables.

14. In the **Visibility** list, set the variable P1 to **Show** and the other variables to **Hide**.

We have now created distance variables that control the stiffener shape.

![Example: Modify the dialog box of a stiffener detail](image)
Create bindings to control the stiffener position
In this example, we will bind custom component handles to a plane to control the position of the stiffeners.

1. Open the stiffener detail in the custom component editor.
   a. Right-click the custom component in the model.
   b. Select Edit Custom Component.
      The custom component editor opens showing the custom component editor toolbar, the component browser, and four views of the custom component.

2. On the custom component editor toolbar, select Component planes from the list.

3. Select all the handles of both stiffeners.

4. Right-click and select Bind to Plane.

5. Bind the handles to the vertical component plane.
We have now created distance variables that control the position of the stiffeners.

**Create variables to control the stiffener thickness**

In this example, we will control the stiffener thickness so that it is one and a half times the web thickness rounded up to the nearest available plate thickness. The available thickness values are 10, 12, and 16 mm.

1. Open the stiffener detail in the custom component editor.
   a. Right-click the custom component in the model.
   b. Select *Edit Custom Component*.

   The custom component editor opens showing the custom component editor toolbar, the component browser, and four views of the custom component.

2. In the custom component editor, click the *Display variables* button.
   The *Variables* dialog box opens.

3. Click *Add* to create a new parametric variable P2.

4. Modify the variable P2 as follows:
   a. In the *Formula* box, enter \(1.5*\).
   b. In the *Visibility* list, select *Hide*.
   c. In the *Label in dialog box* box, enter *Plate calculation*.

5. Select the beam in the custom component editor to highlight the beam (primary part) in the custom component browser.
6. In the **Custom component browser**, select **Web thickness** of the primary part.

7. Right-click and select **Copy Reference**.

8. Paste the reference value to **Formula** after \(1.5 \times\).

\[
P_2 = 1.5 \times \text{Web thickness} = 12.75 \text{ Length}
\]

**NOTE** A reference function refers to the property of an object, such as the web thickness of a part. If the object property changes, so does the reference function value.

9. Click **Add** to create a new parametric variable \(P_3\).

10. Modify the variable \(P_3\) as follows:
   a. In the **Value type** list, select **Number**.
   b. In the **Formula** box, enter
   \[
   =\text{if (}P_2 < 12 \text{ and } P_2 > 10\text{)} \text{ then } 12 \text{ else if (}P_2 > 12\text{)} \text{ then } 16 \text{ else } 10 \text{ endif endif}
   \]

   This means that if \(P_2\) is less than 12 and greater than 10, the thickness is 12. If \(P_2\) is greater than 12, the thickness is 16. If none of these conditions are met, the thickness is 10.

11. In the **Custom component browser**, link the variable \(P_3\) to the **Profile** property of the first contour plate.
12. Repeat step 11 for the second contour plate.

We have now created and linked all required variables that control the stiffener thickness according to the web thickness.

**Create variables to control the creation of stiffener plates**
In this example, we will create five variables to control which stiffener plates are created and what is the class of the plates.

1. Open the stiffener detail in the custom component editor.
   a. Right-click the custom component in the model.
   b. Select **Edit Custom Component**.
      
      The custom component editor opens showing the custom component editor toolbar, the component browser, and four views of the custom component.

2. In the custom component editor, click the **Display variables** button.
   
   The **Variables** dialog box opens.

3. Click **Add** to create a new parametric variable P4.

4. Modify the variable P4 as follows:
   a. In the **Formula** box, enter 2.
   b. In the **Value type** list, select **Number**.
   c. In the **Visibility** list, select **Show**.
   d. In the **Label in dialog box** box, enter Plates created.

5. Click **Add** to create a new parametric variable P5.
6. Modify the variable P5 as follows:
   a. In the Formula box, enter =if P4==0 then 0 else 1 endif.
   b. In the Value type list, select Yes/No.
   c. In the Visibility list, select Hide.
   d. In the Label in dialog box box, enter Do not create right.

7. Click Add to create a new parametric variable P6.

8. Modify the variable P6 as follows:
   a. In the Formula box, enter =if P4==1 then 0 else 1 endif.
   b. In the Value type list, select Yes/No.
   c. In the Visibility list, select Hide.
   d. In the Label in dialog box box, enter Do not create left.

9. Click Add to create a new parametric variable P7.

10. Modify the variable P7 as follows:
    a. Rename P7 as LeftC.
    b. In the Formula box, enter 4.
    c. In the Value type list, select Number.
    d. In the Visibility list, select Show.
    e. In the Label in dialog box box, enter Left plate class.

11. Click Add to create a new parametric variable P8.

12. Modify the variable P8 as follows:
    a. Rename P8 as RightC.
    b. In the Formula box, enter 5.
    c. In the Value type list, select Number.
    d. In the Visibility list, select Show.
    e. In the Label in dialog box box, enter Right plate class.

13. In the Custom component browser, link the variables P5 and RightC to the right stiffener plate.
14. Link the variables P6 and LeftC to the left stiffener plate.

Example: Add a list with images
In this example, we will add an illustrative list in the stiffener dialog box. You can do this either in the custom component dialog editor or by editing the input (.inp) file manually.

When we begin, the dialog box has the text box shown below, and the user needs to know the values (0 is left, 1 is right, and 2 is both plates) that control the creation of stiffener plates.

```
Plates created
```

We will replace the text box with a list that is easier to use:
Add a list by using the dialog editor

1. Create a custom stiffener detail (page 93) with all the needed variables that control which stiffener plates are created.
   In our example, the variable is called **Plates created**.

2. Open the stiffener dialog box for editing.
   a. In the model, select the custom stiffener detail.
   b. Right-click and select **Edit Custom Component Dialog Box**.

3. Check the name of the parametric variable that controls the plate creation.
   a. In the dialog editor, double-click the **Plates created** box.
      The **Object Properties** dialog box opens.
   b. Check the name of the parametric variable.
In our example, the name is P4.

![Object Properties](image)

1. Click **Cancel** to close the dialog box.
2. Select the **Plates created** text box and press **Delete**.
3. Click **Insert** --> **Attribute** to add a new attribute list.
4. Drag the attribute list to a suitable location, next to the **Plates created** label.
5. Select the attribute list and then click **Modify** --> **Properties** to edit its properties.
6. Enter **P4** as the **Name** of the attribute.
7. Now the attribute list is linked to the parametric variable that controls the plate creation.
8. Click **Edit Values** to add the list items.
9. In the **Edit Attribute Values** dialog box, add an image for the left plate.
   a. Click **Browse Add**.
   b. Browse for a suitable image.
   c. Click **Open**.
10. Repeat step 9 to add an image for the right plate, and then for both plates.

Modify the dialog box of a custom component 107

Example: Modify the dialog box of a stiffener detail
12. In the **Edit Attribute Values** dialog box, select the image of both plates and then click **Default** to make the attribute the default value.

13. Click **OK**.

14. Click **Apply** in the **Object Properties** dialog box, and then click **Cancel** to close the dialog box.

15. In the dialog editor, click **File --> Save** to save the changes.

16. Close and reopen the model for the change to take effect.

**Add a list by editing the .inp file**

1. Create a custom stiffener detail (page 93) with all the needed variables that control which stiffener plates are created.

   In our example, the variable is called **Plates created**.

2. In the model, click **File --> Open the model folder** to open the current model folder.

3. Go to the **CustomComponentDialogFiles folder**.

4. Open the **.inp file** in a text editor.
5. Remove the following line:
parameter("Plates created", "P4", integer, number, 2)

6. Add a new **Plates created** attribute with the following settings:

```
attribute("Plates created", label, "%s", none, none, "0", "0", 334, 118)
```

7. Add a new **P4** attribute with the following settings:

```
attribute("P4", "", option, "%s", none, none, "0.0", "0.0", 360, 151, 90)
```

The list now contains three options, and **Both** is the default value. The list options are linked to the variable **P4** that controls the creation of the stiffener plates.

Modify the dialog box of a custom component 109 Example: Modify the dialog box of a stiffener detail
8. Edit the line numbers so that there are no empty rows between the variables in the dialog box.

   page("TeklaStructures",""")
   { 
      detail(1, "Stiffeners")
      { 
         tab_page("", " Parameters 1 ", 1)
            { 
               parameter("Stiffener set back", "Pl", distance, number, 1)
               parameter("Left plate class", "Lt", integer, number, 1)
               attribute("", "Plates created", label, "%", none, none, "0", "0", 334, 118)
               attribute("Pl", "", option, "%", none, none, "0.0", "0.0", 360, 151, 90)
            { 
               value ("Left", 0)
               value ("Right", 0)
               value ("Both", 1)
            } 
      } 

9. Browse for the images you want to use in the dialog box.

   If you make new images, make sure they are in the bitmap (.bmp) format. Save the images in the ..\ProgramData\Tekla Structures \<version>\Bitmaps folder.

10. Replace the option texts with the actual filenames of the images, but with the filename extension .xbm.

    page("TeklaStructures",""")
    { 
       detail(1, "Stiffeners")
       { 
          tab_page("", " Parameters 1 ", 1)
             { 
                parameter("Stiffener set back", "Pl", distance, number, 1)
                parameter("Left plate class", "Lt", integer, number, 1)
                parameter("Right plate class", "Rt", integer, number, 1)
                attribute("", "Plates created", label, "%", none, none, "0", "0", 334, 118)
                attribute("Pl", "", option, "%", none, none, "0.0", "0.0", 360, 151, 90)
             { 
                value ("CC_Left.xbm", 0)
                value ("CC_Right.xbm", 0)
                value ("CC_Both.xbm", 1)
             } 
      } 

11. Save the .inp file.

12. Close and reopen the model for the change to take effect.

Example: Arrange text boxes and labels

In this example, we will arrange the text boxes and labels around a list in the dialog box. You can do this either in the custom component dialog editor or by editing the input (.inp) file manually.
When we begin, the dialog box looks like the following:

We will arrange the dialog box elements more nicely, in the following manner:

Arrange the elements by using the dialog editor
1. Create a custom stiffener detail (page 93) with all the needed variables that control the creation of stiffener plates.
2. Open the stiffener dialog box for editing.
   a. In the model, select the custom stiffener detail.
   b. Right-click and select Edit Custom Component Dialog Box.
3. Drag the Plates created label above the list with images.
4. Drag the Left plate class label and the corresponding text box to the left side of the list.
5. Drag the Right plate class label and the corresponding text box to the right side of the list.
6. Drag the Stiffener set back label and the corresponding text box underneath the list.
7. In the dialog editor, click File --> Save to save the changes.
8. Close and reopen the model for the change to take effect.
**Arrange the elements by editing the .inp file**

1. Create a custom stiffener detail (page 93) with all the needed parametric variables that control the creation of stiffener plates.
2. In the model, click **File --> Open the model folder** to open the current model folder.
3. Go to the **CustomComponentDialogFiles folder**.
4. Open the **.inp file** in a text editor.
5. Edit the file as follows:
   ```plaintext
   page("TeklaStructures","")
   detail(1,"Stiffeners")
   { 
     tab_page(",", "Parameters 1 ", 1)
     { 
       attribute(",", "Plates created", label, "%c", none, none, "0", "0", 334, 118)
       attribute("M4", ",", option, "%s", none, none, "0.0", "0.0", 160, 151, 50)
       { 
         value("CC_left.xbm", 0)
         value("CC_Right.xbm", 0)
         value("CC_Both.xbm", 1)
       }
       attribute(",", "Left plate class", label, "%s", none, none, "0", "0", 125, 157)
       attribute(",", "Right plate class", label, "%s", none, none, "0", "0", 497, 160)
       parameter("", "LeftC", integer, number, 146, 192, 160)
       parameter("", "RightC", integer, number, 522, 194, 160)
       parameter(",", "Pl", distance, number, 357, 289, 160)
       attribute(",", "Stiffener set back", label, "%s", none, none, "0", "0", 330, 255)
     }
   }
   }

6. Save the **.inp file**.
7. Close and reopen the model for the change to take effect.

**Example: Dim unavailable options**

In this example, we will dim the unavailable options in the stiffener dialog box based on conditions. You can do this either in the custom component dialog editor or by editing the input (**.inp**) file manually.

When we begin, all the options are available:

![Stiffener dialog box](image)

We will define that the **Left plate class** text box is unavailable if only the right plate is created, and vice versa.
Dim unavailable options by using the dialog editor

1. Create a custom stiffener detail (page 93) with all the needed parametric variables that control the creation of stiffener plates.

2. Open the stiffener dialog box for editing.
   a. In the model, select the custom stiffener detail.
   b. Right-click and select Edit Custom Component Dialog Box.

3. Define that the Left plate class text box must be dimmed if only the right stiffener plate is created.
   a. In the Plates created list, select the image for the right plate class. Note that a blue selection border must be displayed for the image:

   ![Plates created](image)

   b. Hold down the Ctrl key and click the Left plate class text box.

   ![Left plate class](image)

   c. Click the Toggle visibility button.
4. Unselect the **Left plate class** text box by clicking the **Right plate class** text box.

5. Define that the **Right plate class** text box must be dimmed if only the left stiffener plate is created.
   a. In the **Plates created** list, select the image for the left plate class. Note that a blue selection border must be displayed for the image:

   ![Image of plates created with left plate selected]

   b. Hold down the **Ctrl** key and select the **Right plate class** text box.

   ![Image of plates created with right plate selected]

   c. Click the **Toggle visibility** button.

   The **Left plate class** text box is now dimmed:

   ![Image of plates created with left plate dimmed]

6. In the dialog editor, click **File --> Save** to save the changes.

7. Close and reopen the model for the change to take effect.
**Dim unavailable options by editing the .inp file**

1. Create a custom stiffener detail (page 93) with all the needed parametric variables that control the creation of stiffener plates.

2. In the model, click **File --> Open the model folder** to open the current model folder.

3. Go to the **CustomComponentDialogFiles** folder.

4. Open the .inp file in a text editor.

5. Add the following line to the end of the attribute P4 line:

   "toggle_field:LeftC=0;RightC=1"

   ```
   page("twistStructures","
   { detail(1, "Stiffeners")
   { tab_page("", "Parameters", 1, 1)
     { attribute("", "Plates created label", "ks", none, none, "0", "0", 354, 118)
       attribute("P4", ", option, "ks", none, none, "0", "0", 300, 151, 90, "toggle_field:leftC=0;rightC=1")
     }
     attribute("", "Right plate class label", "ks", none, none, "0", "0", 125, 157)
     attribute("", "Right plate class label", "ks", none, none, "0", "0", 497, 160)
     parameter("", "leftC", integer, number, 0, 100, 100)
     parameter("", "rightC", integer, number, 252, 394, 160)
     parameter("", "PI", distance, number, 357, 289, 100)
     attribute("", "Stiffener set back", label, "ks", none, none, "0", "0", 310, 255)
     }
   }
   }
   ```

6. Save the .inp file.

7. Close and reopen the model for the change to take effect.

---

**TIP**
If you want to hide unavailable options instead of dimming them in the stiffener dialog box, add an exclamation mark in the conditions:

"toggle_field:!LeftC=0;!RightC=1"

The option is now completely hidden when unavailable:
### 6.4 Dialog editor settings

Click **Tools --> Options** in the **Custom Component Dialog Editor** to view and modify some basic settings of the dialog editor.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image folder</strong></td>
<td>The location of the image folder. To restore the default folder setting, click <strong>Default</strong>.</td>
</tr>
<tr>
<td><strong>Project folder</strong></td>
<td>The location of the project folder. When you create a completely new input file by clicking <strong>File --&gt; New</strong> and then save it, the file is saved in the project folder. Note that existing input files are saved under the model folder.</td>
</tr>
<tr>
<td><strong>Parameter width</strong></td>
<td>The default width for text boxes.</td>
</tr>
<tr>
<td><strong>Attribute width</strong></td>
<td>The default width for lists.</td>
</tr>
<tr>
<td><strong>Grid spacing X</strong></td>
<td>The spacing of the pixel grid (page 89) in the X and Y directions. The default value is 5.</td>
</tr>
<tr>
<td><strong>Grid spacing Y</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Snap to grid</strong></td>
<td>Select to show or hide the pixel grid.</td>
</tr>
</tbody>
</table>

**See also**

*Modify the dialog box of a custom component (page 89)*
Add a custom component to a model

Use the Applications & components catalog to add your custom component to a model.

1. Click the Applications & components button in the side pane to open the Applications & components catalog.
2. To search for a component, browse the catalog or enter a search term in the search box. Custom components have the following symbols in the catalog:

<table>
<thead>
<tr>
<th>Type</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom part</td>
<td>![Symbol]</td>
</tr>
<tr>
<td>Custom connection or seam</td>
<td>![Symbol]</td>
</tr>
<tr>
<td>Custom detail</td>
<td>![Symbol]</td>
</tr>
</tbody>
</table>

3. Select the custom component you want to add.
4. Follow the instructions on the status bar to add the custom component in the model.
5. To modify the properties, double-click the custom component in the model.

Example: Add a custom connection to a model

In this example, you will add a previously created custom end plate connection to a model. Because you have not modified the custom component to adapt to different situations in the model, you need to add it to a similar location where it was created. Otherwise the custom component may not work as required.

1. Click the Applications & components button in the side pane to open the Applications & components catalog.
2. In the catalog, select the custom end plate connection you want to add. 
   Tekla Structures displays instructions on the status bar.
3. Select the column as the main part.
4. Select the beam as the secondary part.
   Tekla Structures adds the end plate connection to the model.

**See also**

Add or move a custom part in the model (page 119)
Add or move a custom part in the model

Use the direct modification handles and dimensions when you add or move custom parts. If you are unable to select custom parts in the model, ensure that the **Select components** selection switch is active.

**NOTE** This method cannot be used when adding custom parts to surfaces that have cuts or edge chamfers. You need to hide the cutting parts and edge chamfer objects from the view before you add custom parts on cut or chamfered surfaces using direct modification.

We do not recommend using this method with custom parts that are parametric, and in which the input points define the dimensions of the custom part. The preview is simplified, based on the default custom part dimensions, and snapping has a different focus than usually.

1. Ensure that **Direct modification** is switched on.

2. Click the **Applications & components** button in the side pane to open the **Applications & components** catalog.

3. In the catalog, select the custom part you want to add.

4. Move the mouse pointer over part faces and edges in the model, and see how the custom part turns over and adjusts to the part faces.

   If you are adding the custom part to another part, Tekla Structures shows location dimensions from the first input point of the custom part to the nearest part faces.

   If you are adding a custom part that has only one input point, press **Tab** to rotate it in 90-degree steps around the work plane Y axis.

5. Depending on the number of custom part input points, pick one or two points to place the custom part in the model.
Tekla Structures shows the coordinate axes, rotation handles, and location dimensions that you can use to fine-tune the location and rotation of the custom part. The handles are red, green, and blue, according to the local coordinate system of the custom part.

6. Click the middle mouse button to confirm the location and rotation. Tekla Structures adds the custom part to the model.

7. To move the custom part along any of its coordinate axes, drag the relevant axis handle to a new location.

8. To rotate the custom part around any of its coordinate axes, drag the relevant rotation handle to a new location.
9. To move or rotate the custom part by specifying a distance or angle:
   a. Select an axis handle, a rotation handle, or a dimension arrowhead.
   b. Type the value by which you want the dimension to change.
      When you start typing, Tekla Structures displays the **Enter a Numeric Location** dialog box.
   c. Click **OK** to confirm the new dimension.
10. To stop modifying, press **Esc**.
You can import and export custom components as .uel files between models.

**TIP** You can share your custom components in Tekla Warehouse, and also download custom components made by other users.

### 9.1 Import custom components

You can import previously made custom components to another model.

1. Click the **Applications & components** button in the side pane to open the **Applications & components** catalog.
2. Click the **Access advanced features** button, and then select **Import**.
4. Select the export file.
5. Click **Open** to import the custom components.

**TIP** You can import custom components to a new model automatically by using the XS UEL IMPORT FOLDER advanced option. Export all custom components to certain folders and enter these folders as the value for the XS UEL IMPORT FOLDER advanced option to easily import the custom components to new models.
9.2 Export custom components
You can export custom components in an .uel file. Do not change the file name after exporting the custom components.

1. Click the Applications & components button in the side pane to open the Applications & components catalog.
2. In the catalog, select the custom components you want to export.
3. Right-click the selection, and then select Publish.
5. Enter a name for the export file.
   By default, the file name extension is .uel.
6. Click Save to export the custom components.

TIP If you want to export custom components as separate files, select the custom components in the Applications & components catalog, right-click, and then select Publish separately.
Here you will find more information about the various custom component properties and plane types.

- **Custom component properties (page 124)**
  You must define these properties when you create new custom components. You can change some of these properties when you modify an existing custom component.

- **Default properties of a custom component (page 127)**
  Each custom component has a dialog box that you can modify. By default, the dialog box has a **Position** tab for custom parts and a **General** tab for custom connections, details, and seams.

- **Plane types (page 132)**
  When you create distance variables for a custom component, you must select a plane type. The plane type defines what planes you can select.

- **Variable properties (page 136)**
  Use the **Variables** dialog box to define properties for distance and parametric variables.

### 10.1 Custom component properties

You must define these properties when you create new custom components with the **Custom Component Wizard**. You can change some of these properties when you modify an existing custom component.

For more information, see **Create a custom component (page 16)** and **Modify a custom component (page 31)**.

**Type/Notes tab properties**

On the **Type/Notes** tab, you have the following options:
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Select the type of the custom component. Type affects how you insert the custom component in the model. Type also defines if the custom component connects to existing parts.</td>
</tr>
<tr>
<td>Name</td>
<td>Enter a unique name for the custom component.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a short description for the custom component. Tekla Structures shows the description in the Applications &amp; components catalog.</td>
</tr>
<tr>
<td>Component identifier</td>
<td>Enter an additional name or reference for the component, for example a design code reference. This can be shown in general arrangement and assembly drawings, and in lists. To show this in drawings, include Code in the Connection Mark Properties dialog box.</td>
</tr>
</tbody>
</table>

**Position tab properties**

On the Position tab, you have the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up direction</td>
<td>Sets the default up direction.</td>
<td>Not available for parts.</td>
</tr>
<tr>
<td>Position type</td>
<td>The position (or origin) of the component, relative to the main part.</td>
<td>Not available for details and parts.</td>
</tr>
</tbody>
</table>

You can define the position for custom connections and seams. You have the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middle</td>
<td>Where the center lines of the main and secondary parts intersect.</td>
<td><img src="image" alt="Middle Example" /></td>
</tr>
<tr>
<td>Box plane</td>
<td>Where the main part box and the center line of the secondary part intersect.</td>
<td><img src="image" alt="Box plane Example" /></td>
</tr>
</tbody>
</table>
### Advanced tab properties

On the Advanced tab, you have the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail type</td>
<td>Determines on which side of the main part the component is located. The options are:</td>
<td>Only available for details and seams</td>
</tr>
<tr>
<td></td>
<td>• Intermediate detail</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tekla Structures creates all components on the same side of the main part</td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Note</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td><strong>End detail</strong></td>
<td>Tekla Structures creates all components on the side of the main part closest to the details.</td>
<td>Only affects asymmetric components.</td>
</tr>
<tr>
<td><strong>Definition point position in relation to primary part</strong></td>
<td>Determines the position you pick to create the detail, relative to the main part.</td>
<td>Only available for details</td>
</tr>
<tr>
<td><strong>Definition point position in relation to secondary part</strong></td>
<td>Determines where the component is created, relative to the secondary part.</td>
<td>Only available for connections and seams</td>
</tr>
<tr>
<td><strong>Allow multiple instances of connection between same parts</strong></td>
<td>Select this option to create many components to the same main part, in different locations.</td>
<td>Only available for connections and seams</td>
</tr>
<tr>
<td><strong>Exact positions</strong></td>
<td>Select this option to position the seam based on the positions that you pick in the model.</td>
<td>Only available for seams</td>
</tr>
<tr>
<td>Clear the check box to let Tekla Structures use automatic seam recognition to position the seam. This is useful especially with warped seams.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Use the center of the bounding box in positioning</strong></td>
<td>Select to position the custom part based on the center of its bounding box (the box that surrounds the actual part profile).</td>
<td>Only available for parts</td>
</tr>
</tbody>
</table>
10.2 Default properties of a custom component

Each custom component has a dialog box that you can modify. By default, the dialog box has a Position tab for custom parts and a General tab for custom connections, details, and seams.

For more information, see Modify the dialog box of a custom component (page 89).

To view the current properties, double-click the custom component in the model.

Default properties of custom parts

By default, the dialog box of a custom part has the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>On plane</td>
<td>Changes part location on the work plane.</td>
<td>Middle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Left</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Rotation</td>
<td>Rotates the part in steps of 90 degrees.</td>
<td>Top and Below</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Front and Back</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>At depth</td>
<td>Changes part location perpendicular to the work plane.</td>
<td>Middle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Front</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Behind</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>Show third handle</strong></td>
<td>Sets the third handle of a nested custom part visible in the desired direction. You can bind the third handle in the desired direction and thus force the part to follow the rotation of another part.</td>
<td><img src="image1" alt="Example Image" /></td>
</tr>
<tr>
<td>None</td>
<td>None</td>
<td><img src="image2" alt="Example Image" /></td>
</tr>
<tr>
<td>Above</td>
<td>Above</td>
<td><img src="image3" alt="Example Image" /></td>
</tr>
<tr>
<td>On the left</td>
<td>On the left</td>
<td><img src="image4" alt="Example Image" /></td>
</tr>
</tbody>
</table>

**Default properties of custom connections, details, and seams**

By default, the dialog box of a custom connection, detail, or seam has the following options:

Custom component settings  130  Default properties of a custom component
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up direction</td>
<td>Indicates how the component is rotated around the secondary part, relative to the current work plane. If there are no secondary parts, Tekla Structures rotates the connection around the main part.</td>
<td></td>
</tr>
<tr>
<td>Position in relation to primary part</td>
<td>The creation point of the component, relative to the main part.</td>
<td>Only available for details.</td>
</tr>
<tr>
<td>Position in relation to secondary part</td>
<td>Tekla Structures automatically places the component according to the selected option.</td>
<td>By default, only available for seams.</td>
</tr>
<tr>
<td>Place to picked positions</td>
<td>Select this option to place the seam at the points you pick.</td>
<td>Only available for seams.</td>
</tr>
<tr>
<td>Detail type</td>
<td>Determines on which side of the main part the component is located. The options are:</td>
<td>Only available for details.</td>
</tr>
</tbody>
</table>
|                             | • Intermediate detail  
--- Tekla Structures creates all components on the same side of the main part.                                                                                                                    |                                                                     |
|                             | • End detail  
--- Tekla Structures creates all components on the side that is closest to the details.                                                                                                               | Only affects asymmetric components.                                  |
| Locked                      | Select Yes to prevent other users from modifying the properties.                                                                                                                                             |                                                                     |
### Option Description Note

**Class**
The class of the parts that the custom component creates.

**Connection code**
Identifies the component. You can display this connection code in connection marks in drawings.

**AutoDefaults rule group**
The rule group used for setting the connection properties.

**AutoConnection rule group**
The rule group Tekla Structures uses to select the connection.

---

#### 10.3 Plane types

When you add distance variables to a custom component, you must select a plane type. The plane type defines what planes you can select.

You have the following options:

<table>
<thead>
<tr>
<th>Plane type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Boundary planes</strong></td>
<td>You can select the edges of a bounding box that surrounds the profile.</td>
<td><img src="image" alt="Boundary planes" /></td>
</tr>
<tr>
<td>Plane type</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Center planes</td>
<td>You can select the center planes of a profile.</td>
<td><img src="image1" alt="Image" /></td>
</tr>
<tr>
<td>Outline planes</td>
<td>You can select the outer and inner surfaces of a profile.</td>
<td><img src="image2" alt="Image" /></td>
</tr>
<tr>
<td>Cut planes</td>
<td>If the part contains line, part, or polygon cuts, this option enables you to select cut surfaces. Fittings cannot be selected.</td>
<td><img src="image3" alt="Image" /></td>
</tr>
<tr>
<td>Component planes</td>
<td>What you can select depends on the component type and the <strong>Position type</strong> of the custom component.</td>
<td><img src="image4" alt="Image" /></td>
</tr>
</tbody>
</table>
Examples of component planes
See below for examples of possible component planes. What you can select depends on the component type and the Position type of the custom component.

Part component planes

Connection component planes
Detail component planes

Seam component planes

Custom component settings  

Plane types
10.4 Variable properties

Use the Variables dialog box to view, modify, and create parametric variables, and to view fixed and reference distance variables.

Tekla Structures uses variables with custom components (page 34), sketched cross sections, and parametric modeling. The examples below are given for custom components, but the same principles apply also to sketched cross sections and parametric modeling.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique name of a variable. Use this name to refer to the variable in the custom component editor. The maximum length is 19 characters.</td>
</tr>
<tr>
<td>Formula</td>
<td>Use this box to enter a value or a formula. Formulas begin with '='.</td>
</tr>
<tr>
<td>Value</td>
<td>Shows the current value of Formula.</td>
</tr>
<tr>
<td>Value type</td>
<td>Select a value type from the list. The type determines what kind of value you can enter for the variable. For more information, see the table below.</td>
</tr>
<tr>
<td>Variable type</td>
<td>This property can be either Distance or Parametric.</td>
</tr>
<tr>
<td>Visibility</td>
<td>Use this setting to control the visibility of a variable. Set to Show to display the variable in the custom component dialog box.</td>
</tr>
<tr>
<td>Label in dialog box</td>
<td>The name of the variable that Tekla Structures displays in the custom component dialog box. The maximum length is 30 characters.</td>
</tr>
</tbody>
</table>

You have the following options for the value type:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>A whole (integer) number. Use for quantity and multiplier.</td>
</tr>
<tr>
<td>Length</td>
<td>A decimal (floating point) number. Use for lengths and distances. Length numbers have unit (mm, inch, etc.) and they are rounded to two decimal places.</td>
</tr>
<tr>
<td>Text</td>
<td>A text (ASCII) string.</td>
</tr>
<tr>
<td>Factor</td>
<td>A decimal value without a unit. You can set the number of decimals for the value type in File menu --&gt; Settings --&gt; Options --&gt; Units and decimals.</td>
</tr>
<tr>
<td>Angle</td>
<td>A decimal number type for storing angles, stored to one decimal place, in radians.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>A data type associated with the material catalog. Use to select a material from the standard material dialog.</td>
</tr>
<tr>
<td><strong>Profile</strong></td>
<td>A data type associated with the profile catalog. Use to select a profile from the standard profile dialog.</td>
</tr>
<tr>
<td><strong>Bolt size</strong></td>
<td>Data types linked to the bolt catalog. <strong>Bolt size</strong> works with <strong>Bolt standard</strong>. They have a fixed naming format: Px_diameter and Px_screwdin. Do not change the fixed name.</td>
</tr>
<tr>
<td><strong>Bolt standard</strong></td>
<td>To show values for these in the component’s dialog box, x must be the same for both, for example, P1_diameter and P1_screwdin.</td>
</tr>
<tr>
<td><strong>Bolt type</strong></td>
<td>For determining the bolt type (site/workshop) in the custom component dialog box. Linked to the <strong>Bolt type</strong> property of bolts in the <strong>Custom component browser</strong>.</td>
</tr>
<tr>
<td><strong>Stud size</strong></td>
<td>Data types linked to the bolt catalog. <strong>Stud size</strong>, <strong>Stud standard</strong> and <strong>Stud length</strong> work together. They have a fixed naming format: Px_size, Px_standard and Px_length. Do not change the fixed names.</td>
</tr>
<tr>
<td><strong>Stud standard</strong></td>
<td>To show values for these in the component’s dialog box, x must be the same for all of them. For example, P9_size, P9_standard, and P9_length.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Distance list</td>
<td>Use with options that have several length values, such as bolt spacings. Use space as a separator between the distances.</td>
</tr>
<tr>
<td>Weld type</td>
<td>A data type for selecting the weld type.</td>
</tr>
<tr>
<td>Chamfer type</td>
<td>A data type for determining the shape of a chamfer. For more information, see Corner chamfer types and dimensions.</td>
</tr>
<tr>
<td>Welding site</td>
<td>A data type for determining the welding place: workshop or building site.</td>
</tr>
<tr>
<td>Rebar grade</td>
<td>Data types linked to reinforcement catalog. <strong>Rebar grade</strong>, <strong>Rebar size</strong>, and <strong>Rebar bending radius</strong> work together. They have a fixed naming format: P&lt;sub&gt;x&lt;/sub&gt;_grade, P&lt;sub&gt;x&lt;/sub&gt;_size, and P&lt;sub&gt;x&lt;/sub&gt;_radius. Do not change the fixed name.</td>
</tr>
<tr>
<td>Rebars to split</td>
<td>Used for rebar set splitters, to specify how the bars are to be split (1/1, 1/2, and so on).</td>
</tr>
<tr>
<td>Rebar hook type</td>
<td>Used for rebar set end detail modifiers, to specify the hook type.</td>
</tr>
</tbody>
</table>

### Distance list description

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula</th>
<th>Value</th>
<th>Value type</th>
</tr>
</thead>
<tbody>
<tr>
<td>P&lt;sub&gt;3&lt;/sub&gt;_size</td>
<td>6.35</td>
<td>NELSON</td>
<td>Stud size</td>
</tr>
<tr>
<td>P&lt;sub&gt;3&lt;/sub&gt;_standard</td>
<td>NELSON</td>
<td></td>
<td>Stud standard</td>
</tr>
</tbody>
</table>

### Weld type description

- **Stud size** (6.35, NELSON)
- **Stud standard** (NELSON)

### Chamfer type description

For more information, see Corner chamfer types and dimensions.

### Welding site description

- **Stud size** (6.35, NELSON)
- **Stud standard** (NELSON)

### Rebar grade, Rebar size, Rebar bending radius

Data types linked to reinforcement catalog. **Rebar grade**, **Rebar size**, and **Rebar bending radius** work together. They have a fixed naming format: P<sub>1</sub>_grade, P<sub>1</sub>_size, and P<sub>1</sub>_radius. Do not change the fixed name.

To show values for these in the component’s dialog box, x must be the same for all, for example, P<sub>1</sub>_grade, P<sub>1</sub>_size, and P<sub>1</sub>_radius.

- **Rebar grade** (P<sub>1</sub>_grade, P<sub>1</sub>_size, P<sub>1</sub>_radius)
- **Rebar size** (P<sub>1</sub>_size, P<sub>1</sub>_radius)
- **Rebar bending radius** (P<sub>1</sub>_radius)
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rebar stagger type</strong></td>
<td>Used for rebar set splitters, to specify the stagger type (left/right/middle).</td>
</tr>
<tr>
<td><strong>Rebar lap side</strong></td>
<td>Used for rebar set splitters, to specify the side of the overlap (left/right/middle).</td>
</tr>
<tr>
<td><strong>Rebar lap placement</strong></td>
<td>Used for rebar set splitters, to determine whether the lapping bars are parallel to each other or on top of each other.</td>
</tr>
<tr>
<td><strong>Rebar lap type</strong></td>
<td>Used for rebar set splitters, to determine whether the reinforcing bars are kept straight at lap splices by offsetting entire bars, or placed slanted by offsetting bar ends.</td>
</tr>
<tr>
<td><strong>Reinforcement mesh</strong></td>
<td>For determining meshes in custom components. Linked to the <em>Catalog name</em> property of reinforcement meshes in the <em>Custom component browser</em>.</td>
</tr>
<tr>
<td><strong>Cross bar location</strong></td>
<td>Used for rebar meshes, to determine whether the crossing bars are located above or below the longitudinal bars.</td>
</tr>
<tr>
<td><strong>Component name</strong></td>
<td>Use <em>Component name</em> for replacing a sub-component inside a custom component with another sub-component. Linked to the <em>Name</em> property of objects in the <em>Custom component browser</em>.</td>
</tr>
<tr>
<td><strong>Component attribute file</strong></td>
<td>Use <em>Component attribute file</em> for setting the properties of a sub-component inside a custom component.</td>
</tr>
<tr>
<td></td>
<td><em>Component name</em> and <em>Component attribute file</em> work together. They have a fixed naming format: P_x_name and P_x_attrfile. Do not change the fixed name.</td>
</tr>
<tr>
<td></td>
<td>To show values for these in the component's dialog box, x must be the same for both, for example, P2_name and P2_attrfile.</td>
</tr>
<tr>
<td><strong>Yes/No</strong></td>
<td>For determining whether or not Tekla Structures creates an object in a custom component. Linked to the <em>Creation</em> property of objects in the <em>Custom component browser</em>.</td>
</tr>
<tr>
<td><strong>Bitmask</strong></td>
<td>For defining bolt assembly (nuts and washers) and parts with slotted holes. Linked to the <em>Bolt structure</em> and <em>Parts with</em></td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>slotted holes</td>
<td>properties of bolts in the Custom component browser. The value is a five-digit series of ones and zeros. This relates to the check boxes in the Bolt Properties dialog box. 1 means that a check box is selected, 0 means that a check box is clear. In the example below, the value of 10010 means that a bolt with a washer and a nut is created in the bolt assembly.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bolt Structure</th>
<th>10010</th>
</tr>
</thead>
</table>

Include in bolt assembly: ✓ ✓ ✓ ✓

In the example below, the value of 10010 means that a bolt with a washer and a nut is created in the bolt assembly.
Here you will find some useful tips on how to create and use custom components more efficiently.

- **Tips for creating custom components (page 141)**
  Follow these guidelines when creating new custom components.

- **Tips for sharing custom components (page 142)**
  Follow these guidelines when sharing custom components with colleagues.

- **Tips for updating custom components to a new version (page 142)**
  When you start using a new version of Tekla Structures, always check that custom components created in older versions work correctly in the new version.

### 11.1 Tips for creating custom components

Follow these guidelines when creating new custom components.

- **Enter short, logical names for custom components.**
  Use the description field to describe the component and to explain what it does.

- **Create simple components for specific situations.**
  Simple components are easier and faster to model, and also much easier to use. Avoid creating a single, complex component which you will use for every possible purpose.

- **Consider creating a separate component model.**
  Use that model when you create and test custom components.
• **Use the simplest part you can.**
  For example, if all you need is a rectangular shape, use a rectangular plate, not a contour plate. Rectangular plates only have two handles, so you only need to create a few bindings to manipulate them. Contour plates require more bindings because they have four handles.

1. Rectangular plate  
2. Contour plate

• **Model parts only as accurately as you need.**
  If the only part information required is a part mark in a general arrangement drawing, plus a quantity on a materials list, create a simple bar or plate. If you need to include the part in a detailed view later on, simply re-model the part more accurately at that point.

• **Model embeds as custom parts and include them in components.**

### 11.2 Tips for sharing custom components

Follow these guidelines when sharing custom components with colleagues.

• **Use Tekla Warehouse to share and store custom components.**

• **Provide essential information.**
  If you distribute your component to other users, remember to list the profiles it works with.

• **Use fixed profiles whenever possible.**

• **If your custom component contains user-defined profile cross sections, remember to include them when you copy the custom component to a new location.**
11.3 Tips for updating custom components to a new version

When you start using a new version of Tekla Structures, always check that custom components created in older versions work correctly in the new version.

When you edit custom components created with an older version of Tekla Structures, and the new version contains improvements that require an update, Tekla Structures asks whether you want to update the component. If you do not update the component, it works in the same manner as in the version where it was originally created, but you do not gain the benefits of the new improvements.

If you choose to update the component, you need to check and sometimes recreate dimensions depending on the improvements. When you delete a dimension and create a new one (even with the same name), the equations that contain the dimension also need to be modified, because the dependency created by the equation is lost when a dimension is deleted. You can recreate dimensions and modify equations in the custom component editor.
Hints and tips for using custom components

Tips for updating custom components to a new version
Index

A
Applications & components...........................5
  exporting custom components........... 122
  importing custom components...........122
  arithmetic operators.............................55

B
binding objects
  plane types.............................................132
  to a plane................................................. 35
  boundary planes........................................ 132

calculations.................................................... 53
center planes...............................................132
closing
  custom component editor..................... 88
  component planes.............................. 132,133
  component symbol................................. 5
  components
    exploding................................................... 19
    nested components.............................. 19
    sub-components..................................... 19
connections
  custom connections....................................5,9
  construction lines.....................................43
  construction planes....................................43
copying
  property references.................................... 51
creating
  custom components..................................... 16
Custom component browser.................... 31,51
Custom component dialog editor............. 89
  setting options........................................ 116
custom component editor........................... 31
  closing....................................................... 88
  saving........................................................ 88
custom component view................................. 31
Custom component wizard........................... 16
  properties............................................... 124
custom components
  about custom components........................... 5
  adding to model......................................... 117
  arithmetic operators............................. 55
  binding objects...................................... 35,43
  construction planes and lines.....................43
  copying property references........................ 51
  creating................................................. 16
  default properties.................................. 127
  dialog box files...................................... 92
  exporting............................................... 122
  importing............................................... 122
  inp files............................................... 92
  locking................................................... 92
  modifying............................................... 31
  modifying the dialog box...........................89
  nested components................................. 19
  plane types.............................................132
  preventing modifications..............................92
  protecting with passwords...........................32
  seams..................................................... 13
  settings................................................... 124
  sub-components........................................ 19
  thumbnail images......................................21
  types............................................................ 5
  variables................................................. 34
custom connections...................................... 9
custom details............................................ 11
custom parts............................................. 7
  adding to model....................................... 119
custom seams............................................. 13
cut planes.................................................. 132
cyclic dependencies.................................... 67

dependencies
  in variable formulas.................................. 67

D
145
<table>
<thead>
<tr>
<th>Detail</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>custom details</td>
<td>5, 11</td>
</tr>
<tr>
<td>dialog boxes</td>
<td>89</td>
</tr>
<tr>
<td>modifying</td>
<td>89</td>
</tr>
<tr>
<td>distance variables</td>
<td>34, 35</td>
</tr>
<tr>
<td>distance reference distance</td>
<td>45</td>
</tr>
</tbody>
</table>

**E**

editing

<table>
<thead>
<tr>
<th>Custom component dialog editor</th>
<th>89</th>
</tr>
</thead>
</table>

examples

| adding option to create object in custom component | 70 |
| component planes | 133 |
| creating custom connection | 23 |
| creating parametric variable | 69 |
| determining bolt group distance from beam flange | 78 |
| determining bolt size and bolt standard | 77 |
| determining number of bolt rows in custom component | 80 |
| exploded components | 23 |
| modifying custom component dialog box | 93 |
| modifying dialog box of stiffener detail | 93 |
| nested components | 26 |
| replacing sub-components in custom component | 71 |
| sub-components | 26 |
| using construction planes in custom component | 26 |
| using Excel spreadsheets with custom components | 86 |
| using properties file in custom component | 73 |
| using template attributes in custom components | 84 |
| using user-defined attributes in custom components | 82 |

**Excel**

| using with custom components | 86 |

exploding components | 19 |

exporting

| custom components | 122 |

**F**

formulas

| arithmetic operators | 55 |
| in custom components | 53 |
| functions | 53 |

**G**

grid planes | 132 |

**I**

importing

| custom components | 122 |
| inp files | 89, 92 |

locking and unlocking

| custom components | 92 |

**L**

linking

| parametric variables to object properties | 48 |

**M**

magnetic construction planes and lines | 43 |

modifying

| custom components | 16, 31 |

**N**

nested components | 19, 26 |