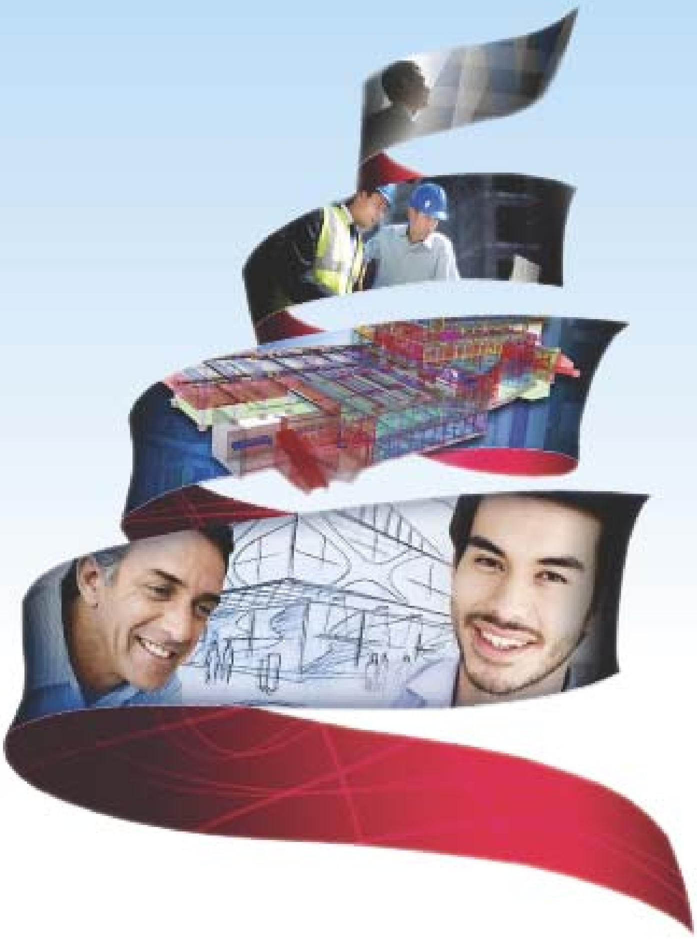


Template Editor

User's Guide

Product version 3.33

September 2014

Conventions used in this guide

Typographical conventions

The following typographical conventions are used in this guide:

Font	Usage
Bold	Any text that you see in the user interface appears in **bold**. This font is used, for example, for window and dialog box titles, box and button names, and list items.
Italic bold	New terms are in ***italic bold*** when they appear in the current context for the first time.
`Monospace`	Extracts of program code, HTML, or other material that you would normally edit in a text editor, appear in `monospaced` font. This font is also used for file names and folder paths, and for any text that you should type yourself.

Noteboxes

The following types of noteboxes are used in this guide:

A **tip** might introduce a shortcut, or suggest alternative ways of doing things.

A **note** draws attention to details that you might easily overlook. It can also point you to other information in this guide that you might find useful.

You should always read very **important notes and warnings**, like this one. They will help you avoid making serious mistakes, or wasting your time.

This symbol indicates **advanced or highly technical information** that is usually of interest only to advanced or technically-oriented readers.

Contents

1 About Template Editor

Template Editor, also known as TplEd or TempEd is for creating, editing and managing template definitions in your products.

You can produce labels, reports and legends using Template Editor, allowing you to gather and produce accurate and targeted information your product.

Tekla Structures features a number of ready-made templates and you can use Template Editor to edit the existing template or create new ones to suit your needs.

See also Main features of Template Editor (9)

1.1 Main features of Template Editor

With Template Editor you can do the following:

- Print data from your product, either textual or graphical.
- Modify the size and use of columns and customize template output.
- Print different page headers and footers by defining one of many different output options.
- Customize the way the row component is output when creating listings of product objects.
- Build a row hierarchy that reassembles the hierarchy of objects in the product database, and also output a part list of a product object.
- Exclude value fields or rows from the output template.
- Use free attributes to customize the way the template is output. Free attributes are product-specific.
- Use AutoCAD and MicroStation files in templates.
- Print company logos from picture files and use symbols from your product symbol files.
- Create simple HTML reports.

See also User interface overview (13)

Column settings (43)

Template output options (82)

Template field objects (11)

Hiding rows from output (45)

Hiding value fields from output (55)

Free attributes (78)

AutoCAD and Microstation files (67)

Symbols and pictures (63)

Creating an HTML report (86)

1.2 What are templates?

Templates are descriptions of forms and tables that can be included in your products. Templates are either graphical or textual. The contents of the template fields are filled in by the product at run time.

Templates are used for a variety of different purposes, for example, to print a list of parts used in a construction assembly, to denote the legend on an electrical network map, or to provide statistics on the contents of the map such as city area, scale or date.

See also Graphical templates (10)

Textual templates (10)

Graphical templates

Graphical templates are generally used to display map legends and labels, or project and company information. In addition to text, they can contain graphics, such as tables, pictures or symbols. Graphical templates can also employ different font types and settings. Graphical template definitions have the file extension **.tpl**.

See also Textual templates (10)

Textual templates

Textual templates only contain text. They are primarily used for creating reports or listings of application area-specific objects, for example in material lists of steel construction assemblies in Tekla Structures.

You can modify the column settings in order to print templates with multiple columns.

You can use various font types in textual templates, but they do not appear in the output template. If you want to use specific font types in template output, you have to use a graphical template even if the template contains only text.

Textual template definitions have the file extension **.rpt**.

See also Graphical templates (10)

Font properties for objects (39)

Column settings (43)

File Location preferences (18)

1.3 Template components

Template layout is designed with template components. Templates can be made up of five different component types, but not all components have to be present to build a template. There are many different options for specifying the output policy.

Header	A header appears once at the beginning of a template.
Page header	A page header appears at the beginning of template page.

Row	A template can have multiple rows. A row defines the things that are listed in the template, each row usually representing an object from the product database. Rows contain field objects that define the attributes to get from the database.
Page footer	A page footer is output at the end of a template page.
Footer	A footer appears once at the end of a template.

See also Template rows (41)

Inserting components and objects (29)

Modifying the name of a template component or object (33)

Modifying the size of a template component or object (34)

1.4 Template objects

Template objects are inserted into template components and they can include:

Drawing shapes	Basic geometric shapes like lines, rectangles and circles.
Text objects	Static texts are headings or title line texts.
Symbols	Symbols from product symbol libraries.
Pictures	Raster format files.
Imported files	AutoCAD and MicroStation files.
Field objects	Textual or graphical data gathered from the product database, known as value fields and graphical fields.

See also Template shapes and text (11)

Template field objects (11)

Template symbols and files (12)

Template shapes and text

Template Editor shapes include lines, arcs, polylines or polygons. They only appear in graphical templates.

Template Editor text objects are static text strings. Text objects can appear in both graphical templates and textual templates.

See also Inserting components and objects (29)

Modifying component and object properties (33)

Drawing with the numeric pad (38)

Template field objects

Template field objects are graphical fields and value fields, which can be used to get data from the product.

Value fields contain textual information that is referenced from the product database during template output. Data is retrieved by using value field formulas.

A **graphical field** is a space that is left empty until the product draws what should appear in the field: a symbol, picture or text. Graphical fields can appear only in graphical templates. Template Editor displays the area that is reserved for product graphics as a rectangle. The line color of the rectangle is copied from the default settings for rectangle objects. This outline does not appear in the final template.

See also Value fields (51)

Inserting a graphical field (31)

Adding a logo to a template (64)

Modifying component and object properties (33)

Template symbols and files

Symbols and files of different formats can be inserted into graphical templates.

See also Adding symbols to a template (63)

Adding pictures to a template (64)

Importing AutoCAD and MicroStation files with import method defined (67)

Importing AutoCAD and MicroStation files with no import method defined (68)

1.5 Template definition files

Template Editor definition files contain product-specific information that is used in rows, value fields and graphical fields. Definition files are editable text files. There are three different types of definition files: row content types, global attributes or value field meanings. The contents and syntax of the files are described at the beginning of each file.

See also Content types (42)

Value field meanings (53)

Global attributes (79)

2 User interface overview

This is an overview of the Template Editor user interface and its basic features.

Click the links below to find out more:

See also Interface components (13)

Template windows (15)

Setting up the interface (16)

Tips for the Template Editor interface (20)

2.1 Interface components

This is an overview of the Template Editor user interface components.

Click the links below to find out more:

Content Browser (14)

Workarea (13)

Preview pane (15)

Template Editor toolbars (13)

Modifying interface components (15)

Workarea

The **Workarea** is the primary area where you modify templates. It shows the selected template, so you can insert components and objects. You can customize the appearance of the **Workarea** to your liking, for example, the background and component outline colors.

If you have multiple templates open, each of them appears in its own window.

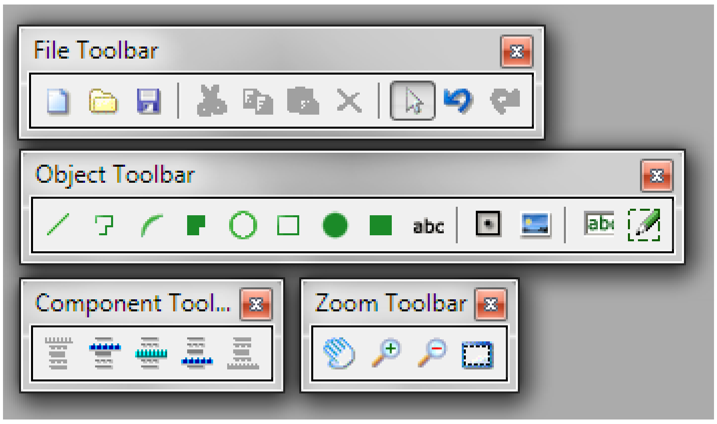
If you right-click the mouse in the **Workarea**, a pop-up menu opens access to the most common viewing and modifying operations.

See also Template windows (15)

Modifying Workarea preferences (17)

Template Editor toolbars

The toolbars contain buttons that give easy access to some of the most frequently-used commands.

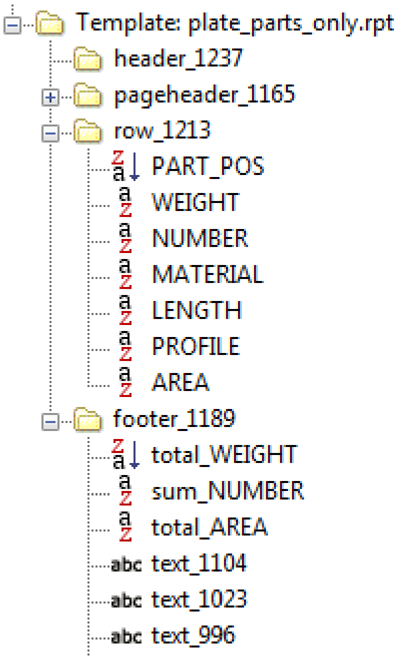


You can either dock the toolbars at the edges of the application window or leave them floating anywhere on your screen.

To show or hide a toolbar, click **View** > **Toolbars** and click the toolbar name. Visible toolbars have a check mark beside their name. Template Editor saves and maintains toolbar visibility settings across sessions.

Content Browser

In the **Content Browser** you can see a folder list that describes the contents of an open template. Each template appears in it as a top-level folder. Template components appear as sub folders, each of them containing a list of included template objects. Template components and objects are identified by their names which do not appear in the final template.



```
Template: plate_parts_only.rpt
    header_1237
    pageheader_1165
    row_1213
        PART_POS
        WEIGHT
        NUMBER
        MATERIAL
        LENGTH
        PROFILE
        AREA
    footer_1189
        total_WEIGHT
        sum_NUMBER
        total_AREA
        abc text_1104
        abc text_1023
        abc text_996
```

If you right-click in the **Content Browser**, a pop-up menu opens access to the most common viewing and modifying operations.

See also Template components and objects (29)

Modifying component and object properties (33)

Viewing or hiding template objects (39)

Searching for text (40)

Preview pane

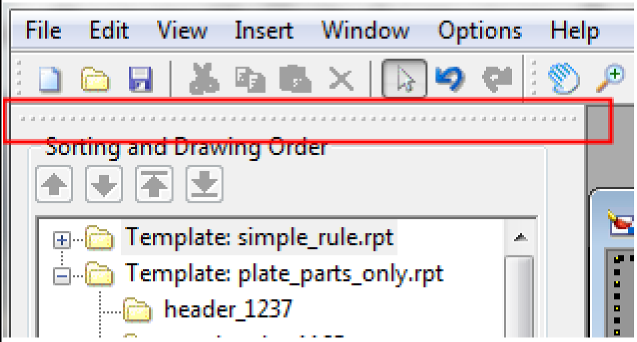
The **Preview Pane** is a read-only table within the **Content Browser**. It shows the properties of the selected template, component or object.

If you want to modify the properties, you must open the property dialog box for the selected part. You can do that by clicking **Edit** > **Properties**.

See also Content Browser (14)

Modifying interface components

You can modify the visibility and location of Template Editor interface components such as toolbars and the **Content Browser**

To...	Do this...
Dock or undock the **Content Browser**	Double-click the move handle at the top of the **Content Browser**.
Dock or undock toolbars	Double-click the move handle on the toolbar.  File Edit View Insert Window Options Help Sorting and Drawing Order Template: simple_rule.rpt Template: plate_parts_only.rpt header_1237
View or hide the **Content Browser**	Click **View** > **Content Browser** > **Visible**.
View or hide toolbars	Click **View** > **Toolbars** > and the toolbar to view or hide.

See also Hiding value fields from output (55)

Viewing or hiding template objects (39)

2.2 Template windows

Template window operations include arranging windows in the **Workarea** and focusing and zooming in a template window.

With Template Editor you can open several templates simultaneously. Each template window is displayed separately inside the **Workarea** and as a folder in the **Content Browser**.

You can cut or copy template objects from one template and paste them into another one when multiple templates are open.

See also Arranging template windows (16)

Panning (16)

Zooming in and out (16)

Restoring the original view (16)

Arranging template windows

If you open multiple templates to modify them, you can arrange the template windows by using the **Window** menu commands. This menu also contains a list of open templates. If you select a template name from the list, Template Editor brings the corresponding template window to the front and activates it.

Panning

You can use panning to focus the view to a selected point or to move the template around the **Workarea**. To pan in a template window:

1. Click the **Pan** button on the **Zoom** toolbar.
2. You can use panning in three ways:
   - *Click* to center the view around the selected point.
   - *Drag* to move the template around the **Workarea**.
   - *Drag* with the middle mouse button (this is often the same as the scroll bar button).
3. Repeat step 2 to continue panning. Press **Esc** to stop panning.

Zooming in and out

Zooming allows you to focus in on a particular area, or pull out for a wider view. To zoom on a template window:

1. If you want to **magnify** the template, click the **Zoom In** button on the **Zoom** toolbar.
   If you want to **reduce** the template to see more of it, use the **Zoom Out** button instead.
2. Click to zoom to the next preset percentage. The display centers around the point you click.
   You can also drag to center the display around the selected area.
3. Repeat step 2 to continue zooming. To stop zooming, press **Esc**.

Restoring the original view

To fit and display the whole template in its window, do one of the following:

- Click **View** > **Restore**.
- Click the **Restore view** icon on the **Zoom** toolbar.
- Right-click and select **Restore** from the pop-up menu.

2.3 Setting up the interface

Before you get started with Template Editor, you can customize the interface. Set up the **Workarea** and the grid settings. It is also critical to set up file location preferences in order for Template Editor value fields and row rules to gather template and part-related information at run time.

See also Modifying general preferences (17)

Modifying Workarea preferences (17)

File Location preferences (18)

Modifying File Location preferences (19)

Activating the grid (19)

Setting the grid density (20)

Showing or hiding the grid (20)

Modifying general preferences

General preferences allow you to select the measurement unit of graphical templates, show different buttons, set message preferences and optionally use TrueType fonts.

Click **Options** > **Preferences**. The **General** tab is active by default. You can modify the settings in the fields listed below. Click **OK** to save the changes.

To set...	Modify this...
Graphical units	Sets the measurement unit for graphical templates to millimeters or inches. All size-related information (for example height and width of components or objects) is displayed using the selected unit. In textual templates the unit is always character (char(s)).
Always ask for import method	Click this option if you want the **Select Import Method** dialog box to appear whenever you import AutoCAD or MicroStation files. If you leave the option unchecked, you will have to double-click for the **Select Import Method** dialog box to appear.
Show TrueType fonts	Select this if you want to use TrueType fonts in Template Editor. If unselected, you can use only Tekla fonts.
Reset messages to visible	Resets the user messages setting so that all messages are displayed again.

See also Importing AutoCAD and MicroStation files with import method defined (67)

Importing AutoCAD and MicroStation files with no import method defined (68)

Template properties (26)

Modifying Workarea preferences

Workarea preferences allow you to change the appearance of the **Workarea** to your liking.

These settings do not affect the output template in any way.

To modify **Workarea** preferences, click **Options** > **Preferences** and click the **Workarea** tab. You can modify the color, line width and grid color.

Click **OK** to save the changes.

Field	Description
Background	Sets the **Workarea** background color.
Grid	Sets the grid color.
Header, Page header, Row, Page footer, Footer	For each template component you can change the color of the component frame (if you use presentations, you select from pen names). You can also change the line thickness.

See also Workarea (13)

Modifying general preferences (17)

File Location preferences (18)

Template grid (19)

Setting the presentation file and pen settings (23)

File Location preferences

File location preferences define the default work directory and locations where Template Editor searches for images and product-specific definition files and directories.

Definition file	Description
Value field meanings	Sets the definition file for value field meanings.
Row contents	Sets the definitions file for row contents.
Global attributes	Sets the definition file for global attributes. If this setting is missing, no attribute name check is performed.
Work directory (*)	Sets the initial working directory for Template Editor. When opening or saving template files it is automatically active in the file selection dialog box.
Symbols, pictures (*)	Sets the folders where you can insert symbols and pictures from. You can define several directories by separating them with semicolons (;).
Fonts (*)	Sets the folders containing font files. You can define several directories by separating them with semicolons (;).
Predefined Rules	Sets the definition file for predefined rules that are applied to row output.
Presentation	Sets the presentation file from which the pen and color settings are loaded. Tekla Structures does not use presentations.
Conversion fonts	Sets the font map file to use in template conversion.
Conversion colors	Sets the color map file to use in template conversion.
Conversion attributes	Sets the attribute map file to use in template conversion.
Fill import	Sets the fill import map file to use in template conversion and in AutoCAD and MicroStation file importation.

Definition file	Description
Color Map	Sets the color map file to use.

See also Modifying File Location preferences (19)

Modifying File Location preferences

File location preferences define the default work directory and locations where Template Editor searches for images and product-specific definition files. If you leave these empty, Template Editor will not be able to find product-specific symbols, pictures and fonts that you may want to use in your templates.

To add or modify the default file locations:

1. Click **Options** > **Preferences** and click the **File Locations** tab. The list of modifiable file locations is in two columns.

   The **Type** column contains the definition file type and cannot be modified. The **Location** field contains the path for the definition file or directory and can be modified.
2. To set a definition file, click on the line that you want to modify and click the **Select File** button. This opens the standard file selection dialog box.
3. To set a directory **(*)**, click in the **Location** field on the line that you want to modify, and type in the directory path, for example `C:\Program Files\Company Product\symbols`.
4. Click **OK** to accept or **Cancel** to discard the changes.

See also File Location preferences (18)

Template grid

The template grid helps you to draw regular shapes and position template objects.

For textual templates the distance between the grid points in the **Workarea** is fixed to one character unit and you cannot change it. For graphical templates you can change the grid size to suit your needs.

See also Activating the grid (19)

Setting the grid density (20)

Showing or hiding the grid (20)

Activating the grid

Activating the grid means that template objects will be aligned to grid points when they are added to the template.

To activate the grid right-click in an empty area in the **Workarea** and click **Grid** > **Visible** from the pop-up menu.

The check mark in front of the option indicates that the grid is activated.

If you have drawn or added template objects before activating the grid, you can align them to the active grid later on.

See also Aligning template objects to the grid (36)

Showing or hiding the grid (20)

Setting the grid density (20)

Setting the grid density

If you want to align objects to the grid points automatically as you draw or modify them, make sure that the grid is active.

You can set the distance between the grid points in graphical templates to aid drawing operations. Do the following:

1. Click **Options** > **Grid** > **Density**. This opens the **Grid Settings** dialog box.
2. Type in the distance between the grid points by defining the **Horizontal** and **Vertical** settings.
3. Click **OK** to save the changes.

See also Activating the grid (19)

Showing or hiding the grid (20)

Showing or hiding the grid

You can show or hide the grid while working on templates. The grid is useful for aligning objects as they are drawn.

To show or hide the template grid, right-click in an empty area in the **Workarea** and click **Grid** > **Visible** from the pop-up menu.

The check mark in front of the option indicates that the grid is visible.

Activating the grid does not automatically make it visible. Note that the grid can be visible without being active.

See also Activating the grid (19)

Aligning template objects to the grid (36)

2.4 Tips for the Template Editor interface

Here are some useful hints and tips that help you use the Template Editor user interface and its basic features more efficiently.

Click the links below to find out more:

Starting Template Editor with the command line (20)

Keyboard shortcuts (21)

Status bar (22)

Pop-up menus (22)

Setting the presentation file and pen settings (23)

Tooltips (22)

Starting Template Editor with the command line

This is aimed at administrator level users, who can start Template Editor through the command prompt. Use these command line options to start Template Editor.

All command line options are case insensitive, so uppercase letters are equal to lowercase letters.

Command line option	Description
-L <language id>	Allows you to select the language for the Template Editor user interface. <language id> is a three-letter language identifier. Available languages depend on your product. Please refer to your product documentation for information on available languages.
-i <file path>	Allows you to set the Template Editor initialization file that holds the information on where Template Editor searches for certain definition files. <file path> refers to an ASCII format initialization file named with the extension -ini. This is an example of such a command line option: `-i C:\My Documents\Product\Templates\mysettings.ini` Click **Options > Preferences** to modify the default file location preferences.
-w <work folder>	Allows you to set the work folder for Template Editor. This folder is, for example, is the base folder for file open and save operations.

Keyboard shortcuts

Keyboard shortcuts in Template Editor:

Shortcut	Action
Ctrl + N	Creates a new template file.
Ctrl + O	Opens a template file.
Ctrl + S	Saves a template file.
Ctrl + Q	Exits Template Editor.
Ctrl + Z	Undoes the previous operation.
Ctrl + Y	Redoes the previous operation.
Ctrl + X	Cuts the selected component or object.
Ctrl + C	Copies the selected component or object.
Ctrl + V	Pastes the selected component or object.
Delete	Deletes the selected component or object.
Ctrl + G	Groups selected individual objects into one group so that they are handled as one object.
Ctrl + U	Turns the selected group of objects into individual objects.

Shortcut	Action
Ctrl + Home	Moves the selected row component higher on the current hierarchy level.
Ctrl + End	Moves the selected row component lower on the current hierarchy level.
Ctrl + Page Up	Moves the selected row to a higher level in the hierarchy.
Ctrl + Page Down	Moves the selected row to a lower level in the hierarchy.
Ctrl + D	Makes a copy of the selected object. Modify the settings in the dialog box.
Ctrl + M	Moves the selected object a specified distance in the template. Modify the settings in the dialog box.
Esc	Ends the current drawing action and enables selection of objects.
Alt + Enter	Opens the property dialog box for the selected item, where you can modify the properties and settings.

Pop-up menus

You can open context sensitive pop-up menus in the main window by right-clicking the mouse. The commands vary depending on which area in the Template Editor main window you click.

The **Content Browser** pop-up menu commands contain the most common modifying operations and allow you to change the location or hierarchy of row components, customize the view in the browser or access properties of the selected part. Some of the commands are also available in the **Sort and Drawing** order toolbar.

The **Workarea** pop-up menu commands contain the most common modifying operations, zooming, panning and setting Template Editor drawing options. You can also open the properties dialog box of the selected template, component or object.

See also Setting the sort order of a value field (53)

Template windows (15)

Template properties (26)

Status bar

The status bar shows information on the selected tool or command and is located on the bottom left side of the main window.

See also Drawing with the numeric pad (38)

Tooltips

When you rest the mouse pointer on a toolbar button, a tooltip appears.

Template Editor only has basic tooltips that display the name of a command.

See also Template Editor toolbars (13)

Setting the presentation file and pen settings

If your product does not use presentations, you can skip the following instructions.

A presentation file defines a set of pens that you can use when drawing. The presentation uses a color map file that defines a set of product-specific colors. You can use a presentation in Template Editor only if your product uses them. The pens and colors of the presentation file replace the Template Editor color, line type and line width. This means you specify the color, line type and line width for a Template Editor part that has these properties by selecting a pen.

Presentation files are created and modified with **Presentation Editor (PresEd)**.

Do the following to define the default presentation file location in Template Editor

1. Click **Options** > **Preferences** and click the **File Location** preferences tab.
2. Click in the **Presentation** line and then click **Select File**.
3. Select the presentation file to use and click **OK**.
4. Click in the **Color Map** line and then **Select File**.
5. Select the color map file to use and click **OK**.
6. Click **OK** to apply the changes and close the **Preferences** dialog box.

The pens of the presentation appear in the template object property dialog boxes and can customize the colors in the Template Editor **Workarea**.

See also File Location preferences (18)

3 Introduction to templates

This is an overview of templates. Here you will find out more about textual and graphical templates, how to use data from your product in templates; and template parts, which you need to be familiar with when you design templates.

See also Creating and saving a template (25)

Opening and closing templates (26)

Template properties (26)

Modifying template page properties (26)

Changing the template type (27)

3.1 Creating and saving a template

Before you start, we recommend you define various Template Editor preferences including the appearance of the **Workarea**, the graphical units (Metric or Imperial) and the references to data files.

1. Click **File** > **New**
2. Select the template type: graphical or textual.
3. Click **OK**.

   An empty template opens in the **Workarea** and you are ready to begin working with the template.
4. Click **File** > **Save As** to save the template.

   The standard file selection dialog box will appear for you to specify the name, save location and file extension. **Template Editor** automatically adds the selected extension to the file name.
5. Click **OK**.

You can begin working with the template and add components and objects.

See also Modifying general preferences (17)

Modifying Workarea preferences (17)

File Location preferences (18)

Graphical templates (10)

Textual templates (10)

Opening and closing templates (26)

3.2 Opening and closing templates

When you open an existing textual template, Template Editor checks for overlapping objects. If you receive a message about overlapping template objects, you should reposition or resize the objects to eliminate overlapping.

If you receive a message regarding template conversion when you open a template, contact your service representative for assistance if you do not know how to proceed.

To open an existing template file:

1. Click **File** > **Open**.
2. When the file selection dialog box opens, click the location you want to open the file from and click **OK**.

   If you have recently worked on a template, you can open it by clicking **File** > **Recent Files**, which lists the ten most recent template files you have worked on.
3. To close a template file, click **File** > **Close**.

   If you have many files open and want close them without exiting Template Editor, click **File** > **Close All**.

   If there are unsaved changes in any of the template files that you close, Template Editor asks if you want to save them. Select the option that best suits your situation.

See also Creating and saving a template (25)

3.3 Template properties

Properties determine things such as the size, color, font and output options of a template, component or object. You can modify properties through specific property dialog boxes.

Click the links below to find out more:

Modifying template page properties (26)

Changing the template type (27)

3.4 Modifying template page properties

Both graphical and textual templates require page and margin settings. The units used in the dimensions of a template are based on the template type. Graphical templates use graphical units (Metric or Imperial) and textual templates use character units.

For a graphical template meant to be a map legend or label, these settings specify the total width and height of it.

To set and modify the template page settings and margin settings:

1. Double-click the template to open the **Template Page Properties** dialog box.
2. In the **Output**, **Workarea** and **Margins** areas, modify the following fields:
   - **Width** and **Height**: set in graphical units or characters.
   - **View height**: sets the page height for template design, but does not affect template output.
   - **Margins**: sets the margins for the left, right and bottom of the template in graphical units or characters.

3. Click **OK** to save the changes.

You can also add notes about the template and assign free attributes to it.

See also

Modifying column settings (45)

Free attributes (78)

Adding free attributes (78)

3.5 Changing the template type

Change the template type when you need to use certain types of content in the template, for example, if you need to use Metric or Imperial measures in your reports. Changing the template type is also useful if you need to transfer the template output to Excel or some other editor.
To change the template type:

1. Click **File** > **Template** > **Type**. A dialog box opens.
   Select the template type.
2. Click **OK**.
3. Be sure to use **Save As** to save the template with the new (and correct) file extension (**.rpt** for textual templates and **.tpl** for graphical templates).

If you turn a graphical template into a textual template, only the text and value fields remain. All graphical objects will be deleted.

See also

Graphical templates (10)

Textual templates (10)

4 Template components and objects

Templates are made up of components and objects. Template layout is designed with template components. Template objects are inserted into template components.

Here you will find out more about common operations applied to template components and objects.

See also

Inserting components and objects (29)

Selecting components and objects (32)

Cutting, copying and pasting components and objects (32)

Modifying component and object properties (33)

Tips for component and object tasks (37)

4.1 Inserting components and objects

Inserting template components and objects is accomplished through the toolbars. Insert template components before inserting template objects. A template can have only one header and one footer component. Other components may be repeated.

To insert a template component:

1. Click the component you want to insert.

   Template Editor inserts components in the following order: header, page header, row, page footer, footer.
2. Modify the properties of the component by selecting and double-clicking it.
3. Click the object you want to insert and then click in the component to position it.

   The object is added to the template with the current default properties, which you can modify.

   The output of the final template is determined by the output settings.

Rows and value fields are used widely and offer more functionality in terms of gathering and outputting information.

If you double-click a button in the **Object** toolbar, you can draw several objects without clicking the button again. Press **Esc** or click another button to stop drawing.

Hold down the **Shift** key while moving an object to lock the movement horizontally or vertically.

See also Template rows (41)

Value fields (51)

Creating rows (41)

Inserting a page header or page footer (31)

Tips for component and object tasks (37)

Creating a value field (51)

Inserting a text object (30)

Drawing a polyline (30)

Modifying component and object properties (33)

Drawing a polyline

To draw a polyline:

1. Click the **Polyline** icon.
2. Click to define the starting point.
3. Move the mouse and click to define the next vertex point.
4. Repeat the previous step to add more points.
5. Double-click where you want to end drawing.

If you draw an unclosed object and later decide to fill it, Template Editor will automatically insert a line to close the shape.

See also Modifying component and object properties (33)

Inserting a text object

You may want to adjust the font settings before you begin, but you can always change them later. To insert text into a template:

1. Click the **Text** icon.
   A dialog box appears prompting you to type in the text.
2. Type the text and click **OK**.
   You can also insert a text object with the default string, simply click **OK**. The default string is defined with the check box **Set as default for new texts** in the **Text Properties** dialog box.
3. Click to define the insertion point of the text.

Double-click the text to modify the text properties and font settings.

See also Setting default properties for new template objects (35)

Font properties for objects (39)

Inserting a graphical field

Insert a graphical field to a template component when you want to reserve the area to include a logo or a graphic.

If the field is placed inside a row component, note that the row content type may block some attributes from use.

To insert a graphical field:

1. Click **Insert** > **Graphical Field** or click the **Graphical field** button.
2. Specify the insertion point:
   - Click twice to define the opposing corners **or**
   - Drag and release to end drawing.
3. Open the **Graphical Field Properties** dialog box.
4. In the **Attribute** list, select a global attribute to determine how the graphical field is used when the template is output,

   You can also click the **Attribute** button to select an attribute from the attribute tree.
5. Click **OK** to accept the changes and close the property dialog box.

See also Modifying the size of a template component or object (34)

Modifying the name of a template component or object (33)

4.2 Inserting a page header or page footer

Use page headers and page footers in a template if you want to print different headers or footers on specified pages during output. Do the following:

1. Click the page header or page footer icon.

   The page header appears below the header, or if no header exists, at the top of the template page.

   The page footer appears above the footer, or if no footer exists, at the bottom of the template page.
2. Set the name, height and output policy of the page header / footer in the properties dialog. The output policy options include:
   - **All** - component prints on every page.
   - **Even** - component prints on even pages.
   - **Odd** - component prints on odd pages.
   - **Exactly on** - component prints only on the page you specify.
   - **Not on** - component does not print on the page you specify.
   - **Last** - component prints on the last page.
   - **Not last** - component does not print on the last page.
   - **From** - component printing starts on the page you specify.
   - **Until** - component prints only up to the page you specify.
   - **Between** - component prints pages between the numbers you specify.

3. Insert template objects inside the component.

The output of the final template is determined by the output policy.

See also Modifying the name of a template component or object (33)

Modifying the size of a template component or object (34)

4.3 Selecting components and objects

In Template Editor you can specify how area selection works. You can choose to select all template components or objects that are either completely or partially inside the drag frame, or just the ones that are completely inside.

To select components and objects:

1. Start with setting the selection properties. Click **Options** > **Pick** > **Partially**.

   The check mark next to the option indicates that the option is on, and parts that fall only partially inside the drag frame are selected.
2. You can select multiple template components or objects or only one. Drag to make an area selection in the **Workarea**.
3. Hold down the **Ctrl** key to select multiple objects (you can do this in the **Content Browser** too).
4. To deselect all components or objects, click in an empty space in the **Workarea**. To deselect one component or object press the **Ctrl** key and click the part to deselect.

Selected parts appear highlighted in the **Workarea** and also show on the status bar. Only the most recently selected part appears in the **Content Browser**.

If the **Pick Multiple** setting is deactivated, you can select only one part.

See also Special drawing operations (38)

4.4 Cutting, copying and pasting components and objects

You can cut or copy template objects and components in Template Editor and paste them to another position in the same template or in another open template.

To cut, copy and paste template components and objects, do any of the following:

To...	Do this...
Cut a component or object	Select it and press **Ctrl** + **X**.
Copy a component or object	Select it and press **Ctrl** + **C**.
Paste the contents of the cut buffer	Press **Ctrl** + **V** and click to give the insertion point in the **Workarea**.

See also Duplicating objects (33)

Duplicating objects

You can make identical duplicates of template objects without having to create a new object from scratch.

To make a duplicate of an existing object:

1. Select the template object and press **Ctrl +D**.
2. Give the offset for the duplicate object in the **Duplicate Object** dialog box:

   The offset is the distance from the current coordinate point in graphical units. Positive values given horizontally and vertically insert the object to the right and up.
3. Click **OK**.

See also Cutting, copying and pasting components and objects (32)

4.5 Modifying component and object properties

The properties of template components and objects determine things such as the size, color, font and the output options. You can modify these properties through part-specific property dialog boxes.

To open the property dialog box for a template component or object, select and double-click the template component or object in the **Workarea** or in the **Content Browser**

Template Editor opens a component- or object-specific property dialog box. The changes you make there update the component or object when you accept them by clicking **OK**.

See also Modifying the name of a template component or object (33)

Modifying the size of a template component or object (34)

Moving a template component (34)

Cropping template components (34)

Changing the template component type (35)

Grouping or ungrouping template objects (35)

Setting default properties for new template objects (35)

Rotating template objects (36)

Aligning template objects (36)

Aligning template objects to the grid (36)

Moving a template object (37)

Modifying the name of a template component or object

The name of a component or object is a critical identifier in Template Editor.

For example, the names of value fields are used when referencing them in formulas or rules. So if you modify the name of a value field after you have used it in formulas or rules, you must also change the name in those formulas and rules. Template Editor does not update formulas or rules with the new name.

To modify the name of a template component or object:

1. Select and double-click the component or object to open the properties dialog box.
2. In the **Name** field type in a new name for the component or object.
3. Click **OK**.

The name you give does not appear during output, but does appear in the **Content Browser**.

See also Modifying the size of a template component or object (34)

Modifying a value field formula (58)

Modifying the size of a template component or object

You can modify the size of a template component or object.

To modify the size of a template component or object:

1. Select the component or object and double-click to open the properties dialog box.
2. Depending on the template component or object modify the size, length or radius.

   For some template objects, such as imported pictures, you can retain the aspect ratio by clicking **Keep aspect ratio** in the properties dialog box.
3. Click **OK** to save the changes.

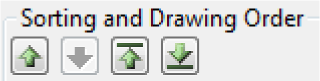
Template components and objects can also be reshaped (and resized) by dragging the resize handles. To do that select the component. Note that dragging the handles of polylines, polygons and free form areas is the only way to resize them.

See also Modifying the name of a template component or object (33)

Moving a template component

You can move a template component by changing its location in the **Content Browser**.

1. Select the template component.
2. Click the appropriate button in the sort and draw toolbar.



3. Save the template.

Moving a template component also affects the sort and draw order during the final output.

See also Row output properties (42)

Value field output properties (52)

Cropping template components

You can crop unneeded space from around template components.

To crop a component:

1. Select the component you want to crop.
2. Depending on which part of the component you want to crop, click **Edit** > **Crop** and the side of the component to crop.

Template Editor redraws the **Workarea** after you crop the template. Cropping cannot be undone once you save the template, the effects of which appear in the final output template.

See also Template components (10)

Changing the template component type

You can change the template component type without losing any of the information it contains.

Remember that you can only insert a header and a footer once in a template. In addition, if you change the template component to a row, you must define the content type for the row.

1. Select the component to crop.
2. Click **Edit** > **Change Type**. A dialog box opens prompting you to select a new template component type.
3. Select the template component type and click **OK**.

   Template Editor changes the component type in the **Workarea** and the **Content Browser**.
4. Save the changes.

See also Content types (42)

Grouping or ungrouping template objects

When you want to handle several objects as one object, you can create a group from existing template objects. When you need to resize or modify an individual object of the group you must ungroup the objects first.

To group and ungroup template objects:

1. Select the template objects you want to group or ungroup.
2. Right-click and select **Group** or **Ungroup**.

Double-click the group to open the **Object Properties** dialog box to modify the group. If you want to modify a single ungrouped item, double-click on it to open the properties dialog box.

Grouped objects have some limitations with regards to modifying them.

- You cannot rotate a group.
- If the group contains different types of objects, for example shapes and texts, you cannot modify its properties.
- If you modify the color properties of the group, the new color is applied to both lines and fills.

See also Inserting components and objects (29)

Modifying the size of a template component or object (34)

Moving a template object (37)

Setting default properties for new template objects

You can set the default properties for new template objects without having to define them every time a new, similar object is added to the template.

To set the default properties for new template objects:

1. Select an object that has the desired properties and object type.
2. Double-click to open the property dialog box for the object.
3. Click **Set as default for new** <template object>.

Note that if you modify other settings, the new ones are also applied to the currently selected object.

4. Click **OK** to save the changes.

Template Editor saves the settings and all newly created objects of the same type will have these properties.

See also Inserting components and objects (29)

Rotating template objects

You can rotate objects such as texts, symbols and value fields in graphical templates. **Note** that grouped objects cannot be rotated.

Rotate objects, for example, when they need to appear rotated in the final output template.

Do the following:

1. Select the object so you can see the resize handles.
2. Hold down the **Ctrl** key and start dragging on a handle.
   The handle where you start dragging is the anchor point of the object as you rotate it.
3. Move the mouse to rotate the object.
4. Release when the object is in its final position.

If you need to specify a specific value, you can also open the properties dialog box and specify the slant of the object in the **Slant** or **Angle** field.

See also Inserting a text object (30)

Adding symbols to a template (63)

Creating a value field (51)

Modifying component and object properties (33)

Aligning template objects

You can insert many objects into a template and align them to improve the layout. This is useful in big templates with lots of objects and quicker than selecting them and moving them one by one.

To align several objects so that they line up horizontally or vertically:

1. Select the objects that you want to align. Template Editor will align the objects according to the left most, right most, bottom or top object in the selection.
   If you hold down the **Shift** key while dragging, the object is moved only vertically or horizontally, whichever has bigger the offset at the time.
2. Click **Edit** > **Align** and select which sides of the objects you want to align.

Aligning template objects to the grid

The objects that you create when the grid is not activated can be moved on to grid points later on.

To align objects to the grid:

1. Activate the grid by clicking **Options** > **Grid** > **Snap**.
2. Select the object that you want to move.

3. Click **Edit > Align to Grid**.

This will change both the coordinates and size of the selected object(s) so they fit to the nearest grid points.

See also Setting the grid density (20)

Moving a template object

You can move an object in a template component by selecting it and dragging it to a new position. If you want to be more precise in moving an object do the following:

1. Select the object.
2. Hold down **Ctrl + M** and a dialog box opens where you can specify the offset values.
   - **Horizontal** is a distance relative to the original position. Positive values insert the object to the right.
   - **Vertical** is a distance relative to the original position. Positive values insert the object up.

   If you hold down the **Shift** key while moving an object, it locks the movement in the horizontal or vertical depending on which direction you choose.
3. To see the results click **Preview** to see the results. If the settings are not to your liking, you can readjust them and repeat the process.
4. Click **OK**.

To move an object to a different template component, to cut and paste it.

See also Offset (88)

Selecting components and objects (32)

4.6 Tips for component and object tasks

Here are some useful hints and tips that are exceptions to the normal use of components and objects.

See also Output options for headers and footers (37)

Special drawing operations (38)

Drawing with the numeric pad (38)

Drawing with the grid (39)

Viewing or hiding template objects (39)

Font properties for objects (39)

Searching for text (40)

Drawing a polyline (30)

Setting the presentation file and pen settings (23)

Template output options (82)

Output options for headers and footers

Template headers and footers have special output options. Select the header or footer and double-click to open the properties dialog box.

Option	Description
All	Prints on every page.
Even	Prints on even pages.
Odd	Prints on odd pages.
Exactly on	Prints only on page specified.
Not on	Does not print on page specified.
Last	Prints on the last page.
Not last	Does not print on the last page.
From	Prints starting on the page specified.
Until	Prints only to the page specified.
Between	Prints pages between the range you specify.

See also Row output properties (42)

Outputting a template (81)

Special drawing operations

There are several options you can set or turn on for customizing your drawing operations in Template Editor. They are described below:

Drawing option	Do this
To draw several objects running with the same tool	1. Click **Options** > **Continuous Action** to turn the option on. 2. Click the icon of the object you want to draw and proceed. You cannot use this option for symbols, pictures or inserted AutoCAD or MicroStation files.
Temporary continuous drawing	Double-click the toolbar icon of the object.
End continuous action	When continuous action is on, you have to end it: press the **Esc** button

See also Selecting components and objects (32)

Drawing with the numeric pad

You can define the length and angle of a line object with the numeric pad on your keyboard after you click to define the starting point. The **NumLock** must be on for this feature to work.

If you do not have a numeric pad, you can draw a line object with the mouse and modify the properties of the line with the **Object Properties** dialog box.

Do the following:

1. Turn the NumLock on by pressing the **NumLock** button.
2. Click the **Line** icon and define the starting point.

3. Using the numeric pad on your keyboard, enter the first number and a dialog box appears prompting you to enter the length of the line object.

   At the same time the line angle appears in the **Status Bar**.

4. Click **OK** to finish drawing the line.

If the grid is activated, the line will snap to the nearest grid point if you enter a length that places the end point in between grid points. For example, if your grid points are 5 mm apart from each other and you enter a length of 24, Template Editor will snap the line's end point to the nearest grid point. In other words, Template Editor rounds the line length up or down.

See also Inserting components and objects (29)

Drawing with the grid

Use the template grid for drawing and positioning template objects. When the grid is active, all clicks during drawing operations align to the nearest grid point as if the grid point had been clicked. The grid can be visible or hidden.

Note the following:

- The distance between grid points is template-specific.
- If you have drawn objects before activating the grid, you can align them to the active grid later on.
- Activating the grid does not automatically make it visible. The grid may also be visible without being active.

See also Setting the grid density (20)

Showing or hiding the grid (20)

Activating the grid (19)

Aligning template objects to the grid (36)

Viewing or hiding template objects

To speed up working with templates, you may want to view or hide template objects in the **Content Browser**. In a graphical template, for example, hide all line objects in order to discern value fields more easily.

To change the visibility of template objects click **View** > **Content Browser** and click the objects to view or hide.

Template objects are still visible in the template and preferences are saved across sessions.

See also Modifying interface components (15)

Hiding rows from output (45)

Hiding value fields from output (55)

Font properties for objects

You can modify the color, type and dimensions of text objects and value fields. The font selection consists of TrueType fonts and Tekla fonts.

In textual templates the font settings are used only in Template Editor, not in the output template.

Select and double-click to open the **Text Properties** or **Value Field Properties** dialog box. Then open the **Select Font** dialog box by clicking the **...** button in the **Font** area of the dialog box.

Property	Description
Font type	Set the font type and select from Tekla fonts (set in the **File Location** preferences) or TrueType fonts (only on the Windows platform). Template Editor auto-selects the font according to letters you type in the **Font** field.
Font style	Set the font style for TrueType fonts (regular, bold, italic, bold italic).
Size	Sets the height of the font in graphical units.
Ratio (w/h)	Sets the width to height ratio of the font.
Slant	Sets the slope of the font and applies only to Tekla fonts.
Color	Sets the font color. If you use presentations, the pen sets the color.
Sample	Preview of the font, the sample is updated as you modify the font settings.

See also Modifying component and object properties (33)

File Location preferences (18)

Modifying general preferences (17)

Setting the presentation file and pen settings (23)

Searching for text

You can search for text in the **Content Browser** with the keyboard shortcut **Ctrl + F**. The search is not case sensitive.

Template Editor searches through all components and objects in the template for the text you have specified. Click the **Find** button to continue through the **Content Browser**.

When a new match is found, the text is highlighted in the **Content Browser**. Use the scrolling arrows to find multiple instances of the same text.

5 Template rows

Template rows are versatile components that are used to create listings of objects that are in the product database. You can insert multiple rows in a template. With rows you can sort, prioritize, hide and filter the information you want to show in a template.

See also Creating rows (41)

Content types (42)

Row output properties (42)

Column settings (43)

Modifying the content type of a row (44)

Modifying the sort type of a row (44)

Modifying column settings (45)

Hiding rows from output (45)

Modifying row hierarchy (45)

Controlling row output with rules (46)

5.1 Creating rows

A template can contain several row components. The row component is most useful in templates that produce listings of objects in the product database.

To create a row do the following:

1. Click the **Row** icon.
2. The **Select Content Type** dialog box appears, prompting you to select a content type for the row.

   If you want to use object-specific attributes in the row, you must select a content type from the list.

   Depending on your product, you may be able to select several content types. Check one or more content types from the list and click **OK**.

   Being able to select more than one content type involves modifying the Template Editor **.ini** file.
3. You can also insert the row without selecting a content type, by just clicking **OK**. Note that if you do not select a content type, the value fields you insert in the row component will not be able to gather the data from the product database.

4. Right-click to open the **Row Properties** dialog box and modify it, for example, the row rule to apply to the row.

See also Content types (42)

Column settings (43)

Creating and saving a rule (47)

Creating a rule with Rule Wizard (47)

5.2 Content types

Content types are object types in the product database. Content types in template row definitions automatically filter out unwanted object types from the output template. The current version of Template Editor uses content type listings. The list of content types as well as their effect is product-specific.

When you create a new row in the template, you should select a content type for the row. The content type determines which template attributes can be used in that row.

See also Template rows (41)

Modifying the content type of a row (44)

5.3 Row output properties

Row properties determine the output preferences and appearance of row components. Select and double-click a row component to open the **Row Properties** dialog box, where you can modify the following properties of a row.

Property	Description
Content type	Set the content type of the row to determine the object being output from the product database.
Free attributes	Apply free attributes to the row when you want to specify product-specific information to a row.
Name	Modify the name of the row (to identify it easily).
Height	Modify the height of the row (to fit template objects more easily).
Sort type	Set the sort type of the row in order to determine the final output of rows.
Output	Hide or show the row in the final output.
Use columns	Use columns to make better use of the space in a template.
Rule	Set a rule on the row to define the information it shows during output.

See also Creating rows (41)

Content types (42)

Free attributes (78)

Modifying the content type of a row (44)

Modifying the sort type of a row (44)

Modifying column settings (45)

Hiding rows from output (45)

Creating and saving a rule (47)

5.4 Column settings

Templates can be printed in several columns (just like newspaper columns). Template Editor cannot visualize several columns, but the column settings determine the width of row components. These column settings are common to all rows in a template.

Field	Description
Min. count	Sets minimum number of columns in output template. Value cannot exceed **Max. count**.
Max. count	Sets maximum number of columns in output template. Value cannot be smaller than **Min. count**.
Spacing	Sets the gap between columns.
Fill direction	Fill direction determines in which direction the columns are filled: • **Horizontal**: fills the columns first to the right, then down. • **Vertical**: fills the columns first down, then to the right.
Fill policy	Fill policy determines how the columns are filled in the template: • **Even**: makes column heights equal. • **Continuous**: fills a column before proceeding to the next one, results in uneven column heights.

If you want to fix the number of columns, give the same values to **Min. count** and **Max. count**. Then the template is always output with this number of columns.

See also Modifying column settings (45)

5.5 Modifying the content type of a row

Modify the content type of a row when you want to filter the information that appears in a template. Before you start, ensure you have defined the row contents file location in the **File Location** preferences dialog box.

To modify the content type of a row:

1. Select and double-click the row to open the **Row Properties** dialog box.
2. Select the **Content type** from the list.
3. Click **OK** to save the changes.

See also Modifying the sort type of a row (44)

Free attributes (78)

File Location preferences (18)

5.6 Modifying the sort type of a row

Change the row sorting properties when you want to sort the output in a template based on rows. Sorting properties are determined by the **Sort type** of the row and how you set the sort order for value fields for the row in question.

The default sort type is to **combine** similar rows so that only the last one of them is output.

To modify the sort type of the row:

1. Select and double-click a row to open the **Row Properties** dialog box.
2. Choose the **Sort type** from the list:
   - **None** - output lines are not sorted.
   - **Combine** - output lines are sorted and duplicate output lines are combined so that only the last of them is output.
   - **Distinct** - output lines are sorted and all lines are output.
3. Click **OK** to save the changes.

If you click **Combine** or **Distinct**, you must also select which value fields affect the sorting and how.

If you click **Combine**, you can also automatically get calculated sums of the information on combined lines.

See also Duplicate lines (49)

Setting the sort order of a value field (53)

Setting the priority of a value field (53)

Modifying the content type of a row (44)

Hiding rows from output (45)

5.7 Modifying column settings

Modify the column settings of a template when you want to output rows in several columns. Templates can be printed in several columns (just like newspaper columns). This is especially useful if you have long lists that need to be divided into columns.

When you are designing the template, you cannot line up rows in columns. Columns will render when the template is output.

Template Editor cannot visualize several columns, but the column settings determine the width of row components. These column settings are common to all rows in a template. If the template is output to multiple columns, an individual row can still be printed using the whole width of the page.

To modify the column settings of rows:

1. Select and double-click a row to open the **Row Properties** dialog box.
2. Click the **Edit** button next to the **Use columns** check box.
3. Modify the column settings in the **Common Settings** dialog box.
4. Click **OK** to save the changes.

See also Column settings (43)

5.8 Hiding rows from output

Hide rows when you want to filter out information when the template is output.

To hide rows from output:

1. Select and double-click a row to open the **Row Properties** dialog box.
2. Click the **Hide in output** check box.
3. Click **OK** to save the changes.

Only the following things can exclude the row from output when the template output process is going on:

- The row content type does not match the object that is being processed. You can select the product object type that you want to handle in a row when you modify the content type of the row.
- Row rules are used to filter out the object that is being processed or prevent output of the row for another reason. You can set the conditions for output when you modify row rules.

See also Creating rows (41)

Content types (42)

Modifying the sort type of a row (44)

Modifying row hierarchy (45)

5.9 Modifying row hierarchy

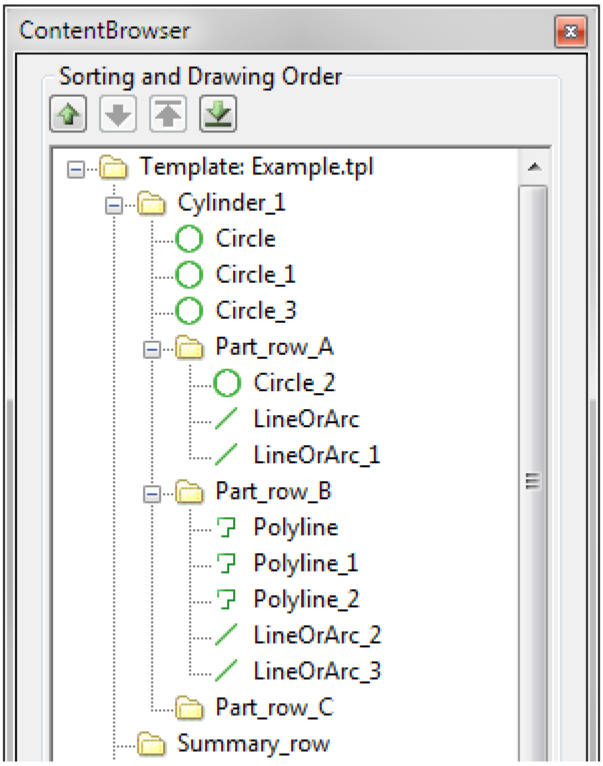
When you want to create hierarchical listings of objects in the product database, you can create a row hierarchy that resembles the hierarchy of those objects. This means that if a row describes the main object, the row(s) below it should describe a part of the main object.

To modify the row hierarchy:

1. Insert a new row, by clicking the row icon.
2. Insert other rows.

   In the **Sorting and drawing order** toolbar above the **Content Browser**, click the icon to shift the row down a level.
3. You can also drag the row to the new location in the **Content Browser**.
4. Save the changes by clicking the **Save** icon.

Note that you can see the changes in the hierarchy levels only in the **Content Browser**. There is no limit to the number of hierarchy levels to use, but hierarchies deeper than three levels affect the performance during output and may be difficult to work with.



See also Creating rows (41)

Hierarchical listings (84)

5.10 Controlling row output with rules

Row rules are conditional expressions that determine exceptions in the output of a row.

By default, all rows of a template are output for each processed database object. The easiest way to control output is to use a row content type; in which case the row is output only if the processed object type and content type match. If this is not a sufficient condition, then you need row rules, for example, in the following cases:

- You need to **step over** a row. For example, a summary row may be output only when all objects of a type have been processed and before starting to process different types of objects.
- You need to **step out** from processing a database object and continue with the next object. This means that processing continues from the first row of the template (or from the rows higher in the hierarchy).

Template Editor sorts all rows according to value field sort settings before the rules are read. Remember to set sorting order for all value fields that you use in row rules. This way rows will be correctly sorted. You can hide unnecessary value fields from the output. Do that in the **Value Field Properties** dialog box.

You can create row rules with the **Rule Wizard** dialog box or the **Rule Contents** dialog box.

See also

Content types (42)

Creating and saving a rule (47)

Creating a rule with Rule Wizard (47)

Conditional structure (88)

Modifying a row rule (48)

Pre-defined rules (48)

Creating and saving a rule

You can create more complex rules with the **Rule Contents** dialog box. Rules can act as filters, converters (change information to another format) or transporters (show information from elsewhere).

Create the row before you create the row rule.

To create and save a rule, do the following:

1. Double-click to open the **Row Properties** dialog box.

   You can define the content type of the row and modify other properties if you have not done that previously.
2. Click the **Advanced** button.

   The **Rule Contents** dialog box opens.
3. Inside the conditional statement you can use predefined attributes, already created value fields or combination of these with various data and string operations and formulas.
4. To save the rule for future use, click **Store**.
   - a Give the rule a name in the dialog box that appears.
   - b Click **OK** to save the changes.
5. Click **OK** to close the **Rule Contents** dialog box.
6. Save the template.

If you need to rewrite the rule or delete it completely, press **Clear** to delete the rule in the **Rule Contents** dialog box.

See also

Creating a rule with Rule Wizard (47)

Modifying a row rule (48)

Controlling row output with rules (46)

Pre-defined rules (48)

Creating a rule with Rule Wizard

You can create simple row rules with the **Rule Wizard**. You should have already created a row.

To create a rule with the **Rule Wizard**, do the following:

1. Select a template row and double-click to open the **Row Properties** dialog box.

   You can define the content type of the row and modify other properties if you have not done that previously.

2. Click the **Wizard** button.

   The **Rule Wizard** opens.

3. The **Rule Wizard** is used for filtering content in rows. The conditional structure is already a part of the rule in the **Rule Wizard**.
4. Select the **Attribute** to which the rule refers.
5. Add the conditions which apply to the rule by selecting from the list.
6. Type in the **Value**. If it is a string, enclose it in quotation marks "".
7. Click **OK** to save the changes.
8. Click **OK** to close the **Row Properties** dialog box.

If you want to save the rule for later use, click **Advanced** and then click **Store** in the **Rule Contents** dialog box.

See also Creating and saving a rule (47)

Controlling row output with rules (46)

Modifying a row rule

You can create, modify and check the content of row rules in the **Rule Contents** dialog box. If the row content type property is defined, it appears on the dialog box title bar.

To modify a template row rule:

1. Select and double-click a template row to open the **Row Properties** dialog box.
2. To modify a simple rule click **Wizard**. To modify a more advanced rule click **Advanced**.

   In both cases a dialog box opens, allowing you to make changes.

3. In the **Rule Contents** dialog box, click the **Check** button to ensure the syntax of the rule is correct.
4. Click **OK** to save the changes.
5. Save the template.

See also Creating and saving a rule (47)

Creating a rule with Rule Wizard (47)

Pre-defined rules (48)

Content types (42)

Pre-defined rules

You can save row rules for later use. There are two types of pre-defined rules - global rules and user rules.

Global pre-defined rules are read-only so they cannot be deleted from the menus in Template Editor. Global pre-defined rules are also stored in a different place than the user rules.

User rules are stored in a pre-defined rules file (extension **.pdr**). The file is specified in the **File Location** preferences. These rules can be freely added and deleted in the **Rule Contents** dialog box.

See also File Location preferences (18)

Creating and saving a rule (47)

Modifying a row rule (48)

Duplicate lines

Output lines are duplicates when they have identical values in all value fields that are used in sorting. Duplicates appear in template output when there are several similar objects in the product database. If you want to output only one line instead of outputting a line for each duplicate object, set the row sort type to COMBINE.

See also Modifying the sort type of a row (44)

6 Value fields

Value fields contain textual information. The field contents are references from the product database when the template is output. The data to retrieve is specified with value field formulas.

The size of the field is expressed in characters. In graphical templates the final size of the value field depends on the font size and type settings.

See also

6.1 Creating a value field

You can insert value fields in graphical templates and textual templates in order to gather and show data from the product database. For details on the attributes available in your product, please refer to your product documentation.

To create a new value field:

1. Click the **Value Field** icon.
2. Click to define the insertion point within a component.
3. The **Select Attribute** dialog box appears prompting you to select an attribute for the value field. Select an attribute in the list.

   If you want to insert the value field without defining an attribute, just click **OK**.
4. You can define a formula for the field by clicking the **Formula** button and specifying which data to gather.

   You can skip this step and leave the formula undefined, in which case Template Editor inserts `GetValue("ID")` by default.

5. Click **OK**.

The value field appears in the template component with a Template Editor-generated name.

If a value field formula consists of a single attribute function, Template Editor automatically sets the **Data type**, **Justify** (alignment), **Length**, **Meaning** and **Unit** properties. Depending on the **Unit** property, **Precision** or **Decimal** properties are also set. The values are read from the global attributes definition file.

See also Creating and saving a formula (56)

Global attributes (79)

Template definition files (12)

6.2 Value field output properties

Value fields can be modified based on what you want to calculate and show in a report. Open the **Value Field Properties** dialog box to modify the output properties.

Property	Description
Formula	The formula that has been applied to the value field. Modify the field by selecting an attribute and/or modifying the formula.
Name	A unique name for the value field. If you change the name of the value field, you must change it in all formulas and rules it appears in.
Data type	There are three options: • **Number** (an integer) • **Number with decimals** (set the number of decimals in the **Decimal** list) • **Text** (plain text)
Value field meaning	Defines the format for the value field and is product-specific. Format, for example, can be **Length**, **Weight**, **Force**, etc.
Unit	When the value field meaning is set, the related units are available in a list.
Layout properties	Modify the **Font**, **Length** of the field and the justification settings for the value field.
Field order	Value field order is determined by the sort type applied to the row in the **Row Properties** dialog box. Sort type must be defined in order to use this option.
Set as default for new value fields	Click the check box to save the settings applied to all new value fields.

See also Modifying a value field formula (58)

Modifying the name of a template component or object (33)

Setting the data type of a value field (54)

Assigning value field meanings (54)

Modifying the sort type of a row (44)

Setting default properties for new template objects (35)

Value field meanings

Value field meanings define output formatting for value fields.

The product applies formatting when the template is output.

For example, a value field meaning **Weight** might add a unit postfix such as **kg** or **lbs** to the value field value.

The value field meaning file is set in Template Editor **File Location** preferences.

See also Assigning value field meanings (54)

6.3 Setting the sort order of a value field

You can use a value field in row sorting. Set the sorting order of a value field when you want to output different values of the value field in different rows. Template Editor always sorts all rows based on value field settings before row rules are processed.

To set the sort order:

1. Select and double-click the value field to open the **Value Field Properties** dialog box.
2. Select the **Order** in the list:
   - **None**
   - **Ascending** (A to Z)
   - **Descending** (Z to A)
   - **Application** (determined by the product)
3. Click **OK** to save the changes.

You can determine the sort order only if the surrounding row component has **Sort type** defined. If you want to sort by multiple fields, you must set the sort order for each one of them and the sort priority for each field.

See also Setting the priority of a value field (53)

Setting the layout of value fields (55)

6.4 Setting the priority of a value field

Sort priority determines how several value fields affect the sorting of a row.

Do the following:

1. Open the **Content Browser** if it is not already visible.
2. Select the value field to set the priority for. Select from the fields that have sort order defined.
3. To increase the sort priority of the selected field, click the **Move up** button.

   To decrease the sort priority, click the **Move down** button.

4. Repeat steps 2 and 3 until the field positions are correct. The most important field in sorting should appear in the upper-most position.

You can also drag the value fields in the **Content Browser** to arrange them in the desired position. The move operations described here do not affect the physical coordinates of the value field, only the sorting properties.

6.5 Setting the data type of a value field

The data type of a value field determines what kind of data can be displayed in the field. Many product attributes have the data type defined already, so you may not even have modify the value field data type.

To set or modify it, however, do the following:

1. Select and double-click the value field to open the **Value Field Properties** dialog box.
2. Select the **Data type** in the list:
   - **Number** - an integer
   - **Number with decimals** - a decimal number, set the number of decimals in the **Decimal** field.
   - **Text** - plain text
3. Select the **Meaning** to apply from the list and then select the **Unit**.
   If the **Unit** applies the precision property, set it in the **Precision** list.
4. Click **OK**.

Be sure that the data type matches the outcome of the value field formula. For string operations, for example, the **Data type** should be set to **Text**.

See also Value field output properties (52)

Setting the priority of a value field (53)

Setting the layout of value fields (55)

6.6 Assigning value field meanings

Before you start, ensure that you have set the path for value field meanings in the **File Location** preferences dialog box.

Value field meanings are formatting for value fields applied at run time. The list of available meanings and their effect is product-specific.

To assign a meaning to a value field:

1. Select the value field you want to modify and double-click it to open the **Value Field Properties** dialog box.
2. Select an option from the **Meaning** list that best describes the meaning of the output of the value field.
   For example, a value field meaning for **Weight** might add a unit postfix such as kg or lbs.
3. Click **OK** to save the changes.

See also File Location preferences (18)

Setting the sort order of a value field (53)

Setting the priority of a value field (53)

Setting the data type of a value field (54)

Setting the layout of value fields (55)

6.7 Setting the layout of value fields

Setting the layout properties of a value field determines the appearance of the field contents during output.

To set the value field layout:

1. Select and double-click a value field to open the **Value Field Properties** dialog box.
2. In the **Layout** section, set the following
   - **Font** (style, size, color)
   - Justification (**Left**, **Right** or **Center**)
   - **Length** of the field in characters
   - The angle applied to the field
3. Click **OK** to save the changes.

See also Value field output properties (52)

6.8 Hiding value fields from output

There may be cases where you want to hide the contents of a value field from output.

To hide a value field from output:

1. Select and double-click the value field to open the **Value Field Properties** dialog box.
2. Select **Hide in output.**
3. Click **OK** to save the changes.

Even though the field is hidden from output, it can still be used in formulas and rules in the same way as visible fields.

See also Setting the sort order of a value field (53)

Setting the priority of a value field (53)

6.9 Formatting value fields

Before you start, ensure that you have set the path for value field meanings in the **File Location** preferences dialog box.

Format a value field when you want to show units and meanings in output.

To format a value field, do the following:

1. Select the value field and double-click it to open the **Value Field Properties** dialog box.
2. In the **Format** area of the dialog box, select a product-specific value field meaning in the **Meaning** list. Select the unit for the value field value in the **Unit** list.

   If decimals are used, set the number of decimals in the **Decimals** field. If precision is used, set it in the **Precision** list.

3. Click **OK** to save the changes.

See also File Location preferences (18)

Modifying File Location preferences (19)

Setting the data type of a value field (54)

Setting the layout of value fields (55)

6.10 Aligning value field texts in output

Before you start, ensure that you have set the path for attributes in the **File Location** preferences dialog box.

To align value field texts in a template:

1. Select the value field and double-click it to open the **Value Field Properties** dialog box.
2. Select the alignment option you want from the **Justify** list (**Left**, **Center** or **Right**).
3. Click **OK** to save the changes.

When you select the value field, you can see the justification settings in the **Preview Pane**.

See also Value fields (51)

Hiding value fields from output (55)

6.11 Value field formulas

Formulas are an essential part of a value field since they define what is gathered and shown in a value field. A formula can:

- get data from the product database
- produce summaries of data that is gathered with other value fields

Create formulas by applying various functions and operators in the **Formula Contents** dialog box.

See also Creating and saving a formula (56)

Modifying a value field formula (58)

Formula content properties (57)

Creating and saving a formula

Formulas determine what is data is gathered and shown in the value field. The data source can be your product database or other value fields in the current template, and the reference to the data source is made with Template Editor data functions. This is where you use the global attributes listing of your product.

Create a value field before you create a formula.

To create a formula, do the following:

1. Select the value field to modify and double-click it to open the **Value Field Properties** dialog box.
2. Click **Formula** to open the **Formula Contents** dialog box.
3. You can create, modify and check the content of value field formulas in this dialog box. You can select functions and parameters from the lists or type them in the **Formula** text field.
4. Click **Check** to ensure that the formula syntax is correct. Make corrections if necessary.

5. Click **OK** to save the formula and close the **Formula Contents** dialog box.
6. Set the output properties of the value field to match the outcome of the formula.
7. Click **OK** to accept the changes in the **Value Field Properties** dialog box.

If a value field formula consists of a single attribute function, Template Editor automatically sets the **Data type**, **Justify** (alignment), **Length**, **Meaning** and **Unit** properties. Depending on the **Unit** property, **Precision** or **Decimal** properties are also set. The values are read from the global attributes definition file.

See also

Attribute functions (94)

Control functions (91)

Mathematical operations (102)

Trigonometric functions (103)

Statistical functions (104)

Data type conversion functions (100)

String operations (98)

Miscellaneous functions (106)

Formula content properties

You can create, modify and check the content of value field formulas in the **Formula Contents** dialog box. You can select functions and parameters from the lists or type them in the **Formula** text field.

Property	Description
Math	Lists mathematical operations that you can apply.
String	Lists string operations that you can apply.
Attribute	Lists attribute functions for getting data from the product database.
Value field	Lists value field functions that can calculate sums or counts of value fields of this template.
Formula	Functions and parameters appear in this text field. The maximum length of this field is 2000 characters. Modify the formula by cutting, copying and pasting text in the field itself.

See also

Creating and saving a formula (56)

Adding comments to rules and formulas (88)

Modifying a value field formula (58)

Trigonometric functions (103)

Statistical functions (104)

Attribute functions (94)

Value field functions (95)

String operations (98)

Modifying a value field formula

The formula applied to a value field determines what data is gathered with the field. The data source can be your product database or other fields of the current template. The references to the data sources are created with **data functions**.

To modify a value field formula:

1. Select the value field and double-click it to open the **Value Field Properties** dialog box.
2. There are several ways to modify a value field formula:
   - Select a global attribute from the **Formula** list.
   - Click the **Attribute** button and select a global attribute from the attribute tree.
   - Click the **Formula** button. This opens the **Formula Contents** dialog box where you can modify the formula.

   Click **Check** to ensure that the formula syntax is correct. Make corrections if necessary.
3. Click **OK** to accept the changes and close the **Formula Contents** dialog box.
4. Set the output properties of the value field to match the outcome of the formula.
5. Click **OK** to accept the changes in the **Value Field Properties** dialog box.

If a value field formula consists of a single attribute function, Template Editor automatically sets the **Data type**, **Justify** (alignment), **Length**, **Meaning** and **Unit** properties. Depending on the **Unit** property, **Precision** or **Decimal** properties are also set. The values are read from the global attributes definition file.

See also

Get a single piece of data from the database (58)

Creating a listing of objects from the database (59)

Formula and rule reference (87)

Attribute functions (94)

Control functions (91)

Get a single piece of data from the database

Before you get started ensure you have defined the global attributes file path in the **File Location** preferences dialog box.

Use value fields to get data from your product's database. The available database objects and properties are listed in the global attributes definition file. Apply attribute functions in value fields.

Do the following:

1. Insert a value field inside any template component.
2. Select and double-click the value field to open the **Value Field Properties** dialog box.
3. Create a formula using one of the following attribute functions:
   - `GetValue`
   - `PreviousValue`
   - `NextValue`
4. Check the syntax of the formula and click **OK**.

5. Click **OK** to close the dialog box. The changes to the formula are saved.

See also Creating a value field (51)

Creating and saving a formula (56)

Creating a listing of objects from the database (59)

GetValue (94)

PreviousValue (95)

NextValue (94)

Creating a listing of objects from the database

Before you get started ensure you have defined the global attributes file path in the **File Location** preferences dialog box.

The listing of objects is created when you create a report based on this template in your product. The rule of the row in question has an effect on the final output of the listing.

To create a listing of objects from the product database:

1. Insert a value field inside a row component.
2. Select and double-click to open the **Value Field Properties** dialog box.
3. Create a value field formula using the `GetValue` attribute function.
4. Check the syntax of the formula and click **OK**.
5. Click **OK** to close the dialog box. The changes to the formula are saved.

See also Creating a value field (51)

Get a single piece of data from the database (58)

Modifying File Location preferences (19)

GetValue (94)

Creating a summary field

When you to want sum value field values in a summary field rather than outputting individual value fields on their own line, you can turn a value field into a summary field.

Do the following:

1. Make sure the **Sort type** of the row containing the value field is set to **Combine**. This ensures that you will be outputting only summary rows instead of repeating lines with information on individual database objects
2. Select the value field to sum and double-click to open the **Value Field Properties** dialog box.
3. You have these choices:
   - Click **Don't sum values** if you do not want any summaries made.
   - Click **Sum values across all rows** to sum all of the similar values of all rows.
   - Click **Sum values within one row** to add up all of the values of one row. All other rows are ignored.
4. Click **OK** to save the changes.

5. Then click **OK** to close the dialog box.

If you want to print the number of combined output lines, add a value field with the attribute **"NUMBER"**. If you do not want to combine lines but rather output each line and a summary row with an intermediate sum after them, we recommend that you use the **Sum** value field function to calculate the sum.

See also Modifying the sort type of a row (44)

Get a single piece of data from the database (58)

Creating a listing of objects from the database (59)

Value field functions (95)

6.12 Tips for value fields

Plan carefully what you want to output, and develop value field formulas accordingly. Here are some useful hints and tips that help you use value fields more efficiently.

See also Duplicate lines (49)

Offset (88)

Rotating template objects (36)

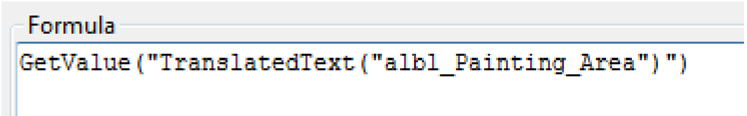
String translations of texts (60)

Setting Tekla Structures advanced options for value fields (60)

String translations of texts

You can use several languages in one template by using value fields to change the string language according to the language selection you set in Tekla Structures. This is useful, for example, if you work in a multi-language environment. This means you can create two different reports out of one template by switching the language in Tekla Structures.

The value field formula can be defined as follows:



Formula

```
GetValue("TranslatedText("albl_Painting_Area")")
```

Tekla Structures retrieves the selected language from the **.ail** file that contains the language strings. The result always shows in the native language and native units.

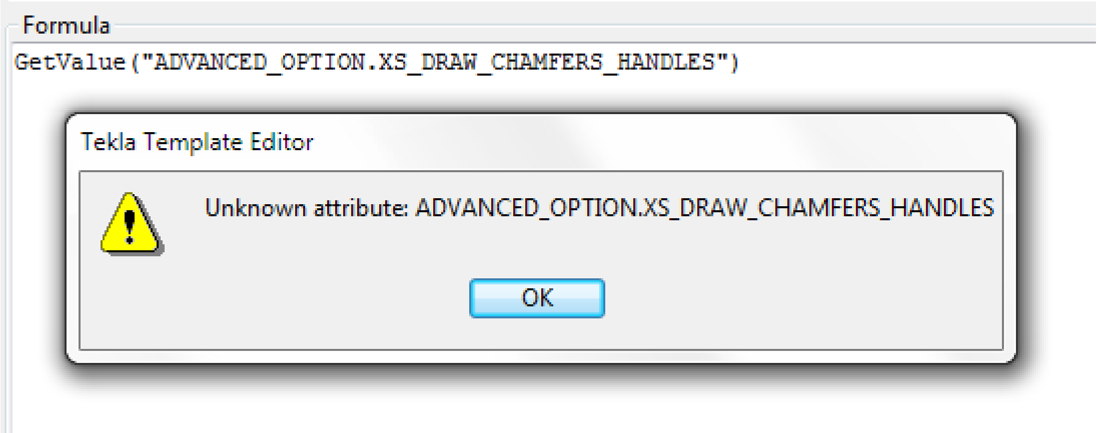
Setting Tekla Structures advanced options for value fields

You can use Tekla Structures advanced options in value field formulas.

1. Insert a value field into a template component.
2. Select the value field and double-click to open the **Value Field Properties** dialog box.
3. Click the **Formula** button to open the **Formula Contents** dialog box.
4. In the **Formula** field enter:

   `GetValue("ADVANCED_OPTION.XS_variablename")`
5. In the **Value Field Properties** dialog box, ensure that you select the right **Data type** for the value field in question: **Number, Number with decimals** or **Text**.

6. Click **Check** to see if the formula syntax is correct. If you see an error message, you can click **OK** to continue.



```
GetValue("ADVANCED_OPTION.XS_DRAW_CHAMFERS_HANDLES")
```

Unknown attribute: ADVANCED_OPTION.XS_DRAW_CHAMFERS_HANDLES

7. Click **OK** to close the **Formula Contents**. If another error message appears, you can ignore it.

The reason for the error indicating unknown attributes is because the advanced options variables are not listed in the `contentattributes_global.lst` or `contentattributes_userdefined.lst` files.

7 Symbols and pictures

Symbols are objects created with Symbol Editor and used in Template Editor and your product. Pictures are raster files.

Both symbols and pictures are inserted with a reference to the original file. If you modify the symbols and pictures outside of Template Editor, the changes also affect the template that contains this symbol or picture, as the file contents are reloaded each time the template is opened.

See also Symbol and picture libraries (63)

Adding symbols to a template (63)

Adding pictures to a template (64)

Adding a logo to a template (64)

Viewing and modifying symbol file contents (65)

Changing the symbol and the symbol file (65)

Tips for symbols and pictures (65)

7.1 Symbol and picture libraries

Symbols and pictures used in Template Editor are gathered in symbol files and directories.

Before you start inserting symbols or pictures into templates, you should ensure that the path for symbol files has been set correctly in the **File Location** preferences dialog box.

See also Modifying File Location preferences (19)

7.2 Adding symbols to a template

Ensure you have set the **File Location** preferences before you begin.

To insert a symbol into a template component:

1. Click the symbol icon.
2. Select the **Symbol library** where the symbol is located. Select the **Directory** and then select the **File**.
3. Click **OK**.

   The **Select Symbol** window opens.
4. Select the symbol and click **OK**.
5. Click the insertion point in the component and click the mouse to finish inserting the symbol.

6. Select the symbol and double-click it to modify the properties or select a different symbol.

See also Changing the symbol and the symbol file (65)

Viewing and modifying symbol file contents (65)

Modifying a symbol in Template Editor (66)

Setting paths for symbols and pictures (66)

7.3 Adding pictures to a template

Ensure you have set the **File Location** preferences before you begin.
To insert a picture:

1. Click the picture icon.
2. Select the directory where the picture is located. Select the **Directory** and then select the **File**.
3. Click **OK**.
4. Select the insertion point in the component.
5. Click and drag to define the area for the picture.
   When you release the mouse button the picture is inserted.

Select the picture in order to position it properly or right-click it to modify the properties or select a different picture file.

See also File Location preferences (18)

Modifying File Location preferences (19)

7.4 Adding a logo to a template

Templates can be customized by adding a logo to the final output. Before you start, ensure that you have set the right directory for the logo in the **File Location** preferences for symbols and pictures.

Logos (pictures) can be of the following file formats: BMP, GIF, GRD, JPG, PPM, PGM, PNG, RLE, TIFF or XKRL.

To add a logo to a template:

1. Click **Insert** > **Picture.**
2. The **Select Picture File** dialog box opens prompting you to select the picture directory where the logo resides.
3. Select the file to insert and click **OK**.
4. Specify the insertion point by doing one of the following:
   - Click twice to define the opposing corners.
   - Drag and release to end drawing.

The logo appears in the template and in the **Content Browser** as a **Picture**.

See also Symbols and pictures (63)

Symbol and picture libraries (63)

Setting paths for symbols and pictures (66)

7.5 Viewing and modifying symbol file contents

You can view the symbol file contents and modify the contents of a symbol file to suit your needs. To view or modify the contents of a symbol file in Tekla Structures:

1. Open the **Mark content – symbol** dialog box by selecting **Symbol** from the available mark elements list in the mark properties dialog box, or by opening the **Symbol properties** dialog box by selecting **Annotating** > **Properties** > **Symbol**.
2. Click **Select** next to the **File** box.
3. Select a file from the **Symbol Files** list and click **Edit**.

   This opens the selected symbol file in Symbol Editor.
4. If you modify the file in the Symbol Editor, save the file by clicking **File** > **Save** or **File** > **Save As** and giving the symbol file a new name.
5. Click **OK**.

In Symbol Editor, you can copy symbols between symbol files (*.sym). Press **Ctrl + C** and select the symbol you would like to copy, then open the symbol file you want to copy to (or a new symbol file), select the location for the symbol and press **Ctrl + V**.

See also Adding symbols to a template (63)

Setting paths for symbols and pictures (66)

Tips for symbols and pictures (65)

7.6 Changing the symbol and the symbol file

You can select a different symbol from the same symbol file or change the symbol file. Do the following:

1. Select a symbol and double-click to open the **Symbol Properties** dialog box.
2. Click the **...** button on the right of the **Identifier** box.

   The **Select Symbol** window opens.

   To change the symbol file, click the**...** button on the right of the **File** box.
3. Select a new symbol.

   Select a new symbol file and then select a new symbol.
4. Click **OK** to save the changes.

See also Adding symbols to a template (63)

7.7 Tips for symbols and pictures

These tips will help you to use some basic symbol and picture functionalities more efficiently.

Click the links below to find out more:

Modifying a symbol in Template Editor (66)

Setting paths for symbols and pictures (66)

Modifying a symbol in Template Editor

The modifications you make to a symbol in Template Editor do not affect the original symbol but allow you to use the same symbol in different colors, for example. The changes in line and fill properties only affect the transparent parts of the symbol.

If you want to modify the contents of symbol files, you must do it with Symbol Editor.

See also Adding symbols to a template (63)

Modifying component and object properties (33)

Setting paths for symbols and pictures

In order to access symbol and picture files in Template Editor, you must set the default directories first.

1. Click **Options** > **Preferences** and click the **File Locations** tab.
2. Click in the **Symbols and pictures (*)** line in the **Location** field and type in the directory paths.

   To give several paths, separate paths using a semi-colon. For example `"C:\pictures;C:\Product\symbols"`.
3. Click **OK** to save the changes.

See also Adding symbols to a template (63)

Adding pictures to a template (64)

Adding a logo to a template (64)

Viewing and modifying symbol file contents (65)

Changing the symbol and the symbol file (65)

8 AutoCAD and Microstation files

You can use AutoCAD (DXF or DWG) files and MicroStation (DGN) files in graphical templates. Template Editor imports the files and converts their contents to a group of shapes, so you can modify the imported data in Template Editor.

See also Importing AutoCAD and MicroStation files with import method defined (67)

Importing AutoCAD and MicroStation files with no import method defined (68)

Setting fill type and font import settings (69)

Limitations in AutoCAD and MicroStation file imports (70)

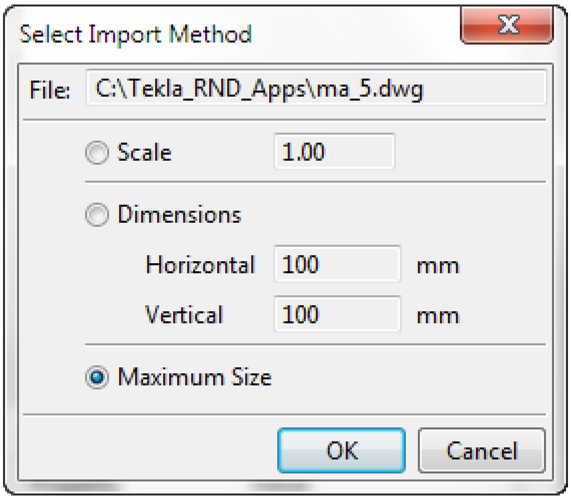
Grouping or ungrouping template objects (35)

8.1 Importing AutoCAD and MicroStation files with import method defined

Before you start, ensure that you have defined the import preferences for the size of AutoCAD and MicroStation files in the **General** preferences dialog box. If you want Template Editor to ask for the size when the file is being imported click **Always ask for import method**.

To import a file:

1. Click **Insert** > **File**.
   The **Import File** dialog box opens.
2. Select the file you want to import and click **OK**.
3. You are prompted to click the insertion point for the file. When you do that the **Select Import Method** dialog box opens.
4. Because you specified the import preferences, you can click in a template component and define the insertion point. You have three options for specifying the size:



Select Import Method

File: C:\Tekla_RND_Apps\ma_5.dwg

○ Scale 1.00

○ Dimensions

Horizontal 100 mm

Vertical 100 mm

● Maximum Size

- Import by **Scale** - Template Editor multiplies the dimensions of the original file and will resize the component if the scale results in an image size that is larger than the component.
- Import by specifying **Dimensions** - Type the values in the **Horizontal** and **Vertical** fields. Template Editor maintains the aspect ratio when the file is inserted.
- Import in the **Maximum Size** - Template Editor inserts the file in the maximum possible size without resizing the component.

5. Click **OK** to insert the file into the template component.
6. You can modify the file in needed, for example, by ungrouping it and modifying individual file objects.

If you have designed a template with AutoCAD, you can use free attributes in AutoCAD to mark value fields. Template Editor converts them automatically to value field objects as the file is imported. Template Editor uses the name and the value of the AutoCAD free attribute as the default contents for the value field. For example, a free attribute named **REVNO** with the value **xx** is converted to a value field with the formula **GetValue("REVNO")**.

See also Modifying general preferences (17)

Importing AutoCAD and MicroStation files with no import method defined (68)

Setting fill type and font import settings (69)

Limitations in AutoCAD and MicroStation file imports (70)

AutoCAD and Microstation import errors (70)

Grouping or ungrouping template objects (35)

8.2 Importing AutoCAD and MicroStation files with no import method defined

Before you start, ensure that you have defined the import preferences for the size of AutoCAD and MicroStation files in the **General** preferences dialog box. If you do not want Template Editor to ask for the size when the file is being imported leave **Always ask for import method** unchecked.

To import a file:

1. Click **Insert** > **File**.
2. Select the file you want to import and click **OK**.
3. Click in a template component and define the insertion point, then click and drag to define the two opposite corner points of the image.

The image is inserted in the template. You can ungroup it if you need to modify parts of the image.

See also

Modifying general preferences (17)

AutoCAD and Microstation files (67)

Importing AutoCAD and MicroStation files with import method defined (67)

Limitations in AutoCAD and MicroStation file imports (70)

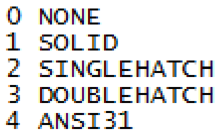
AutoCAD and Microstation import errors (70)

Grouping or ungrouping template objects (35)

8.3 Setting fill type and font import settings

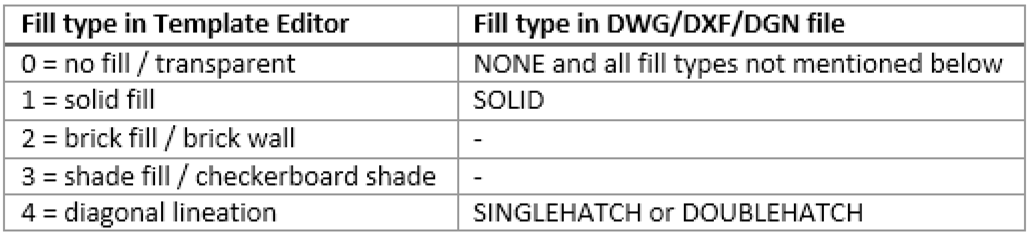
You can set the way **fill type** and **font** settings map from AutoCAD and MicroStation files to templates by modifying the fill type import and font map files. The fill type and font map paths are set in the **File Location** preferences dialog box.

The fill type import file is an ASCII file so it can be modified with any text editor. Each text line in the example below has a Template Editor fill type id and a DWG/DXF/DGN fill type name. This is a five-line example of a fill type mapping file.



```
0 NONE
1 SOLID
2 SINGLEHATCH
3 DOUBLEHATCH
4 ANSI31
```

The table below lists the available fill type ids and names that you can combine in the way you want. Each row describes the default mapping the Template Editor uses if the fill settings remained unchanged.



Fill type in Template Editor	Fill type in DWG/DXF/DGN file
0 = no fill / transparent	NONE and all fill types not mentioned below
1 = solid fill	SOLID
2 = brick fill / brick wall	-
3 = shade fill / checkerboard shade	-
4 = diagonal lineation	SINGLEHATCH or DOUBLEHATCH

See also

Modifying File Location preferences (19)

File Location preferences (18)

8.4 Limitations in AutoCAD and MicroStation file imports

Here are some of the limitations in AutoCAD and MicroStation file import operations.

Limitation	Description
Object type limitations	Only the following object types are imported from the original files: ARC, LINE, CIRCLE, CELL, CURVE, SHAPE, SOLID, TEXT, TRACE, INSERT, POLYLINE, LWPOLYLINE, LINESTRING, HATCH and FREE ATTRIBUTE (AutoCAD). See also Text limitations and Free attributes below.
Text limitations	The fonts used in text objects are mapped to Tekla fonts. This may slightly distort the font alignment within the text objects as the original font geometry is not available in Template Editor. Text objects that have with only white space characters (i.e tabs or spaces) are not imported.
Free attributes	If an AutoCAD free attribute has no name or value or if it consists of white space characters only, Template Editor does not import it. Free attributes with a valid name and value are converted to value fields.
Other limitations	You may experience problems with imported grouped objects and receive error messages during the import operation. User-defined line styles and world unit line widths from AutoCAD files are not imported.

See also AutoCAD and Microstation files (67)

Importing AutoCAD and MicroStation files with import method defined (67)

Importing AutoCAD and MicroStation files with no import method defined (68)

AutoCAD and Microstation import errors (70)

8.5 AutoCAD and Microstation import errors

If you get an error or warning message during the import operation it could be related to the grouped objects in the file:

- Unhandled object type: **"Unhandled object(s) of type XXXXX."**
- Elliptic arc approximated by circular arc: **"Elliptic arc(s) approximated by circular arc."**
- Approximating spline: **"Spline object(s) approximated with polyline through controlpoints."**

It may help to ungroup the object in the original file with a suitable editor and rearrange the objects.

Error messages are saved to the `import.log` file in the same folder as the *tpled.exe* application. If you need assistance with resolving file import errors, contact your service representative.

See also Importing AutoCAD and MicroStation files with import method defined (67)

Importing AutoCAD and MicroStation files with no import method defined (68)

Limitations in AutoCAD and MicroStation file imports (70)

Grouping or ungrouping template objects (35)

8.6 Modifying imported AutoCAD and MicroStation files

An imported AutoCAD or MicroStation file is a group of shapes.
To modify an imported file do the following:

1. Select the imported image, right-click and select **Properties** to open the **Object Properties** dialog box.

   To modify an individual shape of the group, you must first ungroup it.
2. Modify the properties of the group, or of an individual shape in the group.
3. Click **OK** to save the changes.

   Regroup the shapes when you are done modifying individual shapes.
4. Save the template.

See also Grouping or ungrouping template objects (35)

Importing AutoCAD and MicroStation files with import method defined (67)

Importing AutoCAD and MicroStation files with no import method defined (68)

Limitations in AutoCAD and MicroStation file imports (70)

9 Template attributes

Template attributes represent object properties. You can use template attributes in value field formulas and row rules to get the required data from the product database. At run-time, the product replaces the attribute with the actual value of the corresponding object property.

For example, if you include the attribute `WEIGHT` in a report template, Tekla Structures displays the weight of the model object in the report.

By default, these attribute files are located in `..\Program Files\Tekla Structures\<version>\nt\TplEd\settings`, but the location may be different in your environment.

Template attributes are defined in the following files:

File name	Description
`contentattributes.lst`	This is a container file listing all the files that contain the actual attribute definitions. The files are added with `INCLUDE` sentences. The order of the files included in `contentattributes.lst` defines the reading order of the files. This file is overwritten in the installation when you install a newer version of your product. **Ensure that you make a copy of this file before updating**. Generally, there is no need to modify `contentattributes.lst`. Do not modify it if you are not an administrator.
`contentattributes_global.lst`	This file contains attributes that are hard-coded into the program. **Do not edit this file.**

File name	Description
contentattributes_userdefined.lst	This file contains user-defined attributes, the same as in the objects.inp file. This file is overwritten in the installation when you install a newer version of your product. To use your own attributes in templates and reports, create a copy of this file and add the necessary attributes to that file.

See also See your product documentation for more information on product-specific attributes.

Value field formulas (56)

Controlling row output with rules (46)

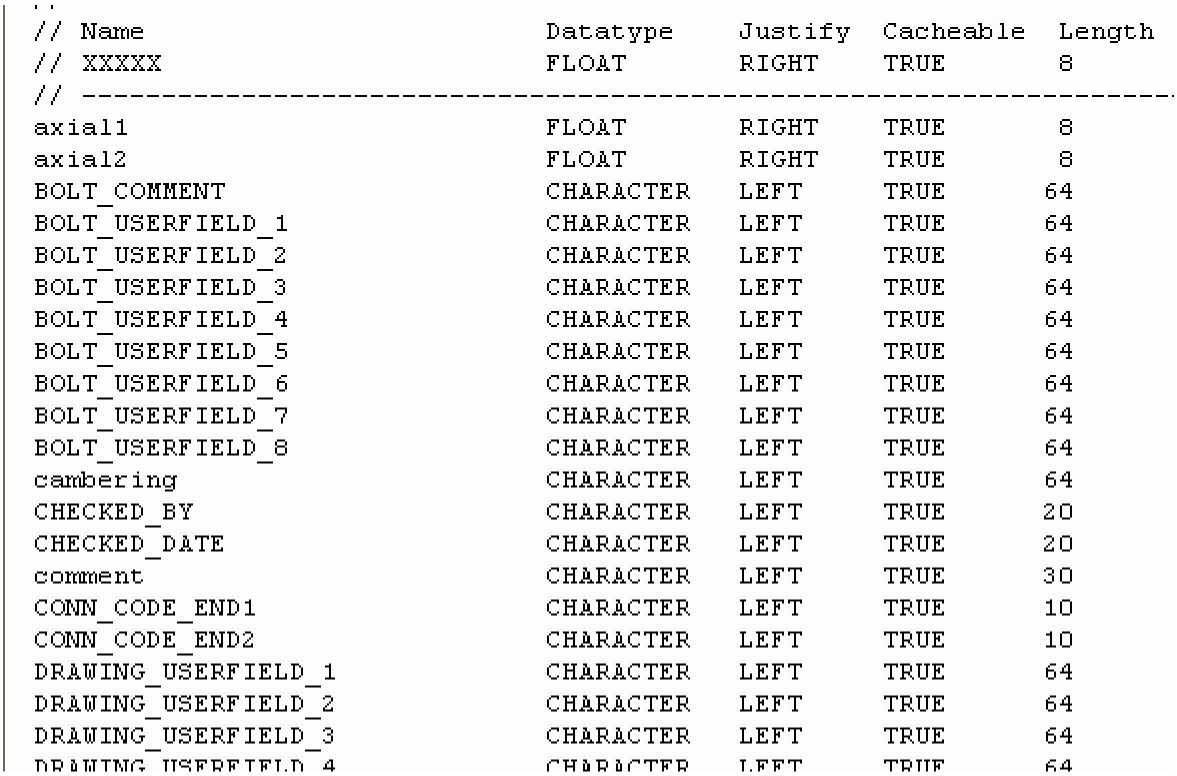
User-defined template attributes (74)

9.1 User-defined template attributes

User-defined template attributes are defined in the contentattributes_userdefined.lst file. By default, this file includes most of the user-defined attributes that are visible in the part properties dialog boxes. To use your own attributes in templates and reports, you should make a copy of the file, rename it appropriately, and add the necessary attributes to that file.

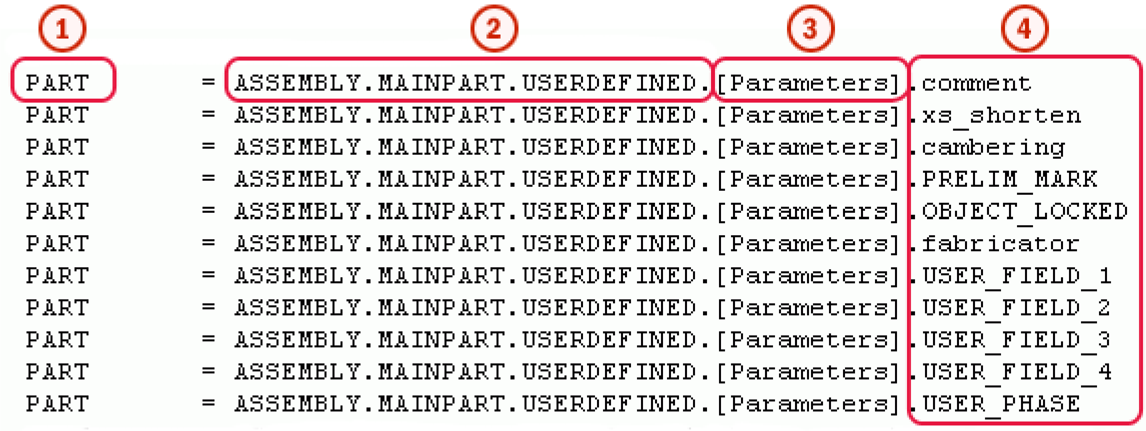
The contentattributes_userdefined.lst file is divided into two sections:

- A list of attributes and the default settings:



```
// Name                         Datatype    Justify  Cacheable  Length
// XXXXX                        FLOAT       RIGHT    TRUE       8
// --------------------------------------------------------------------
axial1                          FLOAT       RIGHT    TRUE       8
axial2                          FLOAT       RIGHT    TRUE       8
BOLT_COMMENT                    CHARACTER   LEFT     TRUE      64
BOLT_USERFIELD_1                CHARACTER   LEFT     TRUE      64
BOLT_USERFIELD_2                CHARACTER   LEFT     TRUE      64
BOLT_USERFIELD_3                CHARACTER   LEFT     TRUE      64
BOLT_USERFIELD_4                CHARACTER   LEFT     TRUE      64
BOLT_USERFIELD_5                CHARACTER   LEFT     TRUE      64
BOLT_USERFIELD_6                CHARACTER   LEFT     TRUE      64
BOLT_USERFIELD_7                CHARACTER   LEFT     TRUE      64
BOLT_USERFIELD_8                CHARACTER   LEFT     TRUE      64
cambering                       CHARACTER   LEFT     TRUE      64
CHECKED_BY                      CHARACTER   LEFT     TRUE      20
CHECKED_DATE                    CHARACTER   LEFT     TRUE      20
comment                         CHARACTER   LEFT     TRUE      30
CONN_CODE_END1                  CHARACTER   LEFT     TRUE      10
CONN_CODE_END2                  CHARACTER   LEFT     TRUE      10
DRAWING_USERFIELD_1             CHARACTER   LEFT     TRUE      64
DRAWING_USERFIELD_2             CHARACTER   LEFT     TRUE      64
DRAWING_USERFIELD_3             CHARACTER   LEFT     TRUE      64
DRAWING_USERFIELD_4             CHARACTER   LEFT     TRUE      64
```

- A list of attributes assigned to content types:



```
PART        = ASSEMBLY.MAINPART.USERDEFINED.[Parameters].comment
PART        = ASSEMBLY.MAINPART.USERDEFINED.[Parameters].xs_shorten
PART        = ASSEMBLY.MAINPART.USERDEFINED.[Parameters].cambering
PART        = ASSEMBLY.MAINPART.USERDEFINED.[Parameters].PRELIM_MARK
PART        = ASSEMBLY.MAINPART.USERDEFINED.[Parameters].OBJECT_LOCKED
PART        = ASSEMBLY.MAINPART.USERDEFINED.[Parameters].fabricator
PART        = ASSEMBLY.MAINPART.USERDEFINED.[Parameters].USER_FIELD_1
PART        = ASSEMBLY.MAINPART.USERDEFINED.[Parameters].USER_FIELD_2
PART        = ASSEMBLY.MAINPART.USERDEFINED.[Parameters].USER_FIELD_3
PART        = ASSEMBLY.MAINPART.USERDEFINED.[Parameters].USER_FIELD_4
PART        = ASSEMBLY.MAINPART.USERDEFINED.[Parameters].USER_PHASE
```

1. The content type of the row in Template Editor

2. The attribute hierarchy in Template Editor

3. Customizable comments, such as the tab name in the user-defined attributes dialog box

4. The name of the user-defined attribute, the same as in the `objects.inp` file.

See also Adding user-defined template attributes in Template Editor (75)

Adding comments to user-defined template attributes (76)

Adding hierarchy to user-defined template attributes (77)

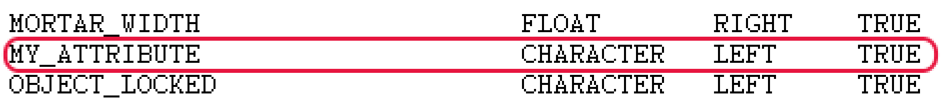
Adding user-defined template attributes in Template Editor

This example shows how to add your own user-defined attributes to the attribute tree in Template Editor. Before you start, add the user-defined attribute to the `objects.inp` file otherwise the attribute will not work. For example, you might add an attribute named `MY_ATTRIBUTE` to the user-defined properties for drawings.

To add user-defined attributes to the attribute tree:

1. Open the `contentattributes_userdefined.lst` file in a text editor.
2. Save the file with an appropriate name, for example `MY_contentattributes_userdefined.lst`, in the same folder.

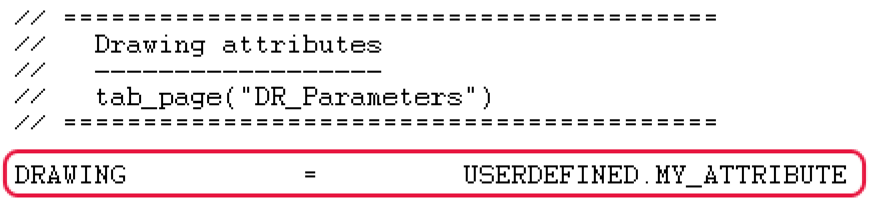
   Do not modify the `contentattributes_userdefined.lst` file.
3. Add `MY_ATTRIBUTE` to the list of attribute names and, define the settings as follows:



```
MORTAR_WIDTH                        FLOAT        RIGHT     TRUE
MY_ATTRIBUTE                        CHARACTER    LEFT      TRUE
OBJECT_LOCKED                       CHARACTER    LEFT      TRUE
```

4. Add `MY_ATTRIBUTE` to the list of attributes assigned to content types.
5. Select the content type according to which object the attribute is associated to in the `objects.inp` file. Add the attribute in the format `USERDEFINED.<ATTRIBUTE_NAME>`.

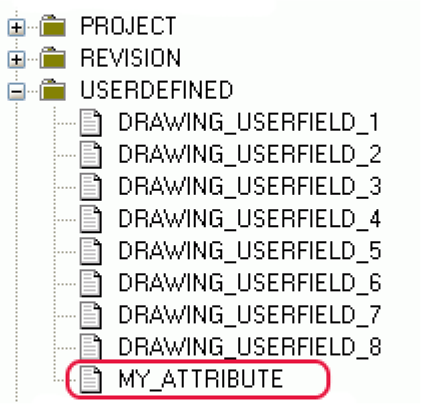
   In this example, the content type is `DRAWING`.



```
// ==========================================
//   Drawing attributes
//   ------------------
//   tab_page("DR_Parameters")
// ==========================================
DRAWING              =           USERDEFINED.MY_ATTRIBUTE
```

6. Save the changes.
7. Open the contentattributes.lst file.
8. Add the following line in the file: [INCLUDE MY_contentattributes_userdefined.lst].
9. Save the changes.

The attribute is shown in the attribute tree in Template Editor, under DRAWING > USERDEFINED:



```
PROJECT
REVISION
USERDEFINED
    DRAWING_USERFIELD_1
    DRAWING_USERFIELD_2
    DRAWING_USERFIELD_3
    DRAWING_USERFIELD_4
    DRAWING_USERFIELD_5
    DRAWING_USERFIELD_6
    DRAWING_USERFIELD_7
    DRAWING_USERFIELD_8
    MY_ATTRIBUTE
```

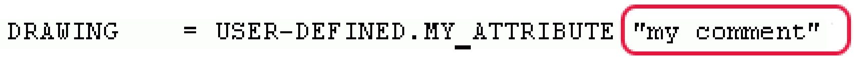
See also User-defined template attributes (74)

Adding comments to user-defined template attributes

You can add your own comments to the Template Editor attribute tree. Do the following:

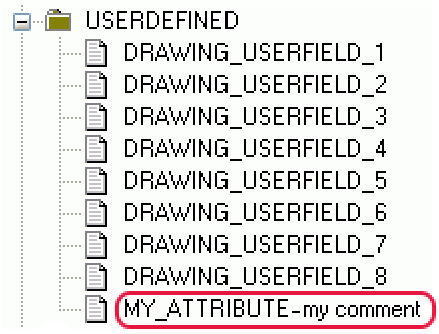
1. Open your copy of the contentattributes_userdefined.lst file. For example, MY_contentattributes_userdefined.lst.

   Do not modify the original contentattributes_userdefined.lst file.
2. Scroll down to the list of attributes assigned to content types.
3. Add your comment inside quotation marks, after the attribute name. For example:



```
DRAWING     = USER-DEFINED.MY_ATTRIBUTE "my comment"
```

4. The comment you added is displayed in the attribute tree in Template Editor.



```
USERDEFINED
    DRAWING_USERFIELD_1
    DRAWING_USERFIELD_2
    DRAWING_USERFIELD_3
    DRAWING_USERFIELD_4
    DRAWING_USERFIELD_5
    DRAWING_USERFIELD_6
    DRAWING_USERFIELD_7
    DRAWING_USERFIELD_8
    MY_ATTRIBUTE-my comment
```

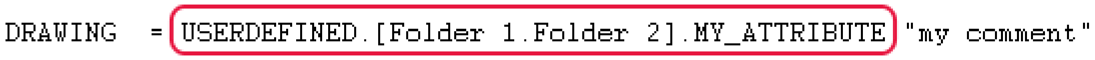
See also User-defined template attributes (74)

Adding hierarchy to user-defined template attributes

You can modify the hierarchy of the attributes in the Template Editor attribute tree.

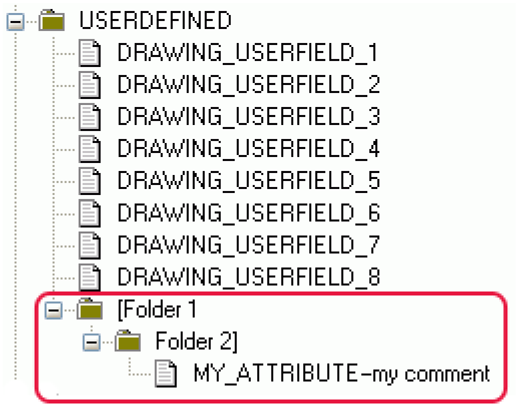
Do the following:

1. Open your copy of the `contentattributes_userdefined.lst` file. For example, `MY_contentattributes_userdefined.lst`.
   Do not modify the original `contentattributes_userdefined.lst` file.
2. Scroll down to the list of attributes assigned to content types.
3. Define the hierarchy in square brackets, between USERDEFINED. and the attribute name. For example:



```
DRAWING  = USERDEFINED.[Folder 1.Folder 2].MY_ATTRIBUTE "my comment"
```

   Notice the full stops after the brackets, and between the hierarchies.
4. Save the changes. The new hierarchy is shown in the attribute tree.



```
USERDEFINED
    DRAWING_USERFIELD_1
    DRAWING_USERFIELD_2
    DRAWING_USERFIELD_3
    DRAWING_USERFIELD_4
    DRAWING_USERFIELD_5
    DRAWING_USERFIELD_6
    DRAWING_USERFIELD_7
    DRAWING_USERFIELD_8
    [Folder 1
        Folder 2]
            MY_ATTRIBUTE-my comment
```

User-defined attributes are case sensitive. Ensure that you enter the attribute name using the correct case for all characters.

See also User-defined template attributes (74)

9.2 Free attributes

A free attribute provides additional product-specific information for a template, a row, a value field or a graphical field. It can, for example, define whether to output a scale with steel bar bending shapes or not. In Template Editor free attributes can either be user-defined (not necessarily product-specific) or application-defined (product-specific).

See also Adding free attributes (78)

Deleting free attributes (78)

Modifying free attributes (79)

Adding free attributes

You can add free attributes to a template, a row, a value field or a graphical field.
Do the following:

1. Select and double-click the template, row, value field or graphical field to open the properties dialog box.
2. Click the **Free attributes** button. The **Free Attributes** dialog box opens.
3. Add the attributes you want to apply to the selected element. You can add two types of attributes:
   - To add a free attributes defined by your product, work on the **Application** tab, which is active by default. Select an attribute name from the **Name** list.
   - To create a new user-defined free attribute, select the **User** tab, and type a name for the attribute in the **Name** field.
4. Depending on the attribute, you can select the value for the attribute from the **Value** list, or you may have to type a value in the **Value** field.
5. Click **Add** to add a new attribute. The attribute appears in the **Properties** table of the tab. You can add several attributes by repeating steps 3-5.
6. Click **OK** to save the changes, close the dialog box and return to the properties dialog box.

Once you click **OK** in the properties dialog box, Template Editor applies the free attributes you added to the selected template, row, value field or graphical field.

See also Modifying free attributes (79)

Deleting free attributes (78)

Deleting free attributes

To delete an existing attribute from a template, a row, a value field or a graphical field:

1. Select the template, row, value field or graphical field and double-click to open the properties dialog box.
2. Click the **Free attributes** button to open the **Free Attributes** dialog box.
3. Depending on the free attribute you want to delete, select the **User** tab or the **Application** tab.
4. Select an attribute from the **Properties** table.
5. Click **Delete**.

6. Click **OK**.

Once you click **OK** in the properties dialog box, Template Editor removes the deleted free attribute from the selected template, row, value field or graphical field.

See also Free attributes (78)

Adding free attributes (78)

Modifying free attributes (79)

Modifying free attributes

To modify the free attribute of a template, a row, a value field or a graphical field:

1. Select and double-click to open the properties dialog for the template, row, value field or graphical field.
2. Select the **User** tab or the **Application** tab.
3. Select an attribute from the **Properties** table.
4. Depending on the attribute type, either select a value from the **Value** list or type it into the **Value** field.
5. Click **Modify** to modify the existing attribute value. You can repeat steps 4-6 to make the needed changes.
6. Click **OK**.

Once you click **OK** in the properties dialog box, Template Editor updates the modified free attribute value to the selected template, row, value field or graphical field.

See also Deleting free attributes (78)

Adding free attributes (78)

9.3 Global attributes

A global attribute is a property of an object in the product database. Global attributes can be applied to rows, value fields and graphical fields. Database objects can be things like weld assembly types in Tekla Structures or the name of the engineer responsible for a project managed with Trimble products. It can also refer to graphical data such as a symbol or a picture in the product.

The global attributes definition file contains the list of available properties. Templates use field objects for getting the data. Value fields get textual data, and graphical fields are area reservations for product-specific graphics or texts. Each field refers to the product by using one or more global attributes.

Template Editor checks attribute names when formulas and rules are created and displayed to you as they are checked. If no global attributes file is specified, then this check is skipped.

The definition file for global attributes is set in Template Editor **File Location** preferences.

See also File Location preferences (18)

Template rows (41)

Template field objects (11)

Value fields (51)

10 Using templates in products

Here you will find out more about the general process in outputting a templates. There are some simple examples of templates as they are output.

Click the links below to find out more:

See also Outputting a template (81)

Template output options (82)

Material listings (82)

Hierarchical listings (84)

Labels (85)

Creating an HTML report (86)

10.1 Outputting a template

Outputting a template means using it in your product, for example, printing a report or adding project details to a plan. It consists of the following phases:

1. Based on the content types and the global attributes used in the template, the product prepares a list of identifiers. The list contains an id for all the objects that are to be processed. This phase does not appear to the user.
2. The header and the optional page header are output.
3. The first object in the identifier list is chosen for processing.
   - All rows in the template are output, and value field references to global attributes are resolved using the first object. Each row appears as a line in the output template.
   - If a row has a hierarchy the rows below it are processed before continuing to the next row.
   - If a new page is needed, page headers and page footers are printed according to their output settings.
4. Step 3 is repeated for all objects in the identifier list.
5. The optional page footer and the footer are output.

See also Output options for headers and footers (37)

Row output properties (42)

10.2 Template output options

You create templates with Template Editor and output them with your product. The output options of templates may vary between different products. Basically templates can be output on the screen, to a printer or to a file.

For more information on output options please refer to your product documentation.

Option	Output
Screen output	The screen output of a graphical template, for example, a map legend, is handled in the product as an imported picture. It can be placed on top of a background map or within a drawing. Textual templates appear as lines of text, usually in a separate window.
Printer output	Screen output can also be directed straight to a printer. Your product documentation may have different options for printer output as it is possible to save templates in product-specific formats.
File output	Screen output can also be saved to a file, for example an HTML or RTF file. The available file formats depend on the product.

See also Using templates in products (81)

Get a single piece of data from the database (58)

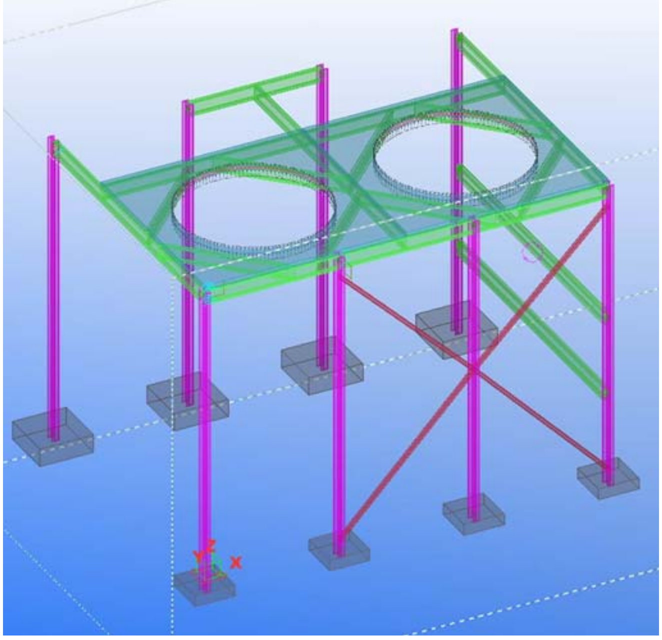
Creating a listing of objects from the database (59)

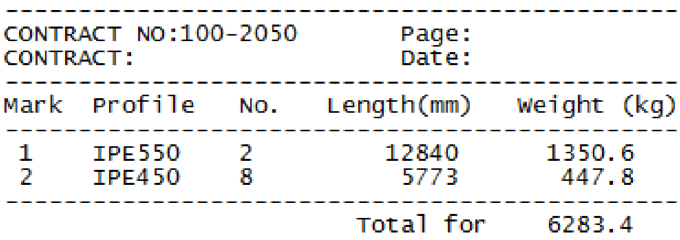
Hierarchical listings (84)

10.3 Material listings

The following illustrates a textual template that creates a listing of selected parts of a building structure.

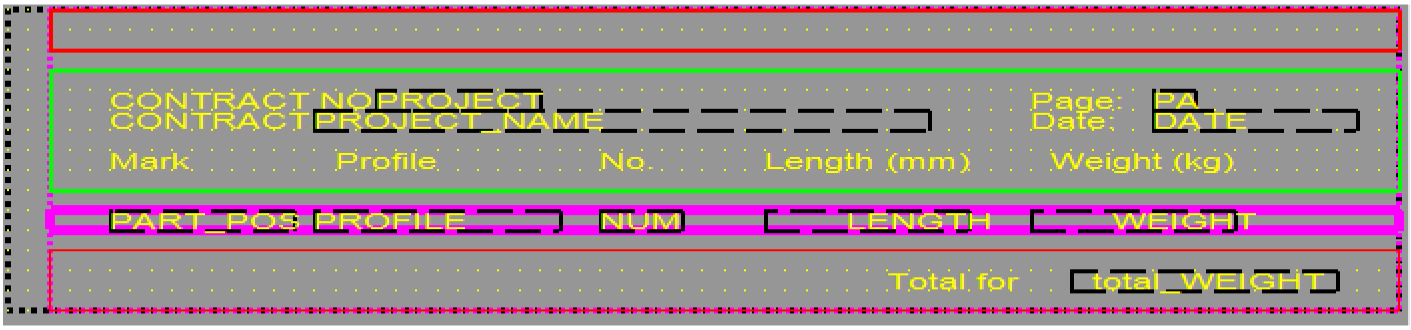
The image below shows a steel structure. A set of steel beams on the roof and along the wall are selected for processing. The selected parts appear in green. The listing contains the type, count, length and weight of selected beams, and the total weight of all the selected parts.





```
-----------------------------------------------
CONTRACT NO:100-2050        Page:
CONTRACT:                   Date:
-----------------------------------------------
Mark  Profile    No.   Length(mm)   Weight (kg)
-----------------------------------------------
 1    IPE550     2        12840        1350.6
 2    IPE450     8         5773         447.8
-----------------------------------------------
                       Total for     6283.4
```

The listing is based on the following template:



```
CONTRACT NOPROJECT                              Page: PA
CONTRACTPROJECT_NAME                            Date: DATE
Mark          Profile       No.     Length (mm)    Weight (kg)
PART_POS PROFILE           NUM      LENGTH          WEIGHT
                                           Total for   total_WEIGHT
```

The template has the following components

Part	Description
Header	The header at the top of the template prints the contract information and the title of the listing. It has the following template objects: dashed lines (text objects), texts ("Mark", "Profile", etc. (text objects) and data from the product: project number and name (value field objects).
Row	The row components define the listing of steel beams and has value fields that get the type, count, length and weight of the selected objects. Similar objects in the database are combined in the output by default, but if you need to separate those objects, they can be output as distinct rows.
Page footer	The page footer contains a value field producing the total sum of weights of all processed steel beams (value field) and the text "Total for" and dashed lines (text objects).

The references to the product data are made by using global attributes in value fields.

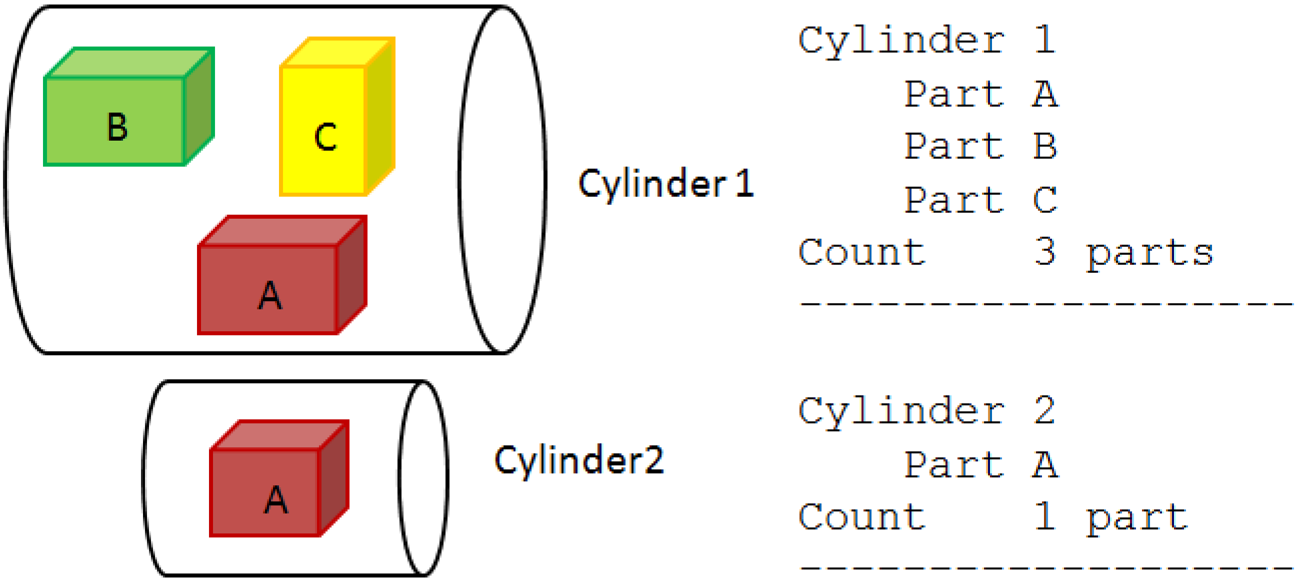
See also Global attributes (79)

Hierarchical listings (84)

Creating an HTML report (86)

10.4 Hierarchical listings

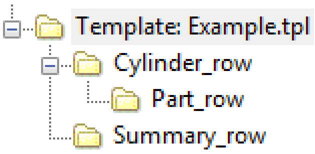
Use row hierarchy to create listings and display information in a logical and organized manner. The example below simulates objects in a product database. The listing describes each cylinder and its parts hierarchically.



```
Cylinder 1
     Part A
     Part B
     Part C
Count     3 parts
-------------------

Cylinder 2
     Part A
Count     1 part
-------------------
```

The listing is based on the following template:



Template: Example.tpl
Cylinder_row
Part_row
Summary_row

Part	Description
Cylinder_1 and **Cylinder_2**	Describes an object in the database. like an assembly. After this row is output, processing continues to the next parts in the hierarchy.
Part_rows	Part rows describe parts of the total object. The rows in the hierarchy are output so that all rows of the main object are processed. Processing continues to the next row.
Summary_row	Outputs the number of parts

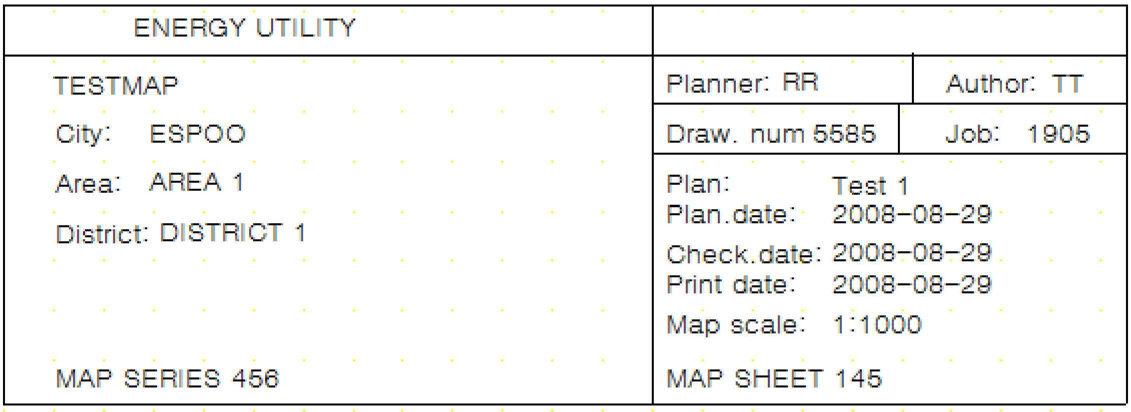
After outputting template rows for the first main object (Cylinder 1), the output process continues to the next main object (Cylinder 2) and to the first row of the template.

See also Creating a listing of objects from the database (59)

10.5 Labels

In addition to textual data, graphical templates may also contain graphics. This example template outputs a simple map legend. A label can be a template that contains only a header component.

The output template is a map legend that can be inserted on a background map. The legend, for example, contains the district name, scale and important dates.



ENERGY UTILITY

TESTMAP

City: ESPOO

Area: AREA 1

District: DISTRICT 1

MAP SERIES 456

Planner: RR

Author: TT

Draw. num 5585

Job: 1905

Plan: Test 1

Plan.date: 2008-08-29

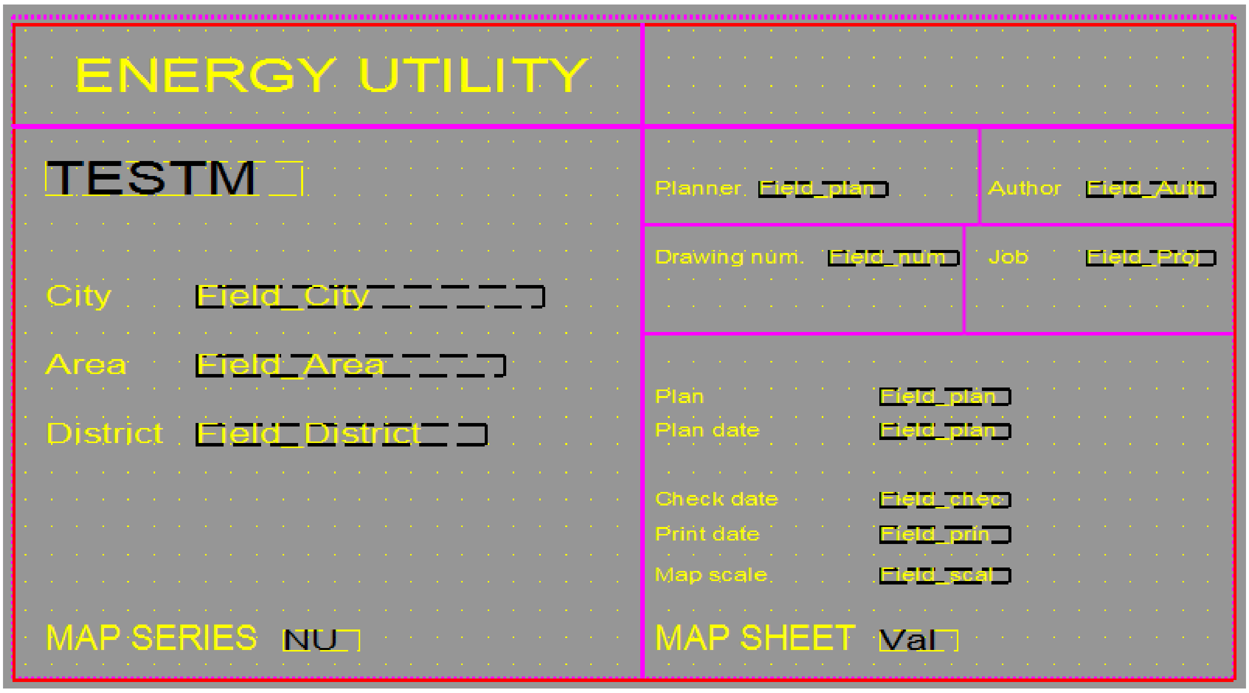
Check.date: 2008-08-29

Print date: 2008-08-29

Map scale: 1:1000

MAP SHEET 145

The legend is based on the following template:



ENERGY UTILITY

TESTM

City Field_City

Area Field_Area

District Field_District

MAP SERIES NU

Planner Field_plan

Author Field_Auth

Drawing num. Field_num

Job Field_Proj

Plan Field_plan

Plan date Field_plan

Check date Field_chec

Print date Field_prin

Map scale Field_scal

MAP SHEET Val

It is a header component that contains the following objects:

Object	Description
Text objects	Title texts "Energy Utility", "City", "Area" and "District"
Line objects	Table outlines

Object	Description
Value fields	Data from the product: name of the city and area, etc. References to the product data are made by using global attributes in value fields.

See also Global attributes (79)

10.6 Creating an HTML report

Templates in HTML format give you more possibilities for different layout, fonts, and images. Templates that generate output in HTML format are graphical and have the file name extension `.html.rpt`.

To create a template in HTML format:

1. Click **File** > **New**.
2. Select **Graphical** template and click **OK**.
3. Add rows in the template.
   - a Click **Insert** > **Component** > **Row** to add a new row.
   - b Select a content type for the row and click **OK**.
   - c Repeat steps a–b for each new row.
4. Add value fields to get the required data from your product database.
   - a Click **Insert** > **Value field**.
   - b Click a point to define the location of the field within the row.

     The **Select Attribute** dialog box appears prompting you to select an attribute for the value field.
   - c Select an attribute and click **OK**.
   - d Repeat steps a–c for each value field.
5. Add a header for each value field.
   - a Click **Insert** > **Component** > **Header**.
   - b Click **Insert** > **Text**.
   - c Enter a heading for the template, and click **OK**.
   - d Click a point to define the location of the heading in the header row.
   - e Repeat steps a–d to create headings for all the value fields.
6. Save the template.
   - a Click **File** > **Save as**
   - b Browse to the template folder where you want to save it.
   - c In the **File name** field, enter a name for the template.

     Include the extension `.html.rpt` in the file name. For example, `Part_list.html.rpt`.
   - d Click **OK**.

See also Outputting a template (81)

11 Formula and rule reference

Value field formulas and row rules are expressions that get data from the product. Formulas and rules use data functions to extract information and can contain mathematical and string operations.

It helps if you are familiar with standard programming language features such as data typing and conditional expressions. You can also use any ANSI C programming language guide to check the basic concepts.

See also

Value field formulas (56)

Controlling row output with rules (46)

Comparison and logical operators (89)

Control functions (91)

Attribute functions (94)

Value field functions (95)

Mathematical operations (102)

Arithmetic operators (102)

Trigonometric functions (103)

Statistical functions (104)

Data type conversion functions (100)

String operations (98)

Miscellaneous functions (106)

11.1 Tips for value field formulas and row rules

Plan carefully what you want to output and develop your row rules and value field formulas accordingly. Here are some tips that may help you.

See also

Adding comments to rules and formulas (88)

String translations of texts (60)

Offset (88)

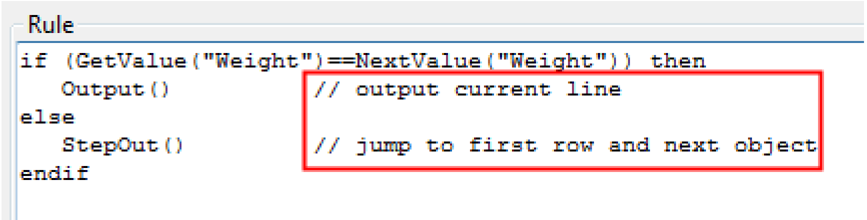
Adding comments to rules and formulas

You can add comments to row rules and value field formulas in order to add information that may be useful or helpful to you.

To add comments to rules and formulas:

1. Select the value field or row and double-click to open the **Value Field Properties** or the **Row Properties** dialog box.
2. Click the **Formula** button to open the **Formula Contents** dialog box or the **Advanced** button to open the **Rule Contents** dialog box.
3. Modify the contents of the formula or rule and add the comments at the end of each line with the following syntax: // <comment>

   You can also insert comments in-line with the following syntax: /* <comment> */



```
if (GetValue("Weight")==NextValue("Weight")) then
   Output()          // output current line
else
   StepOut()         // jump to first row and next object
endif
```

See also Creating and saving a formula (56)

Creating and saving a rule (47)

Modifying a row rule (48)

Modifying a value field formula (58)

Offset

Offset for graphical objects is the distance from the current coordinate point in graphical units, which are mm or inches in graphical templates and character units in textual templates.

Offset for formula string functions is the position of a character within a string. Positions are numbered from zero onwards. The first character in a string has an offset of 0, the second character in a string has an offset of 1, etc.

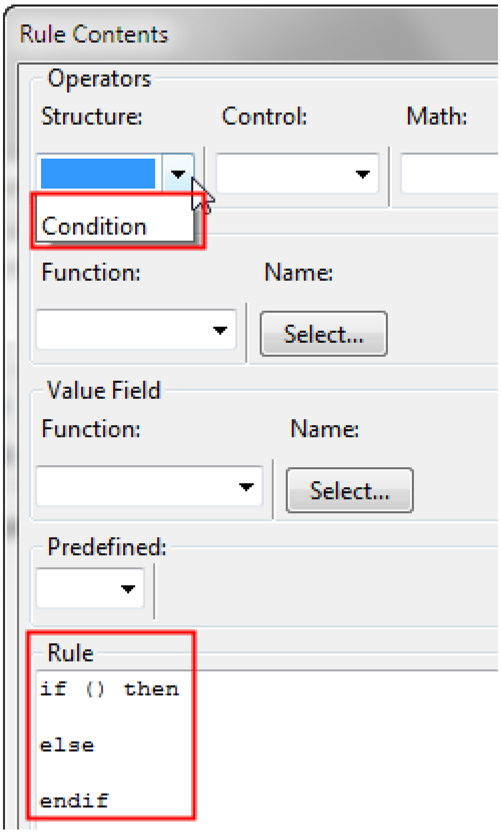
See also Duplicating objects (33)

Moving a template object (37)

11.2 Conditional structure

The conditional structure in Template Editor row rules is like the *if-else* statement in any programming language.

You can create rules by modifying the structure and conditions of the *if-else* statement.



```
if () then

else

endif
```

See also Controlling row output with rules (46)

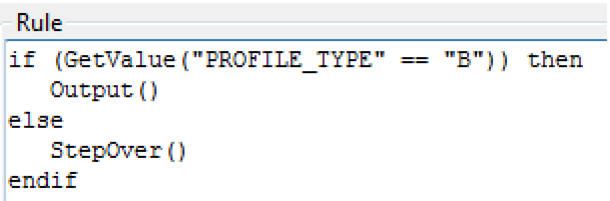
Creating and saving a rule (47)

Modifying a row rule (48)

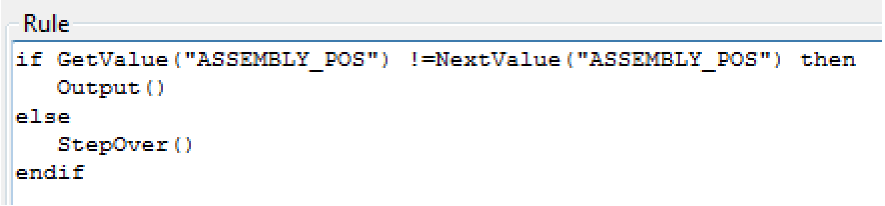
11.3 Comparison and logical operators

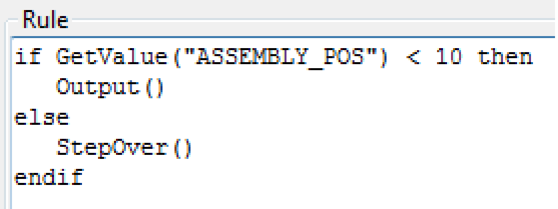
Comparison and logical operators appear in conditional structure to evaluate operands and combine conditions. You can type them in after placing the cursor in the desired position in the **Rule** or **Formula** text field.

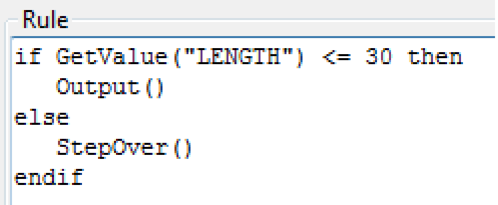
Operator	Description	Example

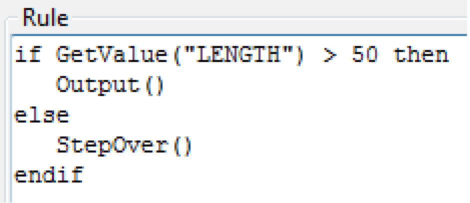
==	Both sides are equal	The following condition checks if the profile type is B.
		
		Rule
		`if (GetValue("PROFILE_TYPE" == "B")) then`
		`    Output()`
		`else`
		`    StepOver()`
		`endif`

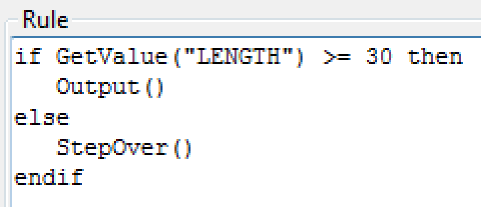
Operator	Description	Example

!=	Sides are not equal	This rule checks if the assembly position is not equal to the next value of assembly position. If this is true, the row is output. If they are equal, the condition is false, and printing the row is skipped.
		
		Rule
		`if GetValue("ASSEMBLY_POS") !=NextValue("ASSEMBLY_POS") then`
		`   Output()`
		`else`
		`   StepOver()`
		`endif`

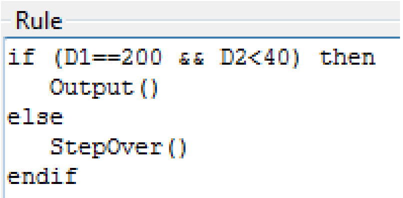
<	Left side is smaller	This rule finds all rows with the assembly position that is less than 10, anything equal to or greater than 10 is skipped.
		
		Rule
		`if GetValue("ASSEMBLY_POS") < 10 then`
		`   Output()`
		`else`
		`   StepOver()`
		`endif`

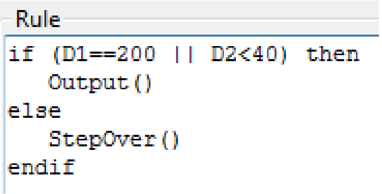
<=	Left side is smaller or equal	This rule outputs values with a LENGTH less than or equal to 30. If those values are not found, they are skipped.
		
		Rule
		`if GetValue("LENGTH") <= 30 then`
		`   Output()`
		`else`
		`   StepOver()`
		`endif`

>	Right side is smaller	This rule outputs values with a LENGTH greater than 50, anything equal to or less than 50 is skipped.
		
		Rule
		`if GetValue("LENGTH") > 50 then`
		`   Output()`
		`else`
		`   StepOver()`
		`endif`

>=	Right side is smaller or equal	This rule outputs values with a LENGTH equal to and greater than 30, anything less than 30 is skipped.
		
		Rule
		`if GetValue("LENGTH") >= 30 then`
		`   Output()`
		`else`
		`   StepOver()`
		`endif`

Operator	Description	Example

&&	Logical AND, both conditions must be true	If D1 is 200 and D2 smaller than 40, the row is output, otherwise it is skipped.
		
		Rule
		`if (D1==200 && D2<40) then`
		`    Output()`
		`else`
		`    StepOver()`
		`endif`

||	Logical OR, only one condition must be true	If D1 is 200 or D2 is smaller than 40, the row is output, otherwise it is skipped.
		
		Rule
		`if (D1==200 || D2<40) then`
		`    Output()`
		`else`
		`    StepOver()`
		`endif`

See also Conditional structure (88)

Value field functions (95)

Control functions (91)

11.4 Control functions

Use control functions within a conditional structure to inquire or define the output of a row and / or rows in the hierarchy. They have no parameters.

To understand how control functions work, you must be familiar with the template output process.

Access control functions in the **Rule Contents** dialog box. Select functions from the **Control** list in the **Operators** area of the dialog box.

See also IsFirst (92)

IsLast (92)

Output (92)

PageBreak (92)

StepIn (93)

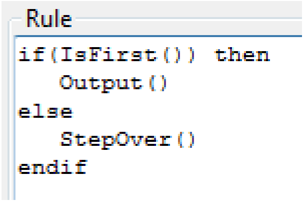
StepOut (93)

StepOver (93)

IsFirst

This function checks if the row is the first one, and returns TRUE if it is.

This rule outputs the first row only and skips the rest. For each row, it checks, if the row is the first one. If it is, the row is output, otherwise the row is stepped over.



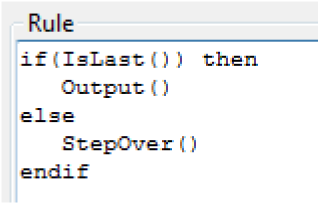
```
if(IsFirst()) then
   Output()
else
   StepOver()
endif
```

See also IsLast (92)

IsLast

This function checks if the row is the last one and returns TRUE if it is.

This rule outputs the last row only and skips the rest. For each row, it checks, if the row is the last one. If it is, the row is output, otherwise the row is stepped over.



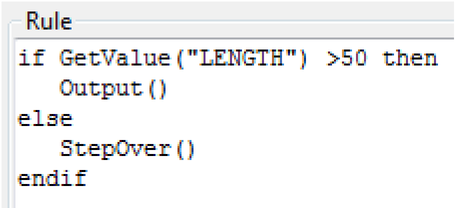
```
if(IsLast()) then
   Output()
else
   StepOver()
endif
```

See also IsFirst (92)

Output

This function outputs the current row. Use this function when you want to output the row that fulfils the condition you specify. The condition can evaluate, for example, the position of the row or the field values on the row.

This rule output values greater than 50. If rows with those values are not found, they are skipped.

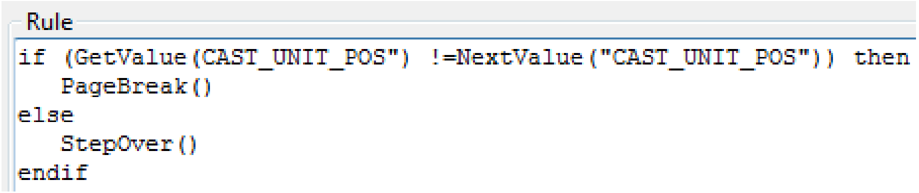


```
if GetValue("LENGTH") >50 then
   Output()
else
   StepOver()
endif
```

PageBreak

This function inserts a page break.

The following rule inserts a page break when the condition is true.

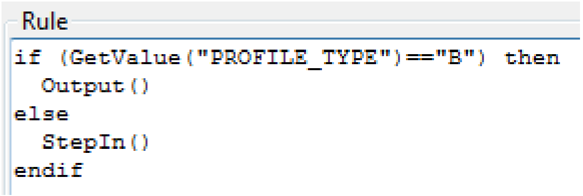


```
if (GetValue(CAST_UNIT_POS") !=NextValue("CAST_UNIT_POS")) then
   PageBreak()
else
   StepOver()
endif
```

StepIn

Use this function when you want to process the rows that are located below the current row in the hierarchy.

The following rule checks the value of a value field on the row. If the value is "B", the row is output, otherwise the rows below in hierarchy are processed. If there are no rows below, the processing continues with the next row on the same level.

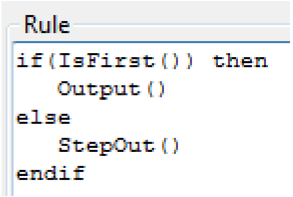


```
if (GetValue("PROFILE_TYPE")=="B") then
  Output()
else
  StepIn()
endif
```

StepOut

Use this function to stop processing the current row and rows below it in the hierarchy, and continue to processing the row higher in the hierarchy. If no row higher in the hierarchy exists or if this was the last row in the template, then the function starts processing again from the first row of the template.

The following rule checks if the row is the first one. If this is true, the row is output, otherwise the processing continues to the row higher in the hierarchy.

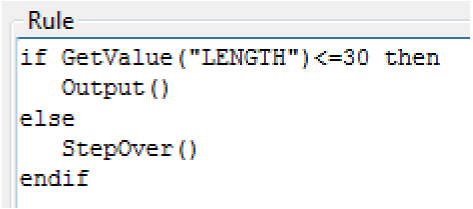


```
if(IsFirst()) then
   Output()
else
   StepOut()
endif
```

StepOver

Use this function when you want to ignore the current row and continue to the next row.

The rule outputs values less than or equal to 30. If those values are not found, they are skipped.



```
if GetValue("LENGTH")<=30 then
   Output()
else
   StepOver()
endif
```

11.5 Attribute functions

Attribute functions get the values of global attributes and summarize them during the template output process. You can access attribute functions in the **Formula Contents** dialog box and in the **Rule Contents** dialog box.

To select a parameter for an attribute function, click the **Select** button in the **Attribute** area of the dialog box. It opens the hierarchical list of global attributes.

Parameters: The parameter must appear inside parentheses and double quotation marks (" ").

In addition to this list you can use:

- free attributes from your product
- a constant parameter CONTENTTYPE that returns the content type of the current row.

See also GetValue (94)

NextValue (94)

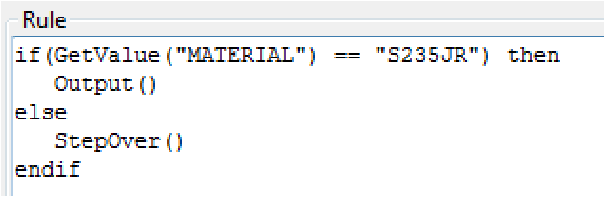
PreviousValue (95)

IsSet (95)

GetValue

Use this function when you want to return the value of the attribute on this output row, i.e. for the currently processed object in the product database.

In the following example, **GetValue** is used in a row rule condition to check if the current row has a desired field value. The rule outputs the row only if the material is **S235JR**.

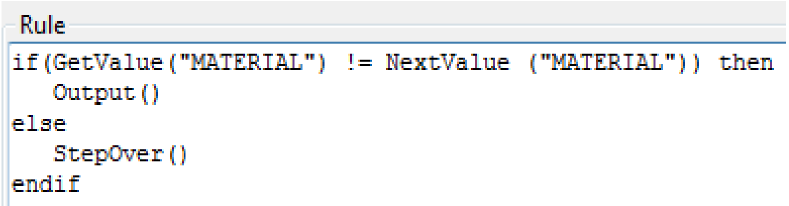


```
if(GetValue("MATERIAL") == "S235JR") then
   Output()
else
   StepOver()
endif
```

NextValue

Use this function when you want to get the value of the attribute on the next output row (not printed yet). You can use this, for example, to inquire material types.

In the following example, **NextValue** is used in a row rule condition to inquire the material on the next row. The rule outputs the current row only if the material on the next row is different. This is a way to avoid outputting duplicate rows in a report.

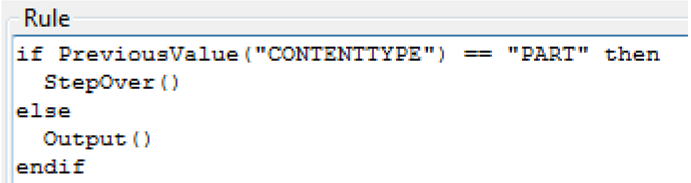


```
if(GetValue("MATERIAL") != NextValue ("MATERIAL")) then
   Output()
else
   StepOver()
endif
```

PreviousValue

Use this function when you want to get the value of the attribute on the previous output row.

In the following example **PreviousValue** is used in a row rule condition to check the content type of the previous row. The rule outputs the current row only if the previous content type is not PART. This is a way to avoid outputting rows not relevant for PART types.

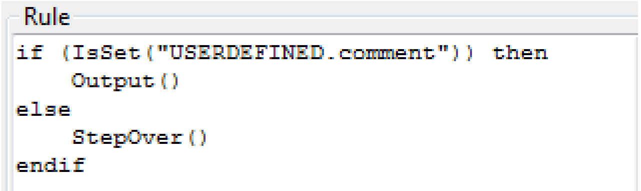


```
if PreviousValue("CONTENTTYPE") == "PART" then
  StepOver()
else
  Output()
endif
```

IsSet

Use this function when you want to check whether a value has been set for a user-defined free attribute. This is only used in rows.

The following rule is an example of using **IsSet** in a row rule condition. The rule outputs the row only if a user-defined "comment" attribute has been set.



```
if (IsSet("USERDEFINED.comment")) then
    Output()
else
    StepOver()
endif
```

All other rows are skipped.

11.6 Value field functions

Value field functions get and summarize data from other value fields in the template. You can access them in the **Formula Contents** dialog box and in the **Rule Contents** dialog box.

In order to access the parameters of the function click **Select** in the **Value Field** area of the dialog box. This opens the hierarchical list of value field names. The parameter must appear inside parentheses and double quotation marks (" ").

Value field functions get values from all processed fields - from the duplicate lines that may have been excluded from the output, and from invisible value fields.

See also GetFieldFormula (96)

CopyField (96)

Sum (97)

Total (97)

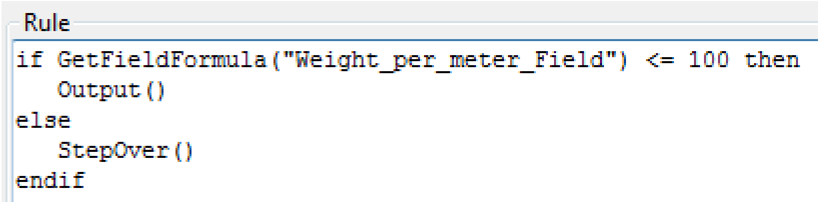
Count (97)

All (98)

GetFieldFormula

Use this function when you want to query the value of a given value field. You can also combine several field formulas into one.

This rule outputs all rows that fulfill the statement:



```
if GetFieldFormula("Weight_per_meter_Field") <= 100 then
   Output()
else
   StepOver()
endif
```

CopyField

This function is intended for copying summary fields from a row lower in the hierarchy into a row higher in the hierarchy. The row higher in the hierarchy should contain a value field with a formula containing the CopyField function. The next row lower in the hierarchy should contain the source value field. CopyField always copies the source value from the row lower in the hierarchy, even if the row is hidden.

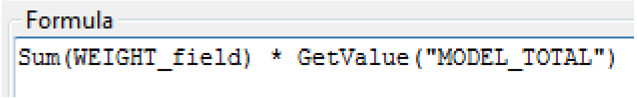
For a header, page headers, page footers or a footer, the function copies the value of the given value field to itself.

If this function is used in row rules, it only copies the values from the row itself and not from other rows at the same level in the hierarchy or rows lower in the hierarchy.

A value field containing the **CopyField** function cannot be sorted.

In this example, there is an **ASSEMBLY** row with two rows on a lower level. One of the lower rows is a **PART** row and the other one is a **SUMMARY** row.

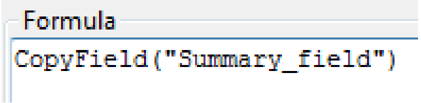
The **SUMMARY** row has a value field **Summary_field** containing the following formula:



Formula

```
Sum(WEIGHT_field) * GetValue("MODEL_TOTAL")
```

The calculated weight is shown on the **ASSEMBLY** row in a value field with the following formula (and you can hide the **SUMMARY** row):



Formula

```
CopyField("Summary_field")
```

See also Hiding rows from output (45)

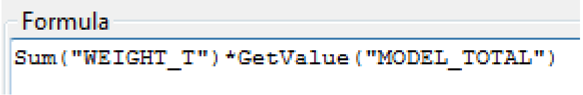
Sum

Use this function when you want to calculate the intermediate sum of a numeric value field. The data type is **Number** or **Number with decimals**. All proceeding values of the parameter before the current output row are counted, and the sum is reset to zero every time after the row is output.

This function is not available for row rules.

To calculate the total sum instead of this intermediate sum, use the **Total** function instead.

This formula calculates the intermediate sum of the value field **WEIGHT_T** and is multiplied with the total value of the model.



Formula

```
Sum("WEIGHT_T")*GetValue("MODEL_TOTAL")
```

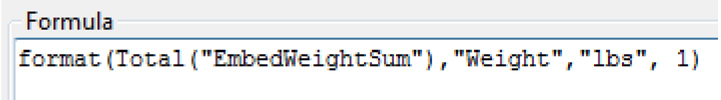
See also Total (97)

Total

Use this function when you want to total the sum of a numeric value field. All the processed values of the value field are counted.

This function is not available for row rules.

The following formula totals the embedded assembly weight in pounds.



Formula

```
format(Total("EmbedWeightSum"),"Weight","lbs", 1)
```

See also Sum (97)

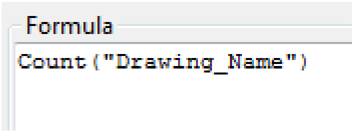
Count

Use this function when you want to calculate the number of value fields that have been output before the current output row. The count is reset after output.

This function is not available for row rules.

To calculate the total number of value fields processed instead of this intermediate count, use the **All** function instead.

This formula returns the intermediate count of value fields with the name **Drawing_Name**.



```
Count("Drawing_Name")
```

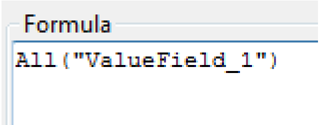
See also All (98)

All

Use this function when you want to calculate the number of value fields that have been output before the current output row.

This function is not available for row rules.

This formula calculates all the previously handled value fields named **ValueField_1**.



```
All("ValueField_1")
```

See also Count (97)

11.7 String operations

String operations are used to convert a text string to something new and are mostly used in value field rules. They cannot operate directly with attributes or value fields; instead, they accept attribute functions and value field functions as parameters.

You can access string functions in the **Formula Contents** dialog box and in the **Rule Contents** dialog box. Select functions from the **String** list in the **Operators** area of the dialog box.

Before you get started ensure that the **Data type** is set to **Text** in the **Value Field Properties** dialog box.

See also

find (98)

match (99)

mid (99)

length (99)

reverse (99)

getat (99)

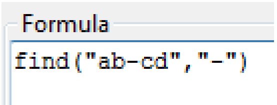
setat (100)

find

This function returns the offset of a substring in a string.

Parameters: string, substring

For example:



```
find("ab-cd","-")
```

The result equals 2.

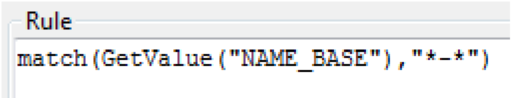
match

This function checks if the given two strings are the same and returns TRUE if they are, FALSE if they are not.

For example, match("abc", "abc")=TRUE and match("abc","b")=FALSE.

Parameters: string1, string2

The following rule uses the **match** function to check if the field value contains a "-" character.



```
match(GetValue("NAME_BASE"),"*-*")
```

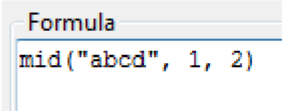
You can also use the wildcard characters ? and * in formulas, for example, match ("aabc", "*b*")=TRUE

mid

This function returns n characters from a string, starting from the position that is given with the offset parameter. If n is omitted, all succeeding characters are returned.

Parameters: string, offset, n

This example returns two characters from the string starting from the second character. The result is "bc".



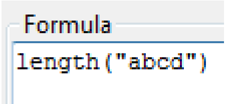
```
mid("abcd", 1, 2)
```

length

This function returns the number of characters in a string.

Parameters: string

This example returns the number of characters in a string. The result is 4.



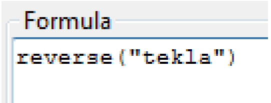
```
length("abcd")
```

reverse

Use this operator to return the mirror image of the string.

Parameter: string

The following example prints "alket" in a report.



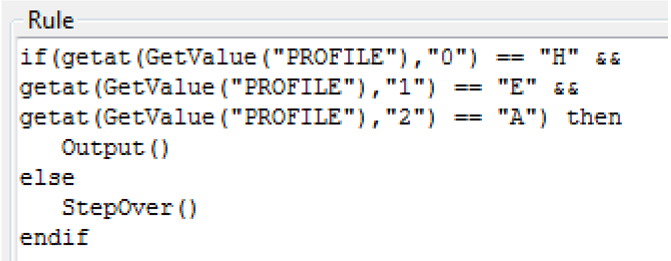
```
reverse("tekla")
```

getat

This function returns the character in a given offset.

Parameters: string, offset

The following rule calculates the offset of the characters H, E and A, essentially filtering and searching for **PROFILE**s that begin with HEA.



Rule

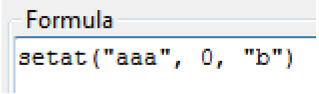
```
if(getat(GetValue("PROFILE"),"0") == "H" &&
getat(GetValue("PROFILE"),"1") == "E" &&
getat(GetValue("PROFILE"),"2") == "A") then
   Output()
else
   StepOver()
endif
```

setat

Use this function to replace a character with another character at the given offset.

Parameters: string, offset, character

The following formula replaces "a" with "b" at the specified offset of 0. The result is "baa".



Formula

```
setat("aaa", 0, "b")
```

11.8 Data type conversion functions

Conversion functions perform conversions between the supported data types: integer, double, string, and vwu (value-with-unit).

You can access conversion functions in the **Formula Contents** dialog box and in the **Rule Contents** dialog box.

Select functions from the **Math** list in the **Operators** area of the dialog box.

Note that string parameters must appear inside double quotation marks (" ") in the parameter list.

See also

int (100)

double (101)

string (101)

vwu (Value with unit) (101)

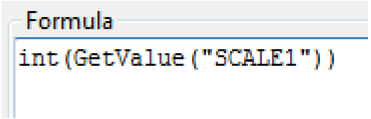
format (102)

int

This function is used to convert an attribute value to an integer.

Parameter: value

This converts a double to an integer. If the attribute value is 3.5, the result is 4.



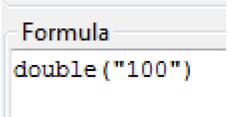
```
int(GetValue("SCALE1"))
```

double

This function is used to convert an attribute value to a double, a number with decimals.

Parameter: value

This formula converts a number expressed in a text string to double format. The result is 100.0000



```
double("100")
```

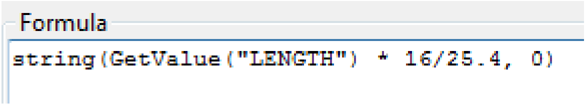
string

This function is used to convert an attribute value to a string.

Parameters: value, MinDecimals, MaxDecimals

- MinDecimals - Minimum number of decimals (optional, effective only when converting double values)
- MaxDecimals = Maximum number of decimals (optional, effective only when converting double values), default = same as MinDecimals

This formula multiplies the **LENGTH** attribute value by 16, divides that by 25.4 and converts the result to a string. The string contains a number with no decimals.



```
string(GetValue("LENGTH") * 16/25.4, 0)
```

A simpler example would be:

string(3.14, 1) = "3.1".

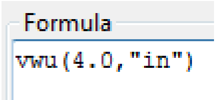
vwu (Value with unit)

This function is used to convert a specified unit to a product unit, which is a number with decimals.

Parameters: value, unit string ("ft" = Feet, "in" = Inch, "m" = Meter, "cm" = Centimeter, "mm" = Millimeter, "rad" = Radian, "deg" = Degree)

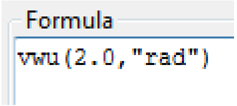
Below are some examples from value field formulas.

This formula converts 4.0 inches to a product unit. It returns 101.60 mm, if length unit is set to mm and decimals are set to 2 in the **Value Field Properties** dialog box.



```
vwu(4.0,"in")
```

This formula converts 2.0 radians to a product unit. It returns 114.59 degrees, if angle is set to degrees and decimals are set to 2 in the **Value Field Properties** dialog box.



```
vwu(2.0,"rad")
```

format

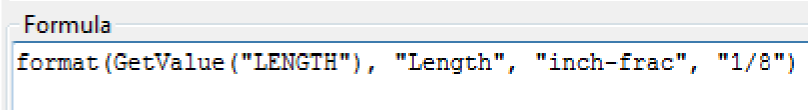
This function converts an attribute value to a formatted information string that is product-specific. The parameters available for the function are listed in the value field meanings definition file. Please refer to your product documentation for information on the available formats and how to use them.

The result depends on the product-specific definitions for meaning string, unit string and precision.

Parameters: value, meaning string, unit string, precision string or number of decimals

This value field formula creates a formatted string using the LENGTH attribute value, the "Length" meaning string, the "inch-frac" unit string and "1/8" precision.

The result depends on the product-specific definitions for meaning string, unit string and precision.



```
format(GetValue("LENGTH"), "Length", "inch-frac", "1/8")
```

11.9 Mathematical operations

Mathematical operations can appear in row rules and value field formulas. All operations accept integer, double and string operands / parameters.

Operations cannot operate directly with attributes or value fields; instead, they accept attribute functions and value field functions as parameters.

See also Arithmetic operators (102)

Trigonometric functions (103)

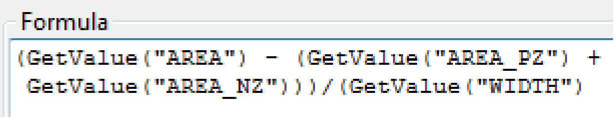
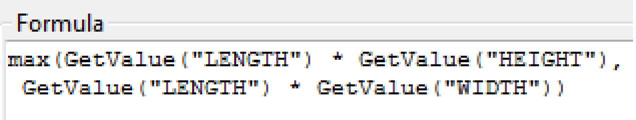
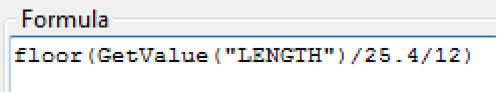
Statistical functions (104)

Data type conversion functions (100)

Miscellaneous functions (106)

Arithmetic operators

You can use arithmetic operators by inserting them in formulas or rules. The standard evaluation order is applied when resolving the values.

Operator	Description	Example
+	Addition. With string parameters the result is string concatenation.	This formula calculates the perimeter by adding up the total area of all the faces, then subtracting the area of the top and bottom face, leaving the total area of the perimeter faces. Divide the area of the perimeter faces by the plate thickness, and it should give you the perimeter length.  Formula (GetValue("AREA") - (GetValue("AREA_PZ") + GetValue("AREA_NZ")))/(GetValue("WIDTH")
-	Subtraction	See the formula above.
*	Multiplication	This formula uses the multiplication to calculate areas and find the biggest value of similar fields.  Formula max(GetValue("LENGTH") * GetValue("HEIGHT"), GetValue("LENGTH") * GetValue("WIDTH"))
/	Division	This formula uses division to transform imperial units to feet and the floor function to round down the result.  Formula floor(GetValue("LENGTH")/25.4/12)

See also Value field formulas (56)

Controlling row output with rules (46)

Trigonometric functions

Trigonometric functions are used to calculate angles between members or inside custom components and are used in angle attributes. The unit of angle for trigonometric functions is given with the prefixes:

- d is degree (values 0-360).
- r is radians (this is the default, values 0-2 π?).
- g is gradient (grad, values 0-400).

For example:

- sin (d180)
- sin (r3.14) or sin (3.14)
- sin (g200)

You can access trigonometric functions in the **Formula Contents** dialog box and in the **Rule Contents** dialog box.

Select functions from the **Math** list in the **Operators** area of the dialog box.

Function	Parameters	Description
sin	angle	Calculates sine.
cos	angle	Calculates cosine.
tan	angle	Calculates tangent.
asin	angle	Calculates the inverse of sin, return value in radians.
acos	angle	Calculates the inverse of cos, return value in radians.
atan	angle	Calculates the inverse of tan, return value in radians.
sinh	angle	Calculates the hyperbolic sine.
cosh	angle	Calculates the hyperbolic cosine.
tanh	angle	Calculates the hyperbolic tangent.
atan2	y, x	Calculates the direction angle of vector (x, y), return value in radians.

Statistical functions

Statistical functions can be used to calculate key figures of the data that is processed and output with the template.

You can access statistical functions in the **Formula Contents** dialog box and in the **Rule Contents** dialog box. Select functions from the **Math** list in the **Operators** area of the dialog box.

See also

ceil (104)

floor (105)

min (105)

max (105)

sqsum (105)

ave (106)

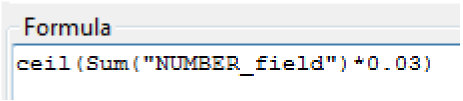
sqave (106)

ceil

Use this function when you want to round up a value to the next bigger integer.

Parameter: parameter

This formula rounds up the sum of the **NUMBER_field** multiplied by 0.03.



```
ceil(Sum("NUMBER_field")*0.03)
```

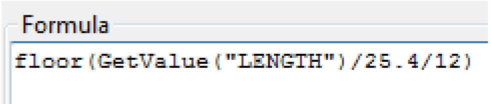
See also floor (105)

floor

Use this function when you want to round down a value to the next smaller integer.

Parameter: parameter

This formula uses division to transform imperial units to feet and the floor function to round down the result.



```
floor(GetValue("LENGTH")/25.4/12)
```

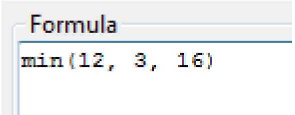
See also ceil (104)

min

This function calculates the minimum value of parameters.

Parameters: x, y, z,...

The following formula calculates the lowest number in the series in a report.



```
min(12, 3, 16)
```

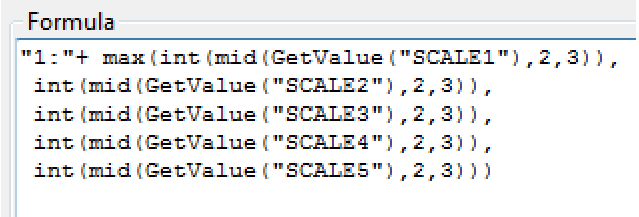
See also max (105)

max

Use this function to calculate the maximum value of parameters.

Parameters: x, y, z,...

The following formula returns the biggest scale used in drawing views (SCALE1 - 5). As the value is string 1:20 for example, you need to convert the numbers following ":" to integer format before the **max** comparison.



```
"1:"+ max(int(mid(GetValue("SCALE1"),2,3)),
 int(mid(GetValue("SCALE2"),2,3)),
 int(mid(GetValue("SCALE3"),2,3)),
 int(mid(GetValue("SCALE4"),2,3)),
 int(mid(GetValue("SCALE5"),2,3)))
```

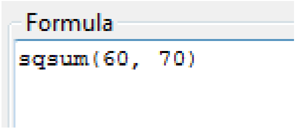
See also min (105)

sqsum

This function calculates the sum of squares.

Parameters: x, y, z,...

The following formula adds the square of 60 plus the square of 70, which prints 8500 in a report.



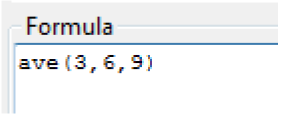
```
sqsum(60, 70)
```

ave

This function calculates the average of parameters.

Parameters: x, y, z,...

Use the following syntax to calculate the average.



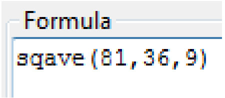
```
ave(3,6,9)
```

sqave

This function calculates the average of the squared parameters.

Parameters: x, y, z,...

The following formula calculates the sum of squares. The report prints 2646.



```
sqave(81,36,9)
```

Miscellaneous functions

Miscellaneous functions are a selection of additional mathematical functions.

You can access them in the **Formula Contents** dialog box and in the **Rule Contents** dialog box. Select functions from the **Math** list in the **Operators** area of the dialog box:

See also

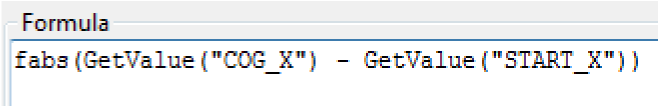
- fabs (106)
- exp (107)
- ln (107)
- log (107)
- sqrt (107)
- mod (108)
- pow (108)
- hypot (108)
- n! (108)
- round (109)
- and (109)
- or (109)

fabs

This function calculates the absolute value of a number.

Parameter: number

The following formula calculates the local center of gravity from the start point with **fabs** giving the absolute number. For example, the absolute value of -3.54 is 3.54.



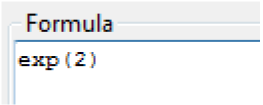
```
fabs(GetValue("COG_X") - GetValue("START_X"))
```

exp

This function calculates the involution of e, to a specified power. e is Euler's number.

Parameter: power

The following formula calculates the involution of e to the power of two, which equals 7.389056 in a report.



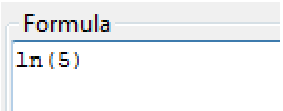
```
exp(2)
```

ln

This function returns the natural logarithm of the parameter (base number e).

Parameter: number

The following formula calculates the natural logarithm of 5, which prints 1,60944 in a report.



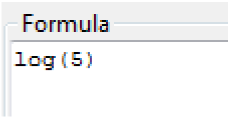
```
ln(5)
```

log

This function returns the logarithm of the parameter (base number 10)

Parameter: number

The following formula calculates the logarithm of the number in the brackets. The report prints the final calculation of 0,698970. The length of the printed number (with decimals) is determined in the **Value Field Properties** dialog box in the **Length** field.



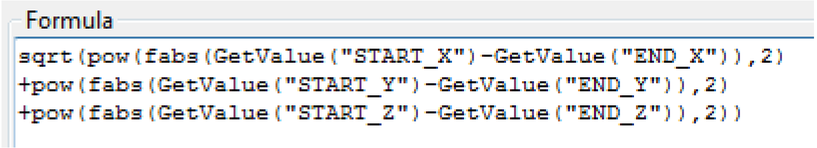
```
log(5)
```

sqrt

This function calculates the square root of a number.

Parameter: number

The following formula acts as a filter to calculate the distance between the start and end points, where **sqrt** calculates the square root and **pow** calculates the involution.



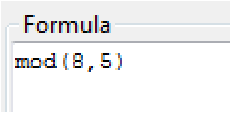
```
sqrt(pow(fabs(GetValue("START_X")-GetValue("END_X")),2)
+pow(fabs(GetValue("START_Y")-GetValue("END_Y")),2)
+pow(fabs(GetValue("START_Z")-GetValue("END_Z")),2))
```

mod

This function calculates modulo, the remainder after division.

Parameters: dividend, divider

The following formula calculates modulo and prints 3 in a report.



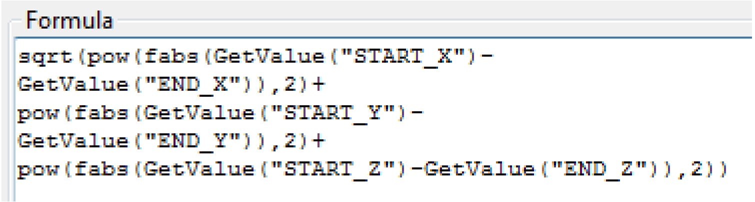
```
mod(8,5)
```

pow

This function calculates the involution of a value.

Parameters: base number, power

This formula calculates the distance between the start and end points, where **sqrt** calculates the square root and **pow** calculates the involution.



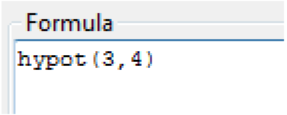
```
sqrt(pow(fabs(GetValue("START_X")-
GetValue("END_X")),2)+
pow(fabs(GetValue("START_Y")-
GetValue("END_Y")),2)+
pow(fabs(GetValue("START_Z")-GetValue("END_Z")),2))
```

hypot

This function calculates the hypotenuse of a triangle.

Parameters: side, side

The following formula calculates the hypotenuse and prints 5 in a report.



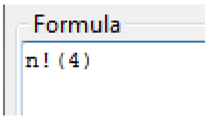
```
hypot(3,4)
```

n!

This function calculates the factorial of a number.

Parameter: number

The following formula calculates the factorial of 4. This translates to 4x3x2x1 and a report would display the result, 24.



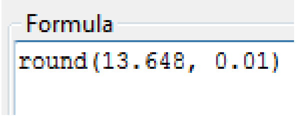
```
n!(4)
```

round

This function rounds off a number according to given accuracy.

Parameters: number, accuracy

This formula rounds the number to two decimals, which is specified in the formula (0.01). The report prints 13.65.



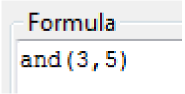
```
round(13.648, 0.01)
```

and

This is a binary operation (and similar to & in the C language).

Parameters: parameter 1, parameter 2

The following example prints 1 in the report.



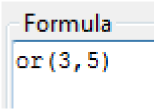
```
and(3,5)
```

or

This is a binary operation (and similar to | in the C language).

Parameters: parameter 1, parameter 2

The following example prints 7 in the report.



```
or(3,5)
```

Index