

Template Editor

User's Guide

Product version 4.50

February 2026

2026 © Trimble Inc.

Contents

- Chapter 1. Template Editor 4.50..... 10**
 - © 2026 Copyright Trimble Inc. and Affiliates. All rights reserved..... 10
 - Conventions used in this guide..... 10
- Chapter 2. About Template Editor..... 13**
 - New in this version..... 13
 - Main features of Template Editor..... 14
 - What are Template Editor templates?..... 15
 - Template Editor components..... 16
 - Template Editor objects..... 16
 - Template definition files..... 18
- Chapter 3. Template Editor user interface..... 19**
 - Template Editor workarea..... 19
 - Template Editor toolbars..... 19
 - The Content Browser in Template Editor..... 20
 - The preview pane in Template Editor..... 20
 - Modifying Template Editor interface components..... 21
 - Template Editor windows..... 21
 - Panning in Template Editor windows..... 22
 - Zooming in and out in Template Editor..... 22
 - Restoring the original view in Template Editor..... 23
 - Setting up the Template Editor interface..... 23
 - Modifying general preferences in Template Editor..... 23
 - Modifying workarea preferences in Template Editor..... 25
 - File Locations preferences in Template Editor..... 26
 - Modifying File Locations preferences in Template Editor..... 27
 - The template grid in Template Editor..... 28
 - Activating the grid in Template Editor..... 28

Setting the grid density in Template Editor.....	29
Showing or hiding the grid in Template Editor.....	29
Tips for the Template Editor interface.....	30
Keyboard shortcuts in Template Editor.....	30
Pop-up menus in Template Editor.....	32
The status bar in Template Editor.....	33
Setting presentation file and pen settings in Template Editor.....	33
Tooltips in Template Editor.....	34
Chapter 4. Introduction to templates.....	35
Creating and saving a template in Template Editor.....	35
Creating and saving a pre-defined template in Template Editor.....	36
Opening and closing templates in Template Editor.....	38
Modifying template page properties in Template Editor.....	39
Changing the template type in Template Editor.....	40
Chapter 5. Template Editor components and objects.....	42
Inserting components and objects in Template Editor.....	42
Drawing a polyline in Template Editor.....	43
Inserting a text object in Template Editor.....	44
Inserting a graphical field in Template Editor.....	44
Setting the visibility of texts, drawing objects and graphical fields in Template Editor.....	45
Inserting a template page header or page footer.....	47
Selecting components and objects in Template Editor.....	48
Cutting, copying and pasting template components and objects.....	49
Duplicating objects in Template Editor.....	50
Modifying component and object properties in Template Editor.....	50
Modifying the name of a template component or object.....	51
Modifying the size of a template component or object.....	52
Moving a template component.....	52
Cropping template components.....	53

Changing the template component type.....	53
Grouping or ungrouping template objects.....	53
Moving a template object.....	54
Trimming template objects in Template Editor.....	55
Extending template objects in Template Editor.....	56
Formatting template objects with format painter in Template Editor.....	56
Modifying template text properties.....	57
Setting default properties for new template objects.....	58
Rotating template objects.....	59
Aligning template objects.....	60
Aligning template objects to the grid.....	60
Trimming template object locations.....	60
Tips for template component and object tasks.....	61
Output options for template headers and footers.....	61
Special drawing operations in Template Editor.....	62
Snapping line objects in Template Editor.....	63
Drawing with the numeric pad in Template Editor.....	64
Drawing with the grid in Template Editor.....	64
Viewing or hiding template objects.....	65
Using the Selection toolbar.....	65
Font properties for texts and value fields in Template Editor.....	66
Searching for text in Template Editor.....	67
Chapter 6. Template rows.....	68
Creating rows in Template Editor.....	68
Content types in Template Editor.....	69
Template row output properties.....	69
Column settings in Template Editor.....	72
Modifying the content type of a row in Template Editor.....	73
Modifying the sort type of a row in Template Editor.....	74

Modifying column settings in Template Editor.....	75
Hiding template rows from output in Template Editor.....	75
Modifying template row hierarchy.....	76
Controlling row output with rules in Template Editor.....	76
Creating and saving a template row rule.....	77
Creating a template row rule with Rule Wizard.....	78
Row rule content properties in Template Editor.....	79
Modifying a template row rule.....	81
Overriding template object colors with row rules in Template Editor.....	82
Pre-defined row rules in Template Editor.....	83
Duplicate lines in Template Editor.....	84
Chapter 7. Value fields in Template Editor.....	85
Creating a value field in Template Editor.....	85
Value field output properties in Template Editor.....	86
Value field meanings in Template Editor.....	90
Setting the sort order of a value field in Template Editor.....	91
Setting the priority of a value field in Template Editor.....	91
Setting the data type of a value field in Template Editor.....	92
Assigning value field meanings in Template Editor.....	93
Setting the layout of value fields in Template Editor.....	94
Resizing value field fonts automatically in Template Editor.....	94
Hiding value fields from output in Template Editor.....	95
Formatting value fields in Template Editor.....	96
Aligning value field texts in output in Template Editor.....	96
Value field formulas in Template Editor.....	97
Creating and saving a formula in Template Editor.....	97
Formula content properties in Template Editor.....	98
Modifying a value field formula in Template Editor.....	100
Get a single piece of data from the database in Template Editor.....	101

Creating a listing of template objects from the database.....	102
Creating a summary field in Template Editor.....	103
Outputting blank value fields in Template Editor.....	104
Tips for value fields in Template Editor.....	104
String translations of texts in Tekla Structures.....	105
Setting Tekla Structures advanced options for value fields.....	105
Chapter 8. Symbols and pictures in Template Editor.....	107
Symbol and picture libraries in Template Editor.....	107
Adding symbols to a template in Template Editor.....	108
Adding pictures to a template in Template Editor.....	108
Adding a logo to a template in Template Editor.....	109
Symbol properties in Template Editor.....	110
Picture properties in Template Editor.....	111
Viewing and modifying symbol file contents in Tekla Structures.....	112
Changing the symbol and the symbol file in Template Editor.....	113
Assigning rules to symbols in Template Editor.....	114
Assigning rules to pictures in Template Editor.....	115
Modifying a symbol in Template Editor.....	116
Setting paths for symbols and pictures in Template Editor.....	116
Chapter 9. AutoCAD and Microstation files in Template Editor.....	118
Importing AutoCAD and MicroStation files with import method defined in Template Editor.....	118
Importing AutoCAD and MicroStation files with no import method defined in Template Editor.....	120
Setting fill type and font import settings in Template Editor.....	120
Limitations in AutoCAD and MicroStation file imports in Template Editor	121
AutoCAD and Microstation import errors in Template Editor.....	122
Modifying imported AutoCAD and MicroStation files in Template Editor.....	123
Chapter 10. Template attributes and Template Editor.....	125
User-defined template attributes and Template Editor.....	126
Adding user-defined template attributes in Template Editor.....	128

Adding comments to user-defined template attributes in Template Editor.....	129
Adding hierarchy to user-defined template attributes in Template Editor.....	130
Free attributes in Template Editor.....	131
Adding free attributes in Template Editor.....	131
Deleting free attributes in Template Editor.....	132
Modifying free attributes in Template Editor.....	133
Global attributes and Template Editor.....	134
Chapter 11. Using templates in Trimble products.....	135
Outputting a template created with Template Editor.....	135
Output options for templates created with Template Editor.....	136
Material listings created with Template Editor.....	137
Hierarchical listings created with Template Editor.....	138
Labels created with Template Editor.....	140
Creating an HTML report with Template Editor.....	141
Chapter 12. Template Editor formula and rule reference.....	143
Adding comments to row rules and value field formulas in Template Editor.....	144
Offset for template objects.....	144
Conditional structure for row rules and formulas in Template Editor.....	145
Comparison and logical operators.....	146
Control functions in Template Editor.....	149
IsFirst.....	149
IsLast.....	150
Output.....	150
PageBreak.....	151
StepIn.....	151
StepOut.....	152
StepOver.....	152
Attribute functions in Template Editor.....	152
GetValue.....	153

NextValue.....	153
PreviousValue.....	154
IsSet.....	154
Value field functions in Template Editor.....	154
GetFieldFormula.....	155
CopyField.....	155
Sum.....	157
Total.....	157
Count.....	158
All.....	158
String operations in Template Editor.....	159
find.....	160
findany.....	160
match.....	161
mid.....	161
length.....	161
reverse.....	162
getat.....	162
setat.....	162
replace.....	163
toupper.....	163
tolower.....	163
fvf.....	164
newline.....	166
join.....	167
join2.....	167
asc.....	168
Data type conversion functions in Template Editor.....	168
int.....	169

double.....	169
string.....	169
vwu (Value with unit).....	170
format.....	171
Mathematical operations in Template Editor.....	171
Arithmetic operators in Template Editor.....	172
Trigonometric functions in Template Editor.....	173
Statistical functions in Template Editor.....	174
Miscellaneous functions in Template Editor.....	177
Index.....	a



© 2026 Copyright Trimble Inc. and Affiliates. All rights reserved.

Use of the Software and of this Software Manual are governed by a License Agreement which determines whether you are an authorized user of the Software and the Software Manual. The warranties and disclaimers set forth in the License Agreement apply to the Software and the Software Manual. Neither the Trimble entity granting the license nor any of its affiliates assume responsibility that the text is free of technical inaccuracies or typographical errors. The right to make changes and additions to this manual is reserved.

Trimble and certain product names are registered trademarks of Trimble Inc. in the United States, the European Union and other countries and may have similar statutory protections. Trademarks of third parties are not mentioned in this Manual to suggest an affiliation with or endorsement by their owners.

Elements of the software described in this Manual may be the subject of pending patent applications in the European Union and/or other countries.

Trimble Technology kits and tools are property of Trimble Finland.

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.
<i>Italic bold</i>	New terms are in <i>italic bold</i> when they appear in the current context for the first time.
<code>Monospace</code>	Extracts of program code, HTML, or other material that you would normally edit in a text editor, appear in <code>monospaced</code> 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:



Tip:

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



Note:

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.



Warning:

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



CAUTION:

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

Chapter 2. About Template Editor

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

You can produce labels, reports and legends, 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 modify the existing template or create new ones to suit your needs.

Find out more:

[New in this version \(on page 13\)](#)

[Main features of Template Editor \(on page 14\)](#)

[What are Template Editor templates? \(on page 15\)](#)

[Template Editor components \(on page 16\)](#)

[Template Editor objects \(on page 16\)](#)

[Template definition files \(on page 18\)](#)

New in this version

Here is what is new in version 4.50 of Template Editor:

- options for multi-line text objects and ability to align them: [Modifying template text properties \(on page 57\)](#)
- options for multi-line value fields and ability to align them: [Value field output properties in Template Editor \(on page 86\)](#)
- an advanced option for combining unique value field text values into one string if rows are combined in output: [Value field output properties in Template Editor \(on page 86\)](#)
- options for defining a background color for rows: [Template row output properties \(on page 69\)](#)
- option for positioning of rule-based images (H/V), allowing precise fitting in their allocated space in a template: [Picture properties in Template Editor \(on page 111\)](#)
- option to set the coding for characters to always be output in UTF-8: [Modifying template page properties in Template Editor \(on page 39\)](#)
- a new string function: [findany \(on page 160\)](#)

- option to remove the form feed character /f from output when a page break occurs during the output of a textual template: [Modifying template page properties in Template Editor \(on page 39\)](#)
- optional for setting the dynamic height of templates during output in Tekla: [Modifying template page properties in Template Editor \(on page 39\)](#)
- options for defining borders for text objects, value fields and rows in textual templates that are output in Excel format: [Modifying template text properties \(on page 57\)](#), [Value field output properties in Template Editor \(on page 86\)](#) and [Template row output properties \(on page 69\)](#)

Main features of Template Editor

With Template Editor you can :

- 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

[Template Editor user interface \(on page 19\)](#)

[Column settings in Template Editor \(on page 72\)](#)

[Output options for templates created with Template Editor \(on page 136\)](#)

[Template Editor objects \(on page 16\)](#)

[Hiding template rows from output in Template Editor \(on page 75\)](#)

[Hiding value fields from output in Template Editor \(on page 95\)](#)

[Free attributes in Template Editor \(on page 131\)](#)

[AutoCAD and Microstation files in Template Editor \(on page 118\)](#)

[Symbols and pictures in Template Editor \(on page 107\)](#)

[Creating an HTML report with Template Editor \(on page 141\)](#)

What are Template Editor templates?

Templates are descriptions of forms and tables that can be included in your products. They are either graphical or textual and 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.

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. They can also employ different font types and settings. Graphical template definitions have the extension `.tpl`.

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, 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 extension `.rpt`.

See also

[Font properties for texts and value fields in Template Editor \(on page 66\)](#)

[Column settings in Template Editor \(on page 72\)](#)

[File Locations preferences in Template Editor \(on page 26\)](#)

Template Editor 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 \(on page 68\)](#)

[Inserting components and objects in Template Editor \(on page 42\)](#)

[Modifying the name of a template component or object \(on page 51\)](#)

[Modifying the size of a template component or object \(on page 52\)](#)

Template Editor objects

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

Drawing shapes	Basic geometric shapes like lines, arcs, polylines rectangles, circles and polygons. They only appear in graphical templates.
Text objects	Static texts are headings or title line texts. Text objects can appear in both graphical templates and textual templates.
Symbols	Symbols from product symbol libraries.
Pictures	Files of different formats can be inserted into graphical templates.
Imported files	AutoCAD and MicroStation files.
Field objects	<p>Template field objects are graphical fields and value fields, which can be used to get data from the product.</p> <p>Value fields contain textual information that is referenced from the product database during template output. Data is retrieved by using value field formulas.</p> <p>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.</p>

See also

[Inserting components and objects in Template Editor \(on page 42\)](#)

[Modifying component and object properties in Template Editor \(on page 50\)](#)

[Drawing with the numeric pad in Template Editor \(on page 64\)](#)

[Value fields in Template Editor \(on page 85\)](#)

[Inserting a graphical field in Template Editor \(on page 44\)](#)

[Adding a logo to a template in Template Editor \(on page 109\)](#)

[Adding symbols to a template in Template Editor \(on page 108\)](#)

[Adding pictures to a template in Template Editor \(on page 108\)](#)

[Importing AutoCAD and MicroStation files with import method defined in Template Editor \(on page 118\)](#)

[Importing AutoCAD and MicroStation files with no import method defined in Template Editor \(on page 120\)](#)

Template definition files

Template Editor definition files contain product-specific information that is used in rows, value fields and graphical fields. Definition files are text files that can be modified. 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 in Template Editor \(on page 69\)](#)

[Value field meanings in Template Editor \(on page 90\)](#)

[Global attributes and Template Editor \(on page 134\)](#)

Chapter 3. Template Editor user interface

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

Find out more:

[Template Editor workarea \(on page 19\)](#)

[Template Editor toolbars \(on page 19\)](#)

[The Content Browser in Template Editor \(on page 20\)](#)

[The preview pane in Template Editor \(on page 20\)](#)

[Modifying Template Editor interface components \(on page 21\)](#)

[Template Editor windows \(on page 21\)](#)

[Setting up the Template Editor interface \(on page 23\)](#)

[Tips for the Template Editor interface \(on page 30\)](#)

Template Editor workarea

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

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 Editor windows \(on page 21\)](#)

[Modifying workarea preferences in Template Editor \(on page 25\)](#)

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.

The Content Browser in Template Editor

In the **Content Browser** you can see a folder list that describes the contents of an open template. 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.

There are several things you can do to modify it in its relation to the **Workarea**:

- If you right-click in the **Content Browser**, a pop-up menu opens access to the most common viewing and modifying operations.
- You can resize the **Content Browser** by clicking and dragging the divider between it and the **Preview Pane**. Resize them both by undocking them and dragging from the window borders.
- You can also undock the **Content Browser** and move it outside of the main window.
- Collapse the **Content Browser** if you need more space in the **Workarea** to work with templates. Click the arrow button to the right of the **Sort and drawing order** toolbar.

See also

[The preview pane in Template Editor \(on page 20\)](#)

[Template Editor components and objects \(on page 42\)](#)

[Modifying component and object properties in Template Editor \(on page 50\)](#)

[Viewing or hiding template objects \(on page 65\)](#)

[Searching for text in Template Editor \(on page 67\)](#)

The preview pane in Template Editor

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

You can resize it vertically by clicking and dragging the divider between it and the **Content Browser**. Resize the **Preview Pane** and the **Content Browser** horizontally by undocking them and dragging from the window borders.

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

[The Content Browser in Template Editor \(on page 20\)](#)

Modifying Template Editor 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 or drag the move handle at the top of the Content Browser .
Dock or undock toolbars	Double-click or drag the move handle on the toolbar.
View or hide the Content Browser	Click View > Content Browser > Visible .
Collapse or expand the Content Browser	Click the arrow button next to the Sort and drawing order toolbar at the top of the Content Browser .
View or hide toolbars	Click View > Toolbars > and the toolbar to view or hide.

See also

[Hiding value fields from output in Template Editor \(on page 95\)](#)

[Viewing or hiding template objects \(on page 65\)](#)

Template Editor windows

Template window operations include focusing and zooming in a template window.

Template Editor is a single document interface application, meaning you can have several instances of it running at one time.

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

Find out more:

[Panning in Template Editor windows \(on page 22\)](#)

[Zooming in and out in Template Editor \(on page 22\)](#)

[Restoring the original view in Template Editor \(on page 23\)](#)

Panning in Template Editor windows

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).

You can also use the middle mouse button to pan in the template.

3. Repeat step 2 to continue panning. Press **Esc** to stop panning.

Zooming in and out in Template Editor

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.

You can also use the middle mouse button to zoom in and out.

3. Repeat step 2 to continue zooming. To stop zooming, press **Esc**.

Restoring the original view in Template Editor

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

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

Setting up the Template Editor 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 value fields and row rules to gather template and part-related information at run time.

Find out more:

[Modifying general preferences in Template Editor \(on page 23\)](#)

[Modifying workarea preferences in Template Editor \(on page 25\)](#)

[File Locations preferences in Template Editor \(on page 26\)](#)

[Modifying File Locations preferences in Template Editor \(on page 27\)](#)

[The template grid in Template Editor \(on page 28\)](#)

[Activating the grid in Template Editor \(on page 28\)](#)

[Setting the grid density in Template Editor \(on page 29\)](#)

[Showing or hiding the grid in Template Editor \(on page 29\)](#)

Modifying general preferences in Template Editor

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

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

To set...	Modify this...
Graphical units	<p>Sets the measurement unit for graphical templates to millimeters or inches.</p> <p>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)).</p>
Always ask for import method	<p>Click this option if you want the Select Import Method dialog box to appear whenever you import AutoCAD or MicroStation files.</p> <p>If you leave the option unchecked, you will have to double-click for the Select Import Method dialog box to appear.</p>
Show TrueType fonts	<p>Select this if you want to use TrueType fonts in Template Editor. If unselected, you can use only Tekla fonts.</p>
Use font cap height for new graphical templates	<p>Select this if you want the font size to be calculated in the final output of a template exactly as it has been defined. This means you have final control over how the text is output.</p>
Reset messages to visible	<p>Resets the user messages setting so that all messages are displayed again.</p>

See also

[Importing AutoCAD and MicroStation files with import method defined in Template Editor \(on page 118\)](#)

[Importing AutoCAD and MicroStation files with no import method defined in Template Editor \(on page 120\)](#)

Modifying workarea preferences in Template Editor

Workarea preferences allow you to change the appearance 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

[Template Editor workarea \(on page 19\)](#)

[Modifying general preferences in Template Editor \(on page 23\)](#)

[File Locations preferences in Template Editor \(on page 26\)](#)

[The template grid in Template Editor \(on page 28\)](#)

[Setting presentation file and pen settings in Template Editor \(on page 33\)](#)

File Locations preferences in Template Editor

File locations 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.

Definition file	Description
	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.
Color Map	Sets the color map file to use.
Advanced options	Sets a file where you can find additional attributes to apply to value field formulas and row rules.

See also

[Row rule content properties in Template Editor \(on page 79\)](#)

[Formula content properties in Template Editor \(on page 98\)](#)

Modifying File Locations preferences in Template Editor

File locations 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 Locations preferences in Template Editor \(on page 26\)](#)

The template grid in Template Editor

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 in Template Editor \(on page 28\)](#)

[Setting the grid density in Template Editor \(on page 29\)](#)

[Showing or hiding the grid in Template Editor \(on page 29\)](#)

Activating the grid in Template Editor

Activating the grid means that template objects will be aligned or "snapped" 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 > Snap** on the pop-up menu.

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

You can also activate the grid by clicking the **Snap** check box in the **Grid Settings** toolbar.

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 \(on page 60\)](#)

[Showing or hiding the grid in Template Editor \(on page 29\)](#)

[Setting the grid density in Template Editor \(on page 29\)](#)

[Snapping line objects in Template Editor \(on page 63\)](#)

Setting the grid density in Template Editor

If you want to align objects to the grid points automatically as you draw or modify them, ensure 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** values.
3. Click **OK** to save the changes.

You can also set the grid density in the **Grid Settings** toolbar by entering values in the **Horizontal** and **Vertical** fields.

Grid density settings are saved in the template you are working on.

See also

[Activating the grid in Template Editor \(on page 28\)](#)

[Showing or hiding the grid in Template Editor \(on page 29\)](#)

[Aligning template objects to the grid \(on page 60\)](#)

Showing or hiding the grid in Template Editor

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** on the pop-up menu.

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

You can also show or hide the grid by clicking the **Visible** check box in the **Grid Settings** toolbar.

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

See also

[Activating the grid in Template Editor \(on page 28\)](#)

[Aligning template objects to the grid \(on page 60\)](#)

[Setting the grid density in Template Editor \(on page 29\)](#)

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.

Find out more:

[Keyboard shortcuts in Template Editor \(on page 30\)](#)

[The status bar in Template Editor \(on page 33\)](#)

[Pop-up menus in Template Editor \(on page 32\)](#)

[Setting presentation file and pen settings in Template Editor \(on page 33\)](#)

[Tooltips in Template Editor \(on page 34\)](#)

Keyboard shortcuts in Template Editor

Keyboard shortcuts in Template Editor:

Shortcut	Action
Ctrl+N	Creates a new template file.

Shortcut	Action
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.
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.

Shortcut	Action
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 operation 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 in Template Editor

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 on 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 in Template Editor \(on page 91\)](#)

[Template Editor windows \(on page 21\)](#)

[Modifying template page properties in Template Editor \(on page 39\)](#)

[Modifying component and object properties in Template Editor \(on page 50\)](#)

The status bar in Template Editor

The status bar shows information on the length of drawing objects as they are being inserted; and the selected tool or command. It is located on the bottom left side of the main window.

See also

[Drawing with the numeric pad in Template Editor \(on page 64\)](#)

Setting presentation file and pen settings in Template Editor

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 Locations** 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 Locations preferences in Template Editor \(on page 26\)](#)

Tooltips in Template Editor

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 \(on page 19\)](#)

Chapter 4. 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.

Find out more:

[Creating and saving a template in Template Editor \(on page 35\)](#)

[Creating and saving a pre-defined template in Template Editor \(on page 36\)](#)

[Opening and closing templates in Template Editor \(on page 38\)](#)

[Modifying template page properties in Template Editor \(on page 39\)](#)

[Changing the template type in Template Editor \(on page 40\)](#)

Creating and saving a template in Template Editor

Before you start, we recommend you define various 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.

To create a new template based on a pre-defined template, select a file from the pre-defined templates section in the **Template Types** window.

3. Click **OK**.

An empty template opens in the **Workarea** and you are ready to begin working with the template. Or in the case of a pre-defined template, one that is ready to be populated.

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.

or

Click **File > Save as version** to save the template in an older supported version of Template Editor. Click the version from the dialog that opens. If you save it in an older version, you are presented

with a list of unsupported features. The standard file selection dialog box will appear for you to specify the name, save location and file extension. Template Editor asks if you want to overwrite the existing file if you save it in a different version.

5. Click **OK**.

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



Note:

Templates created with version 3.60 or later are not compatible with previous versions because of expanded attribute options.

See also

[Modifying general preferences in Template Editor \(on page 23\)](#)

[Modifying workarea preferences in Template Editor \(on page 25\)](#)

[File Locations preferences in Template Editor \(on page 26\)](#)

[Creating and saving a pre-defined template in Template Editor \(on page 36\)](#)

[Opening and closing templates in Template Editor \(on page 38\)](#)

Creating and saving a pre-defined template in Template Editor

You can create a template that can be used as a pre-defined template for new templates. You can also save an existing template as a pre-defined template.

Do this when you often use the same kinds of templates for reports and summaries.

To...	Do this...
Create a new template to save as a pre-defined template.	<ol style="list-style-type: none"> 1. Create a new template, complete with all of the required components and objects, 2. To save it, click File > Template > Save.

To...	Do this...
	<ol style="list-style-type: none"> 3. Give the pre-defined template a name. This is also the file name. 4. Click OK to finish saving the template.
<p>Save an existing template as a pre-defined template.</p>	<ol style="list-style-type: none"> 1. Open the template you want to save as a pre-defined template. 2. Modify the template as needed. 3. To save it, click File > Template > Save. Give the pre-defined template a name. This is also the file name. 4. Click OK to finishing saving the pre-defined template.

Be sure to select where the pre-defined template is saved:

- **Global** - The pre-defined template is saved in the same folder as the tpled.exe program, which is common to all environments.
- **Local** - The pre-defined template is saved in the same folder as the tpled.ini file, which defines environment-specific settings.

Deleting a pre-defined template

If you do not need the pre-defined template anymore, you can delete it. Do the following:

1. Click **File > New**.
2. In the **Template type** dialog box that opens, select the pre-defined template you want to delete.
3. Click the red X on the right hand side of the dialog box and click **OK**.
4. Template Editor prompts you to accept or decline. Click **Yes** to delete the pre-defined template.

You cannot undo this operation.

See also

[Creating and saving a template in Template Editor \(on page 35\)](#)

Opening and closing templates in Template Editor

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.

If you open a template that was created with a previous version of Template Editor, a window opens warning that you cannot revert back to the previous version if you save it again. Click **OK** to accept.

3. To close a template file, click **File > Close**.

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 in Template Editor \(on page 35\)](#)

[Creating and saving a pre-defined template in Template Editor \(on page 36\)](#)

Modifying template page properties in Template Editor

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.

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.

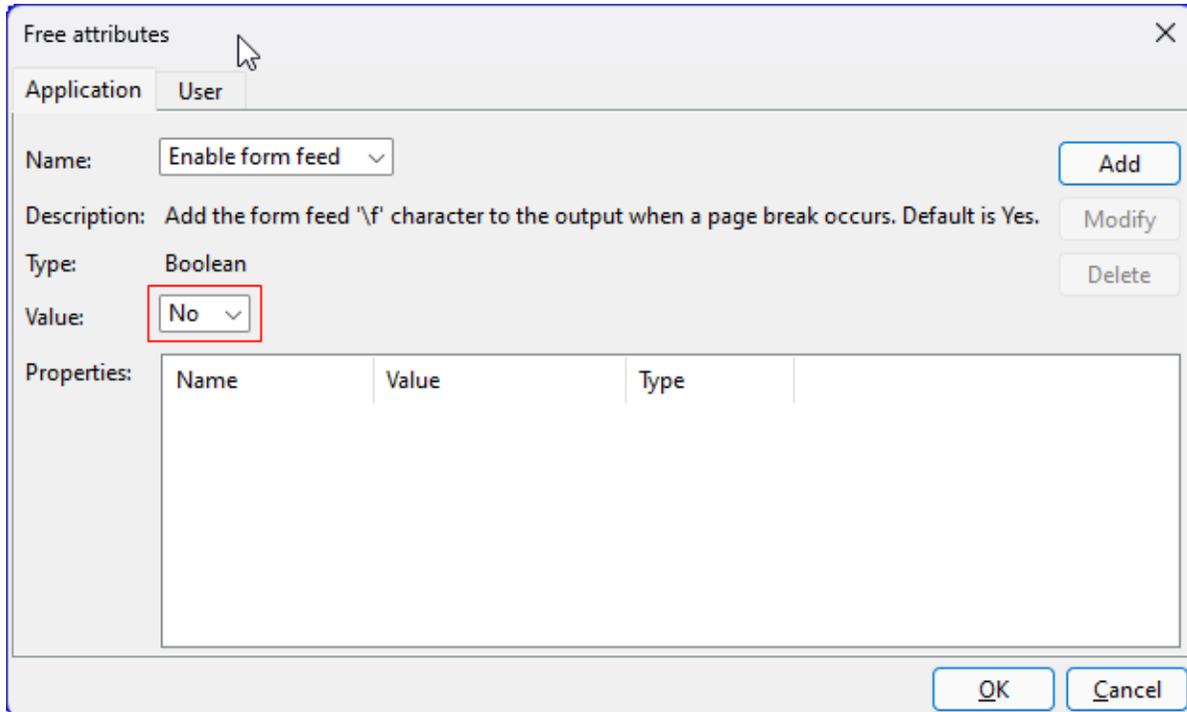
For a graphical template meant to be a map legend or label, these settings specify the total width and height of it.

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:
 - **Width** and **Height**: set in graphical units or characters. There is an option for using pagebreaks if the page height is exceeded. This is checked by default. If this is unchecked then no page breaks appear in the output.
 - **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.
 - **Dynamic height**: sets the output's size to fit its content. You can use the same template for different output sizes. This only works with graphical templates.
 - **Use font CAP height**: sets how the font size is calculated and output in graphical templates. It means that the text size will output exactly as defined.
 - **Output always as UTF8**: sets the coding for characters to always be output in UTF-8.
3. Click **OK** to save the changes.

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

There is now an option to remove the form feed character /f from output when a page break occurs during the output of a textual template. Click **Free Attributes**, which opens this dialog.



1. In the **Application** tab, select **Enable form feed** from the **Name** field.
2. Change the default for **Value** from **Yes** to **No**.
3. Click **Add**.
4. Click **OK** to save the changes.

See also

[Modifying column settings in Template Editor \(on page 75\)](#)

[Free attributes in Template Editor \(on page 131\)](#)

[Adding free attributes in Template Editor \(on page 131\)](#)

Changing the template type in Template Editor

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.

Chapter 5. Template Editor 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.

In this section:

[Inserting components and objects in Template Editor \(on page 42\)](#)

[Inserting a template page header or page footer \(on page 47\)](#)

[Selecting components and objects in Template Editor \(on page 48\)](#)

[Cutting, copying and pasting template components and objects \(on page 49\)](#)

[Modifying component and object properties in Template Editor \(on page 50\)](#)

[Tips for template component and object tasks \(on page 61\)](#)

Inserting components and objects in Template Editor

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.

If you are inserting a line, circle or rectangle, you can see the size of it on the **Status bar**.

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.

**Tip:**

If you double-click a button on 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.

Find out more:

[Drawing a polyline in Template Editor \(on page 43\)](#)

[Inserting a text object in Template Editor \(on page 44\)](#)

[Inserting a graphical field in Template Editor \(on page 44\)](#)

[Setting the visibility of texts, drawing objects and graphical fields in Template Editor \(on page 45\)](#)

See also

[Template rows \(on page 68\)](#)

[Value fields in Template Editor \(on page 85\)](#)

[Creating rows in Template Editor \(on page 68\)](#)

[Inserting a template page header or page footer \(on page 47\)](#)

[Tips for template component and object tasks \(on page 61\)](#)

[Creating a value field in Template Editor \(on page 85\)](#)

[Modifying component and object properties in Template Editor \(on page 50\)](#)

Drawing a polyline in Template Editor

To draw a polyline:

1. On the toolbar, click the **Polyline** button.
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 in Template Editor \(on page 50\)](#)

Inserting a text object in Template Editor

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. On the toolbar, click the **Text** button.

A dialog box appears prompting you to type in the text. You can type text on multiple lines. Press the **Enter** key to move to the next line.

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 \(on page 58\)](#)

[Modifying template text properties \(on page 57\)](#)

[Font properties for texts and value fields in Template Editor \(on page 66\)](#)

Inserting a graphical field in Template Editor

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

[Setting the visibility of texts, drawing objects and graphical fields in Template Editor \(on page 45\)](#)

[Modifying the size of a template component or object \(on page 52\)](#)

[Modifying the name of a template component or object \(on page 51\)](#)

Setting the visibility of texts, drawing objects and graphical fields in Template Editor

You can use a rule to set the visibility of texts, drawing objects or graphical fields during output. Set the visibility of these objects when you are inserting a new ones, or you can modify existing texts, drawing objects or graphical fields.

Do one of the following:

To...	Do this...
<p>Set the text, drawing object or graphical field to always be visible</p>	<ol style="list-style-type: none"> 1. Insert a new text, drawing object or graphical field or double-click an existing one to open the properties dialog box. 2. Click Always visible. This means the text, drawing object or graphical field will always be visible during template output. 3. Click OK to accept the changes and close the dialog box.
<p>Set the text, drawing object or graphical field to be output under certain conditions</p>	<ol style="list-style-type: none"> 1. Insert a new text, drawing object or graphical field or double-click an existing one to open the properties dialog box. 2. Click Visible. This means the text, drawing object or graphical field is only visible according to the rule that is applied to it. 3. Click Rule to open the Rule Contents dialog box. 4. Add the rule and click OK. The result should be TRUE (output) or FALSE (not output) when Visible is selected. <p>This rule below means the graphical field is output when the value for SHAPE_INTERNAL is equal to 1.</p>

To...	Do this...
	<div data-bbox="657 310 1224 506" style="border: 1px solid gray; padding: 5px;"> <p>Rule</p> <pre>if (GetValue("SHAPE_INTERNAL") == 1) then true else false endif</pre> </div> <p data-bbox="625 552 1224 632">5. Click OK to accept the changes and close the dialog box.</p>

See also

[Inserting a graphical field in Template Editor \(on page 44\)](#)

[Modifying template text properties \(on page 57\)](#)

Inserting a template 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. On the toolbar, click the page header or page footer button.

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. You can specify if a page header or page footer should use columns by clicking the **Use columns** check box.
 4. 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 \(on page 51\)](#)

[Modifying the size of a template component or object \(on page 52\)](#)

[Column settings in Template Editor \(on page 72\)](#)

Selecting components and objects in Template Editor

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**.



Note:

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

See also

[Special drawing operations in Template Editor \(on page 62\)](#)

Cutting, copying and pasting template 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 in Template Editor \(on page 50\)](#)

Duplicating objects in Template Editor

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 template components and objects \(on page 49\)](#)

Modifying component and object properties in Template Editor

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**.

Find out more:

[Modifying the name of a template component or object \(on page 51\)](#)

[Modifying the size of a template component or object \(on page 52\)](#)

[Moving a template component \(on page 52\)](#)

[Cropping template components \(on page 53\)](#)

[Changing the template component type \(on page 53\)](#)

[Grouping or ungrouping template objects \(on page 53\)](#)

[Moving a template object \(on page 54\)](#)

[Trimming template objects in Template Editor \(on page 55\)](#)

[Extending template objects in Template Editor \(on page 56\)](#)

[Formatting template objects with format painter in Template Editor \(on page 56\)](#)

[Modifying template text properties \(on page 57\)](#)

[Setting default properties for new template objects \(on page 58\)](#)

[Rotating template objects \(on page 59\)](#)

[Aligning template objects \(on page 60\)](#)

[Aligning template objects to the grid \(on page 60\)](#)

[Trimming template object locations \(on page 60\)](#)

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 \(on page 52\)](#)

[Modifying a value field formula in Template Editor \(on page 100\)](#)

Modifying the size of a template component or object

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

Do the following:

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. If you hold down the Shift key while you drag, it will retain the aspect ratio. Note that dragging the handles of polylines, polygons and free form areas is the only way to resize them.

When you resize template objects, you can see the measures of the objects on the **Status bar**.

See also

[Modifying the name of a template component or object \(on page 51\)](#)

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 on the **Sorting and Drawing** toolbar.

You can also move a template component by selecting it and dragging with the mouse.

3. Save the template.

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

See also

[Template row output properties \(on page 69\)](#)

[Value field output properties in Template Editor \(on page 86\)](#)

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 Editor components \(on page 16\)](#)

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 in Template Editor \(on page 69\)](#)

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.



Note:

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 in Template Editor \(on page 42\)](#)

[Modifying the size of a template component or object \(on page 52\)](#)

[Moving a template object \(on page 54\)](#)

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. Press **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**. 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, cut and paste it.

See also

[Offset for template objects \(on page 144\)](#)

[Selecting components and objects in Template Editor \(on page 48\)](#)

Trimming template objects in Template Editor

Trim template objects such as lines when you want to modify them quickly and accurately. Trimming a line object will align it with the selected line.

Trimming only applies to line objects.

Do the following:

1. Click the line you want to target.
2. Click the trim button on the **Format** toolbar.
3. Click the line to trim to the target line.

You can also make an area selection to select and trim many lines simultaneously.

The line is trimmed by the difference between the target line and the line you select to trim.

See also

[Extending template objects in Template Editor \(on page 56\)](#)

[Formatting template objects with format painter in Template Editor \(on page 56\)](#)

Extending template objects in Template Editor

Extend template objects such as lines when you want to modify them quickly and accurately. Extending a line object will join the selected lines to join others.

This operation works on line objects only.

Do the following:

1. Click the line you want to target.
2. Click the extend button on the **Format** toolbar.
3. Click the line to extend the target.

You can also make an area selection and extend many lines simultaneously.

See also

[Trimming template objects in Template Editor \(on page 55\)](#)

[Formatting template objects with format painter in Template Editor \(on page 56\)](#)

Formatting template objects with format painter in Template Editor

Format template objects such as lines when you want to modify them quickly and accurately.

The format painter does not work on template objects like groups, pictures, symbols or graphical fields.

Do the following:

1. Click the line whose properties you want to use.
2. Click the paintbrush button on the **Format** toolbar.
3. Click the line(s) or other object(s) you want to format with the same properties.

Make an area selection if you want to format several template objects simultaneously.

See also

[Trimming template objects in Template Editor \(on page 55\)](#)

[Extending template objects in Template Editor \(on page 56\)](#)

Modifying template text properties

Text properties can be modified based on what you want to show in a report. Double-click a text object in the template in order to open the **Text Properties** dialog box.

Property	How to modify it
Text	The text inserted in the template component. You can modify it in this field.
Always visible / Visible	You can use a rule to set the visibility of a text during output. You can set the text to always be visible during output, or apply a rule to the text so that it appears under certain output conditions.
Name	A unique name for the text, it does not have to be the same as the text inserted into the template component.
Font	The font applied to the text object. Click ... to open the Font Properties dialog box.
Line spacing	The line spacing for the text. NOTE: This setting is only for graphical templates. Choose from the list: Single, 1.15, 1.5 or Double
Justify	The alignment of the text in the template. Choose from the list: Left, Center, Right.
Angle	The angle (in degrees) that is applied to the text object. It can be positive or negative.

Property	How to modify it
Border(s)	<p>NOTE: This only applies to textual templates that are output as Microsoft Excel files. Apply borders to the text object. You can apply them to any side of the text object by clicking the checkbox. You can choose from: No border, Thin, Thick or Double.</p>
Set as default for new texts	Select the check box to save the settings applied to all new texts.

Click **Apply** to modify the changes and keep the dialog open, or click **OK** to accept the changes and close the dialog. Click **Cancel** if you do not want to retain any of the changes.

See also

[Inserting a text object in Template Editor \(on page 44\)](#)

[Font properties for texts and value fields in Template Editor \(on page 66\)](#)

[Setting the visibility of texts, drawing objects and graphical fields in Template Editor \(on page 45\)](#)

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 in Template Editor \(on page 42\)](#)

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 in Template Editor \(on page 44\)](#)

[Adding symbols to a template in Template Editor \(on page 108\)](#)

[Creating a value field in Template Editor \(on page 85\)](#)

[Modifying component and object properties in Template Editor \(on page 50\)](#)

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.

You can also click the related button on the **Align** toolbar.

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 in Template Editor \(on page 29\)](#)

[Activating the grid in Template Editor \(on page 28\)](#)

[Showing or hiding the grid in Template Editor \(on page 29\)](#)

Trimming template object locations

Trim the location of template objects such as imported DXF or DWG images when you want to place them more accurately inside a template component.

1. Click the template object you want to target.

You can also trim the location of multiple objects by selecting them simultaneously.

2. Click **Edit > Trim location**.

Template Editor places the object so that X and Y coordinates are rounded to the closest millimeter (i.e 157.00) or 1/32 of an inch depending on what environment you are using. Check the **Preview pane** to see the results.

Tips for template component and object tasks

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

Find out more:

[Output options for template headers and footers \(on page 61\)](#)

[Special drawing operations in Template Editor \(on page 62\)](#)

[Snapping line objects in Template Editor \(on page 63\)](#)

[Drawing with the numeric pad in Template Editor \(on page 64\)](#)

[Drawing with the grid in Template Editor \(on page 64\)](#)

[Viewing or hiding template objects \(on page 65\)](#)

[Using the Selection toolbar \(on page 65\)](#)

[Font properties for texts and value fields in Template Editor \(on page 66\)](#)

[Searching for text in Template Editor \(on page 67\)](#)

See also for additional helpful information

[Drawing a polyline in Template Editor \(on page 43\)](#)

[Setting presentation file and pen settings in Template Editor \(on page 33\)](#)

[Output options for templates created with Template Editor \(on page 136\)](#)

Output options for template 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

[Template row output properties \(on page 69\)](#)

[Outputting a template created with Template Editor \(on page 135\)](#)

Special drawing operations in Template Editor

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	<ol style="list-style-type: none"> 1. Click Options > Continuous Action to turn the option on. 2. Click the toolbar button 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 button 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 in Template Editor \(on page 48\)](#)

Snapping line objects in Template Editor

To simplify drawing operations in Template Editor, you can drag line objects to snap to other drawing objects such as lines, rectangles and grouped objects (i.e. lines), where they intersect or to a corner.

Do the following:

1. Click the line you want to drag and the handles appear.
2. Click the handle with your mouse and drag the line to the desired point.

To ensure the line has snapped to the right location, zoom in and make any needed modifications.

If the grid snapping settings are active, you may notice that lines will snap to the grid.

See also

[Activating the grid in Template Editor \(on page 28\)](#)

Drawing with the numeric pad in Template Editor

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. On the toolbar, click the **Line** button 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 on 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 in Template Editor \(on page 42\)](#)

Drawing with the grid in Template Editor

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 in Template Editor \(on page 29\)](#)

[Showing or hiding the grid in Template Editor \(on page 29\)](#)

[Activating the grid in Template Editor \(on page 28\)](#)

[Showing or hiding the grid in Template Editor \(on page 29\)](#)

[Aligning template objects to the grid \(on page 60\)](#)

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

[Using the Selection toolbar \(on page 65\)](#)

[Modifying Template Editor interface components \(on page 21\)](#)

[Hiding template rows from output in Template Editor \(on page 75\)](#)

[Hiding value fields from output in Template Editor \(on page 95\)](#)

Using the Selection toolbar

You can click the **Selection** toolbar buttons to toggle template objects to be selectable or not. This is useful, for example, if you need to change all of the line objects to a different color or width.

Do the following:

1. Click the toolbar buttons of the template objects you want to exclude.

If you make an area selection, for example, and have excluded value fields, everything else will be selected.

2. Modify the selected objects as needed.

If you select objects of different types (i.e. lines and value fields), you cannot modify the properties of both at the same time.

3. Save the changes.

Toggle the buttons on and off as needed in order to modify template objects.

See also

[Viewing or hiding template objects \(on page 65\)](#)

Font properties for texts and value fields in Template Editor

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 a text or a value field 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 Locations preferences dialog) 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.

Property	Description
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 in Template Editor \(on page 50\)](#)

[File Locations preferences in Template Editor \(on page 26\)](#)

[Modifying general preferences in Template Editor \(on page 23\)](#)

[Setting presentation file and pen settings in Template Editor \(on page 33\)](#)

Searching for text in Template Editor

You can search for text in the template, which also includes texts in row rules and value field formulas.

Search for text in the template 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.

Chapter 6. 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.

Find out more:

[Creating rows in Template Editor \(on page 68\)](#)

[Content types in Template Editor \(on page 69\)](#)

[Template row output properties \(on page 69\)](#)

[Column settings in Template Editor \(on page 72\)](#)

[Modifying the content type of a row in Template Editor \(on page 73\)](#)

[Modifying the sort type of a row in Template Editor \(on page 74\)](#)

[Modifying column settings in Template Editor \(on page 75\)](#)

[Hiding template rows from output in Template Editor \(on page 75\)](#)

[Modifying template row hierarchy \(on page 76\)](#)

[Controlling row output with rules in Template Editor \(on page 76\)](#)

Creating rows in Template Editor

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. On the toolbar, click the **Row** button.
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 in Template Editor \(on page 69\)](#)

[Column settings in Template Editor \(on page 72\)](#)

[Creating and saving a template row rule \(on page 77\)](#)

[Creating a template row rule with Rule Wizard \(on page 78\)](#)

Content types in Template Editor

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 \(on page 68\)](#)

[Modifying the content type of a row in Template Editor \(on page 73\)](#)

[Available content types in Tekla Structures \(external link\)](#)

Template 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	<p>Set the sort type of the row in order to determine the final output of rows. Choose from None, Combine or Distinct.</p> <p>If you choose Combine, this results in numerous options for value field output.</p>
Output	Hide or show the row in the final output.
Use columns	Use columns to make better use of the space in a template for output. Click Edit... to set the number of columns.
Background color	<p>NOTE: Only in textual templates and when a template is exported to Excel. Check this option when you want to use color to highlight the row in template output. A dialog opens allowing you to select the color from the color chooser or you can create your own custom color. You can use this, for example, to highlight a summary row to provide contrast in the output.</p>

Property	Description
Use alternate background	NOTE: Only in graphical templates. Check this option when you want to use color to highlight the row in template output. A dialog opens allowing you to select the color from the color chooser or you can create your own custom color. You can use this, for example, to highlight a summary row or every second row to provide contrast in the output.
Row border(s)	NOTE: This only applies to textual templates that are output as Microsoft Excel files. Apply borders to the row that will appear in output. You can apply them to any side of the row by clicking the checkboxes. You can choose from: No border, Thin, Thick or Double .
Rule	Set a rule on the row to define the information it shows during output.

See also

[Creating rows in Template Editor \(on page 68\)](#)

[Content types in Template Editor \(on page 69\)](#)

[Free attributes in Template Editor \(on page 131\)](#)

[Modifying the content type of a row in Template Editor \(on page 73\)](#)

[Modifying the sort type of a row in Template Editor \(on page 74\)](#)

[Modifying column settings in Template Editor \(on page 75\)](#)

[Hiding template rows from output in Template Editor \(on page 75\)](#)

[Creating and saving a template row rule \(on page 77\)](#)

[Row rule content properties in Template Editor \(on page 79\)](#)

Column settings in Template Editor

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.
Start from	<p>Sets the starting point for column distribution. Setting this helps eliminate white space and influences the visual output of templates.</p> <p>Click the list to choose the starting location:</p> <ul style="list-style-type: none"> • Top left • Top right • Bottom left • Bottom right
Fill direction	Fill direction determines in which direction the columns are filled:

Field	Description
	<ul style="list-style-type: none"> • Horizontal: fills the columns first to the right, then down. • Vertical: fills the columns first down, then to the right.
Fill policy	<p>Fill policy determines how the columns are filled in the template:</p> <ul style="list-style-type: none"> • Even: makes column heights equal. • Continuous: fills a column before proceeding to the next one, results in uneven column heights.

**Tip:**

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 in Template Editor \(on page 75\)](#)

Modifying the content type of a row in Template Editor

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 Locations** 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 in Template Editor \(on page 74\)](#)

[Free attributes in Template Editor \(on page 131\)](#)

[File Locations preferences in Template Editor \(on page 26\)](#)

Modifying the sort type of a row in Template Editor

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 in Template Editor \(on page 84\)](#)

[Setting the sort order of a value field in Template Editor \(on page 91\)](#)

[Setting the priority of a value field in Template Editor \(on page 91\)](#)

[Modifying the content type of a row in Template Editor \(on page 73\)](#)

[Hiding template rows from output in Template Editor \(on page 75\)](#)

Modifying column settings in Template Editor

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 in Template Editor \(on page 72\)](#)

Hiding template rows from output in Template Editor

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 in Template Editor \(on page 68\)](#)

[Content types in Template Editor \(on page 69\)](#)

[Modifying the sort type of a row in Template Editor \(on page 74\)](#)

[Modifying template row hierarchy \(on page 76\)](#)

Modifying template 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 button.
2. Insert other rows.

On the **Sorting and drawing order** toolbar above the **Content Browser**, click the button to shift the row down a level.

3. You can also drag the row to the new location in the **Content Browser**.
4. Click **Save** to retain the changes.

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 in Template Editor \(on page 68\)](#)

[Hierarchical listings created with Template Editor \(on page 138\)](#)

Controlling row output with rules in Template Editor

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).

**Note:**

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 in the **Rule Wizard** dialog box or the **Rule Contents** dialog box.

Find out more:

[Creating and saving a template row rule \(on page 77\)](#)

[Creating a template row rule with Rule Wizard \(on page 78\)](#)

[Row rule content properties in Template Editor \(on page 79\)](#)

[Modifying a template row rule \(on page 81\)](#)

[Overriding template object colors with row rules in Template Editor \(on page 82\)](#)

[Pre-defined row rules in Template Editor \(on page 83\)](#)

[Duplicate lines in Template Editor \(on page 84\)](#)

Creating and saving a template row rule

You can create more complex rules in 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 template row rule with Rule Wizard \(on page 78\)](#)

[Modifying a template row rule \(on page 81\)](#)

[Controlling row output with rules in Template Editor \(on page 76\)](#)

[Pre-defined row rules in Template Editor \(on page 83\)](#)

Creating a template row 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 to open the **Rule Wizard**.

It is used for filtering content in rows. The conditional structure is already a part of the rule in the **Rule Wizard**.

3. Select the **Attribute** to which the rule refers.
4. Add the conditions which apply to the rule by selecting from the list.
5. Type in the **Value**. If it is a string, enclose it in quotation marks "".
6. Click **OK** to save the changes.
7. 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 template row rule \(on page 77\)](#)

[Controlling row output with rules in Template Editor \(on page 76\)](#)

Row rule content properties in Template Editor

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

Property	Description
Conditional structure	The conditional structure applied to row rules is like the if-else statement in any programming language.
Control	Lists the control functions you can apply.

Property	Description
Math	Lists mathematical operations you can apply.
String	Lists string operations that you can apply.
Attribute	Lists attribute functions for getting data from the product database. Click Select to open a more extensive attribute list and click More to use advanced options.
Value field functions	Lists value field functions that can calculate sums or counts of value fields of this template. Click Select to open a more extensive attribute list.
Advanced options	<p>Click Select to open a list of advanced options to apply to the formula. Type in the option you are searching for and use the arrows to scroll through the list.</p> <p>The Description provides some information about the advanced option, but click More to open the Tekla User Assistance site and learn more.</p>
Pre-defined rule	<p>Apply a pre-defined rule to the row.</p> <p>Rules are stored in a pre-defined file (extension .pdr). The file is specified in the File Locations preferences. These rules can be freely added (Store) and deleted (Remove).</p>
Rule	Functions and parameters appear in this text field. The maximum length of this field is 4000

Property	Description
	characters. Modify the rule by cutting, copying and pasting text in the field itself.

See also

[Creating and saving a template row rule \(on page 77\)](#)

[Creating a template row rule with Rule Wizard \(on page 78\)](#)

[Adding comments to row rules and value field formulas in Template Editor \(on page 144\)](#)

[Modifying a template row rule \(on page 81\)](#)

[Conditional structure for row rules and formulas in Template Editor \(on page 145\)](#)

[Control functions in Template Editor \(on page 149\)](#)

[Trigonometric functions in Template Editor \(on page 173\)](#)

[Statistical functions in Template Editor \(on page 174\)](#)

[Attribute functions in Template Editor \(on page 152\)](#)

[Value field functions in Template Editor \(on page 154\)](#)

[String operations in Template Editor \(on page 159\)](#)

[Pre-defined row rules in Template Editor \(on page 83\)](#)

Modifying a template 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 template row rule \(on page 77\)](#)

[Creating a template row rule with Rule Wizard \(on page 78\)](#)

[Pre-defined row rules in Template Editor \(on page 83\)](#)

[Content types in Template Editor \(on page 69\)](#)

Overriding template object colors with row rules in Template Editor

Override the colors of text objects, value fields and drawing objects in template rows when you want those objects to stand out in certain situations.

Choose from several colors that you can use in Template Editor. Valid color names are black, white, red, green, blue, cyan, yellow, magenta, brown, darkgreen, navyblue, greenblue, orange and gray. In addition you can use difference shades of gray by name in the row rules: Gray30, Gray50, Gray70 and Gray90.

Do the following:

Numeric id values for colors are also accepted as defined in extended_colors.map colormap file.

1. Select the row to which you want to apply the rule and double-click to open the **Row Properties** dialog box.
2. Click **Advanced** to open the **Rule Contents** dialog box.
3. Create the row rule. Use the **Output** control function with the following syntax in the row rule to define the color override.

```
Output("#<key>=<color>;")
```

The following abbreviations are keys that refer to the template objects in question:

- VFC (value field color)
- TXC (text color)
- DRC (drawing object color)

You can combine these values in one row rule using a semicolon as a separator.

4. Click **Check** to ensure the rule syntax is correct. Make corrections if necessary.
5. Click **OK** to save the rule and close the **Rule Contents** dialog.
6. Click **OK** to close the **Row Properties** dialog.
7. Save the template.

Here is a row rule example:

```

Rule
if (GetValue("LENGTH") > 1000) then
  Output("#VFC=ORANGE;TXC=YELLOW;DRC=BLUE")
else
  Output("#VFC=BLUE;COLOR=GREEN")
endif

```

Result: This rule overrides the colors when the **LENGTH** attribute is longer than 1000. Value fields are orange, text objects are yellow and the drawing color is blue (i.e. lines are blue).

See also

[Output \(on page 150\)](#)

Pre-defined row rules in Template Editor

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 Locations** preferences. These rules can be freely added and deleted in the **Rule Contents** dialog box.

See also

[File Locations preferences in Template Editor \(on page 26\)](#)

[Creating and saving a template row rule \(on page 77\)](#)

[Modifying a template row rule \(on page 81\)](#)

Duplicate lines in Template Editor

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 in Template Editor \(on page 74\)](#)

Chapter 7. Value fields in Template Editor

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.

Find out more:

[Creating a value field in Template Editor *\(on page 85\)*](#)

[Value field output properties in Template Editor *\(on page 86\)*](#)

[Setting the sort order of a value field in Template Editor *\(on page 91\)*](#)

[Setting the priority of a value field in Template Editor *\(on page 91\)*](#)

[Setting the data type of a value field in Template Editor *\(on page 92\)*](#)

[Assigning value field meanings in Template Editor *\(on page 93\)*](#)

[Setting the layout of value fields in Template Editor *\(on page 94\)*](#)

[Resizing value field fonts automatically in Template Editor *\(on page 94\)*](#)

[Hiding value fields from output in Template Editor *\(on page 95\)*](#)

[Formatting value fields in Template Editor *\(on page 96\)*](#)

[Aligning value field texts in output in Template Editor *\(on page 96\)*](#)

[Value field formulas in Template Editor *\(on page 97\)*](#)

[Tips for value fields in Template Editor *\(on page 104\)*](#)

Creating a value field in Template Editor

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. On the toolbar, click the **Value Field** button.
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. You can also search for attributes if needed.

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 a default value.

5. Click **OK**.

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

Formulas are an essential part of a value field since they define what is gathered and shown in a value field. A formula, for example, can get data from the product database and produce summaries of data that is gathered with other value fields.

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

[Value field output properties in Template Editor \(on page 86\)](#)

[Creating and saving a formula in Template Editor \(on page 97\)](#)

[Global attributes and Template Editor \(on page 134\)](#)

[Template definition files \(on page 18\)](#)

Value field output properties in Template Editor

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.

Be aware that there are many different combinations of settings that can be applied to value fields. Some also depend on the properties applied to rows.

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 in which it appears.
Line count	<p>You can add multiple lines of text to a value field, for example, if it has a long description. With multiple lines, you can align the lines to Top, Middle or Bottom. Click your preference in the Align list in the Layout section of the dialog.</p> <p>Outputting multiple lines with a value field requires that the Line count is larger than 1, and that the output data for the value field is either very long (resulting in automatic line breaks) or contains substrings with <code>\n</code> (forcing a line break).</p> <p>Even if you do not use multiple lines, the space will be reserved.</p> <p>NOTE: The height of the row will limit the number of lines you can use.</p>
Data type	<p>There are three options:</p> <ul style="list-style-type: none"> • Number (an integer) • Number with decimals (set the number of decimals in the Decimal list) • Text (plain text)

Property	Description
Output 0 as an empty string	Select the check box to output the value field with a value of zero as an empty string.
Hide in output	Select the check box to exclude this value field from output.
Meaning	Defines the meaning of the value field (the format) 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 this list.
Decimals	When you have specified the Data type as Number with decimals , this field is active. You can set how many decimal places are in the number.
Order	<p>Choose the order for value field output: None, Ascending, Descending or Application (Trimble product).</p> <p>The value field output order is also determined by the sort type applied to the row in the Row Properties dialog box. The sort type for the row must be defined in order to use this option.</p> <p>When rows are combined, there are several options:</p> <ul style="list-style-type: none"> • Don't sum values - Nothing is summed. • Sum values across all rows - All values across all rows are summed. • Sum values within one row - The values for the selected row are summed.

Property	Description
	<p>For value fields that have the Data type defined as Text, a separator can be defined. You can also define the last separator individually. The results is that unique text values in value fields can now be combined into a single text value when rows are combined.</p> <p>If the Data type is number or number with decimals then those number are summed and the result is a single number.</p>
Layout	<p>Modify the settings related to the layout of the value field:</p> <ul style="list-style-type: none"> • Font: Click ... to open a list of the fonts you can use. • Justify: Choose the alignment you want to apply to the text in the value field - Left, Center or Right. • Line spacing: Choose line spacing for the text - Single, 1.15, 1.5 or Double • Length: Set the length of the value field in characters. • Align: This setting is active only when the value field has multiple lines. The text is aligned per line. Choose from Top, Middle or Bottom. • Angle: Choose a positive or negative angle to apply to the value field.
Border(s)	<p>NOTE: This only applies to textual templates that are output as Microsoft Excel files. Apply borders to the value field. You can apply them to any side of the value field by clicking the checkboxes. You</p>

Property	Description
	can choose from: No border, Thin, Thick or Double .
Set as default for new value fields	Select the check box to save the settings applied to all new value fields.

See also

[Value field meanings in Template Editor \(on page 90\)](#)

[Modifying a value field formula in Template Editor \(on page 100\)](#)

[Formula content properties in Template Editor \(on page 98\)](#)

[Modifying the name of a template component or object \(on page 51\)](#)

[Setting the data type of a value field in Template Editor \(on page 92\)](#)

[Hiding value fields from output in Template Editor \(on page 95\)](#)

[Outputting blank value fields in Template Editor \(on page 104\)](#)

[Assigning value field meanings in Template Editor \(on page 93\)](#)

[Modifying the sort type of a row in Template Editor \(on page 74\)](#)

[Setting default properties for new template objects \(on page 58\)](#)

https://support.tekla.com/topic-redirect/tekla-structures/rep_tips_for_templates

Value field meanings in Template Editor

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 (`valuefieldclasses.lst`) is set in **File Locations** preferences dialog. You can open this file to see a list of valid unit and precision strings, which is located in the `..\Program Files`

`\Tekla Structures\ folder. NOTE: Do not modify the contents of this file.`

See also

[Assigning value field meanings in Template Editor \(on page 93\)](#)

[File Locations preferences in Template Editor \(on page 26\)](#)

Setting the sort order of a value field in Template Editor

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** (set by the Trimble 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 in Template Editor \(on page 91\)](#)

[Setting the layout of value fields in Template Editor \(on page 94\)](#)

Setting the priority of a value field in Template Editor

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.

See also

[Setting the sort order of a value field in Template Editor \(on page 91\)](#)

[Setting the data type of a value field in Template Editor \(on page 92\)](#)

Setting the data type of a value field in Template Editor

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 to 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 in Template Editor \(on page 86\)](#)

[Setting the priority of a value field in Template Editor \(on page 91\)](#)

[Setting the layout of value fields in Template Editor \(on page 94\)](#)

Assigning value field meanings in Template Editor

Before you start, ensure that you have set the path for value field meanings file (.lst) in the **File Locations** preferences dialog box.

Value field meanings are formats for value fields applied at run-time. The list of available meanings and how to apply them is product-specific and can be very numerous, for example: **Percentage, Angle, Area, Length, Volume, Weight** and so on.

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 Locations preferences in Template Editor \(on page 26\)](#)

[Setting the sort order of a value field in Template Editor \(on page 91\)](#)

[Setting the priority of a value field in Template Editor \(on page 91\)](#)

[Setting the data type of a value field in Template Editor \(on page 92\)](#)

[Setting the layout of value fields in Template Editor \(on page 94\)](#)

Setting the layout of value fields in Template Editor

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 in Template Editor \(on page 86\)](#)

Resizing value field fonts automatically in Template Editor

Modify attribute settings for value fields so that font sizes will be resized automatically if the value field content is longer than the length of the value field. By automatically resizing fonts, value fields can contain more content.

Do the following:

1. Select the value field you want to modify and double-click to open the **Value Field Properties** dialog box.
2. Click the **Free attributes** button to open the **Free attributes** dialog box.
3. In the **Application** tab, click the **Name** list and select from the following options:

- **Min. font size** - The font will not be resized under any value you have specified. Be aware that all the content might not fit in the value field in this case.
 - **Font size change** - The font is resized by the value you have defined (i.e. 0,2), reducing it by that amount until it fits in the value field or the minimum font size is reached.
 - **List of font sizes** - The font is resized based on the list of values given until the right size font is found or the minimum font size is reached. You can decide what values to use. The separator in the list a semi-colon.
4. Click **Add**.
- If you define both **Font size change** and **List of font sizes**, **List of font sizes** will override the former.
5. Click **OK** to close the **Free Attributes** dialog box.
 6. Click **OK** to close the **Value Field Properties** dialog box.

When the template is output the font will be resized to settings you specified.

See also

[Free attributes in Template Editor \(on page 131\)](#)

Hiding value fields from output in Template Editor

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 in Template Editor \(on page 91\)](#)

[Setting the priority of a value field in Template Editor \(on page 91\)](#)

Formatting value fields in Template Editor

Before you start, ensure that you have set the path for value field meanings in the **File Locations** 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 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 Locations preferences in Template Editor \(on page 26\)](#)

[Modifying File Locations preferences in Template Editor \(on page 27\)](#)

[Setting the data type of a value field in Template Editor \(on page 92\)](#)

[Setting the layout of value fields in Template Editor \(on page 94\)](#)

Aligning value field texts in output in Template Editor

Before you start, ensure that you have set the path for attributes in the **File Locations** 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 in Template Editor \(on page 85\)](#)

[Hiding value fields from output in Template Editor \(on page 95\)](#)

Value field formulas in Template Editor

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.

Find out more:

[Creating and saving a formula in Template Editor \(on page 97\)](#)

[Formula content properties in Template Editor \(on page 98\)](#)

[Modifying a value field formula in Template Editor \(on page 100\)](#)

[Get a single piece of data from the database in Template Editor \(on page 101\)](#)

[Creating a listing of template objects from the database \(on page 102\)](#)

[Creating a summary field in Template Editor \(on page 103\)](#)

[Outputting blank value fields in Template Editor \(on page 104\)](#)

Creating and saving a formula in Template Editor

Formulas determine what 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.



Note:

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 in Template Editor \(on page 152\)](#)

[Control functions in Template Editor \(on page 149\)](#)

[Mathematical operations in Template Editor \(on page 171\)](#)

[Trigonometric functions in Template Editor \(on page 173\)](#)

[Statistical functions in Template Editor \(on page 174\)](#)

[Data type conversion functions in Template Editor \(on page 168\)](#)

[String operations in Template Editor \(on page 159\)](#)

[Miscellaneous functions in Template Editor \(on page 177\)](#)

Formula content properties in Template Editor

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** box.

Property	Description
Conditional structure	The conditional structure applied to value field formulas is like the if-else statement in any programming language.
Control	Lists the control functions you can apply.
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. Click Select to open a more extensive attribute list and click More to use advanced options.
Value field functions	Lists value field functions that can calculate sums or counts of value fields of this template. Click Select to open a more extensive attribute list.
Advanced options	<p>Click Select to open a list of advanced options to apply to the formula. Type in the option you are searching for and use the arrows to scroll through the list.</p> <p>The Description provides some information about the advanced option, but click More to open the Tekla User Assistance site and learn more.</p>
Pre-defined formula	<p>Apply a pre-defined formula to the value field.</p> <p>Formulas are stored in a pre-defined file (extension .pdr). The file is specified in the File Locations preferences dialog box. These formulas can be freely added (Store) and deleted (Remove).</p>

Property	Description
Formula	Functions and parameters appear in this box. The maximum length of this field is 4000 characters. Modify the formula by cutting, copying and pasting text in the field itself.

See also

[Creating and saving a formula in Template Editor \(on page 97\)](#)

[Adding comments to row rules and value field formulas in Template Editor \(on page 144\)](#)

[Modifying a value field formula in Template Editor \(on page 100\)](#)

[Conditional structure for row rules and formulas in Template Editor \(on page 145\)](#)

[Control functions in Template Editor \(on page 149\)](#)

[Trigonometric functions in Template Editor \(on page 173\)](#)

[Statistical functions in Template Editor \(on page 174\)](#)

[Attribute functions in Template Editor \(on page 152\)](#)

[Value field functions in Template Editor \(on page 154\)](#)

[String operations in Template Editor \(on page 159\)](#)

[Pre-defined row rules in Template Editor \(on page 83\)](#)

Modifying a value field formula in Template Editor

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 the appropriate 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.

**Note:**

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

[Value field output properties in Template Editor \(on page 86\)](#)

[Get a single piece of data from the database in Template Editor \(on page 101\)](#)

[Creating a listing of template objects from the database \(on page 102\)](#)

[Template Editor formula and rule reference \(on page 143\)](#)

[Attribute functions in Template Editor \(on page 152\)](#)

[Control functions in Template Editor \(on page 149\)](#)

Get a single piece of data from the database in Template Editor

Before you get started ensure you have defined the global attributes file path in the **File Locations** 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 click **OK**.
5. Click **OK** to close the dialog box. The changes to the formula are saved.

See also

[Creating a value field in Template Editor \(on page 85\)](#)

[Creating and saving a formula in Template Editor \(on page 97\)](#)

[Creating a listing of template objects from the database \(on page 102\)](#)

[GetValue \(on page 153\)](#)

[PreviousValue \(on page 154\)](#)

[NextValue \(on page 153\)](#)

Creating a listing of template objects from the database

Before you get started ensure you have defined the global attributes file path in the **File Locations** 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 in Template Editor \(on page 85\)](#)

[Get a single piece of data from the database in Template Editor \(on page 101\)](#)

[Modifying File Locations preferences in Template Editor \(on page 27\)](#)

[GetValue \(on page 153\)](#)

Creating a summary field in Template Editor

When you want to 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. Ensure 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 in Template Editor \(on page 74\)](#)

[Get a single piece of data from the database in Template Editor \(on page 101\)](#)

[Creating a listing of template objects from the database \(on page 102\)](#)

[Value field functions in Template Editor \(on page 154\)](#)

Outputting blank value fields in Template Editor

If a value field has a final value of zero, you can exclude it from output in a report.

1. Select and double-click the value field to open the **Value Field Properties** dialog box
2. Select **Output 0 as empty string**.
3. Click **OK** to save the changes.

As a result, nothing will be output.

See also

[Value field output properties in Template Editor \(on page 86\)](#)

Tips for value fields in Template Editor

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.

Find out more:

[String translations of texts in Tekla Structures \(on page 105\)](#)

[Setting Tekla Structures advanced options for value fields \(on page 105\)](#)

See also

[Duplicate lines in Template Editor \(on page 84\)](#)

[Offset for template objects \(on page 144\)](#)

[Rotating template objects \(on page 59\)](#)

String translations of texts in Tekla Structures

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.

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("XS_ADVANCED_OPTION_NAME")
```

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

The reason for the error indicating unknown attributes could be due to a spelling error (if you have typed it in) or because the advanced options are not listed in the `contentattributes_global.lst` or `contentattributes_userdefined.lst` files.

That being said, errors related to unknown attributes are rare.

Chapter 8. Symbols and pictures in Template Editor

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.

Find out more:

[Symbol and picture libraries in Template Editor \(on page 107\)](#)

[Adding symbols to a template in Template Editor \(on page 108\)](#)

[Adding pictures to a template in Template Editor \(on page 108\)](#)

[Adding a logo to a template in Template Editor \(on page 109\)](#)

[Symbol properties in Template Editor \(on page 110\)](#)

[Picture properties in Template Editor \(on page 111\)](#)

[Viewing and modifying symbol file contents in Tekla Structures \(on page 112\)](#)

[Changing the symbol and the symbol file in Template Editor \(on page 113\)](#)

[Assigning rules to symbols in Template Editor \(on page 114\)](#)

[Assigning rules to pictures in Template Editor \(on page 115\)](#)

[Modifying a symbol in Template Editor \(on page 116\)](#)

[Setting paths for symbols and pictures in Template Editor \(on page 116\)](#)

Symbol and picture libraries in Template Editor

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 Locations** preferences dialog box.

See also

[Modifying File Locations preferences in Template Editor \(on page 27\)](#)

Adding symbols to a template in Template Editor

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

To insert a symbol into a template component:

1. On the toolbar, click the symbol button.
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 in Template Editor \(on page 113\)](#)

[Viewing and modifying symbol file contents in Tekla Structures \(on page 112\)](#)

[Modifying a symbol in Template Editor \(on page 116\)](#)

[Setting paths for symbols and pictures in Template Editor \(on page 116\)](#)

Adding pictures to a template in Template Editor

Ensure you have set the **File Locations** preferences for pictures before you begin.

To insert a picture:

1. On the toolbar, click the picture button.
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.

You can also specify a rule for a picture. Add and modify the rule in the **Picture Properties** dialog box.

See also

[File Locations preferences in Template Editor \(on page 26\)](#)

[Modifying File Locations preferences in Template Editor \(on page 27\)](#)

[Assigning rules to pictures in Template Editor \(on page 115\)](#)

Adding a logo to a template in Template Editor

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 Locations** 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 in Template Editor \(on page 107\)](#)

[Symbol and picture libraries in Template Editor \(on page 107\)](#)

[Setting paths for symbols and pictures in Template Editor \(on page 116\)](#)

Symbol properties in Template Editor

Symbol properties can be modified based on what you want to show in a report. Open the **Symbol Properties** dialog box to modify the output properties.

Property	Description
File	The symbol in the template.
Rule	The rule that has been applied to the picture. Modify the rule applied to the symbol by clicking the Rule button.
Name	A unique name for the symbol. If you change the name of the symbol, you must change it in all the rules it appears in.
Keep aspect ratio	Select the check box if you want to retain the aspect ratio of the symbol. Click Reset to restore the original aspect ratio for the symbol.
Height, Width	Set the height and width of the symbol.
Slant, Angle	The slant value sets the slope of the symbol. The angle (in degrees) that is applied to the symbol.
Line	Adjust symbol line properties like Color, Type and Width .

Property	Description
Fill	Adjust symbol fill properties like Color and Type .
Set as default for new symbols	Select the check box to save the settings applied to all new symbols.

See also

[Symbol and picture libraries in Template Editor \(on page 107\)](#)

[Adding symbols to a template in Template Editor \(on page 108\)](#)

[Assigning rules to symbols in Template Editor \(on page 114\)](#)

Picture properties in Template Editor

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

Property	Description
File	The picture file in the template. Click Select File to choose a different picture, then click Apply to replace it.
Rule	The rule that has been applied to the picture. Modify the rule applied to the picture by clicking the Rule button, which open the Rule Contents dialog.
Name	A unique name for the picture. If you change the name of the picture, you must change it in all rules it appears in.

Property	Description
Layout	Set the height and width of the picture.
Keep aspect ratio	<p>Select the check box if you want to retain the aspect ratio of the picture.</p> <p>Click Reset to restore the original aspect ratio for the picture.</p>
Align if smaller than container	When the placement of an picture file is based on a rule , set the horizontal (Left, Center, Right) and vertical (Top, Center, Bottom) settings. Then you are able to position a smaller picture in the space that has been reserved for it.
Set as default for new pictures	Select the check box to save the settings applied to all new pictures.

See also

[Symbol and picture libraries in Template Editor \(on page 107\)](#)

[Adding pictures to a template in Template Editor \(on page 108\)](#)

[Assigning rules to pictures in Template Editor \(on page 115\)](#)

[Assigning rules to symbols in Template Editor \(on page 114\)](#)

Viewing and modifying symbol file contents in Tekla Structures

Symbols are used in Template Editor. 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**.

**Note:**

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 in Template Editor \(on page 108\)](#)

[Setting paths for symbols and pictures in Template Editor \(on page 116\)](#)

Changing the symbol and the symbol file in Template Editor

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 in Template Editor \(on page 108\)](#)

Assigning rules to symbols in Template Editor

You can assign rules to symbols to define when they are output.

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

Do the following to apply a rule to the symbol that has already been inserted into the template component:

1. Select the symbol and double-click it to modify the properties.

The **Symbol Properties** dialog box opens.

2. Click **Rule** to open the **Rule Contents** dialog box.
3. Inside the conditional statement you can create a rule for the symbol with various data and string operations.

Here is an example of a rule that includes a symbol. The symbol library and the symbol id are separated by a @ character and they must appear in the rule.

```
Rule
if (GetValue("MAINPART_PROFILE_TYPE") == "I") then
    "parts.sym@3"
else
    "parts.sym@7"
endif
```

4. Click the **Check** button to ensure the syntax of the rule is correct.
5. Click **OK** to save the changes.
6. Save the template.

The symbol will be output according to the rules you have applied to it.

See also

[Adding symbols to a template in Template Editor \(on page 108\)](#)

[Adding pictures to a template in Template Editor \(on page 108\)](#)

[Creating and saving a template row rule \(on page 77\)](#)

[Template Editor formula and rule reference \(on page 143\)](#)

[Symbol properties in Template Editor \(on page 110\)](#)

Assigning rules to pictures in Template Editor

You can assign rules to pictures to define when they are output.

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

Do the following to apply a rule to the picture that has already been inserted into the template component:

1. Select the picture and double-click it to modify the properties.

The **Picture Properties** dialog box opens

2. Click **Rule** to open the **Rule Contents** dialog box
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.

Here is an example of a rule that includes a picture. The name of the picture file must appear in the rule.

```
Rule
if (GetValue("MAINPART_PROFILE_TYPE") == "I") then
    "I-profile.jpg"
else
    "EB-profile.jpg"
endif
```

4. Click the **Check** button to ensure the syntax of the rule is correct.
5. Click **OK** to save the changes.
6. Save the template.

The picture will be output according to the rules you have applied to it.

See also

[Adding pictures to a template in Template Editor \(on page 108\)](#)

[Creating and saving a template row rule \(on page 77\)](#)

[Template Editor formula and rule reference \(on page 143\)](#)

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 in Template Editor \(on page 108\)](#)

[Modifying component and object properties in Template Editor \(on page 50\)](#)

Setting paths for symbols and pictures in Template Editor

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 in Template Editor \(on page 108\)](#)

[Adding pictures to a template in Template Editor \(on page 108\)](#)

[Adding a logo to a template in Template Editor \(on page 109\)](#)

[Viewing and modifying symbol file contents in Tekla Structures *\(on page 112\)*](#)

[Changing the symbol and the symbol file in Template Editor *\(on page 113\)*](#)

Chapter 9. AutoCAD and Microstation files in Template Editor

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.

Find out more:

[Importing AutoCAD and MicroStation files with import method defined in Template Editor \(on page 118\)](#)

[Importing AutoCAD and MicroStation files with no import method defined in Template Editor \(on page 120\)](#)

[Setting fill type and font import settings in Template Editor \(on page 120\)](#)

[Limitations in AutoCAD and MicroStation file imports in Template Editor \(on page 121\)](#)

[AutoCAD and Microstation import errors in Template Editor \(on page 122\)](#)

[Modifying imported AutoCAD and MicroStation files in Template Editor \(on page 123\)](#)

Importing AutoCAD and MicroStation files with import method defined in Template Editor

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:

- 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. Templated 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. You can also select whether the file is imported to the nearest rounded unit (millimeters or inches). Click the check box for **Round object locations to the closest** <unit of measure you have specified>.
 6. Click **OK** to insert the file into the template component.
 7. You can modify the file if needed, for example, by ungrouping it and modifying individual file objects.



Note:

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 in Template Editor \(on page 23\)](#)

[Importing AutoCAD and MicroStation files with no import method defined in Template Editor \(on page 120\)](#)

[Setting fill type and font import settings in Template Editor \(on page 120\)](#)

[Limitations in AutoCAD and MicroStation file imports in Template Editor \(on page 121\)](#)

[AutoCAD and Microstation import errors in Template Editor \(on page 122\)](#)

[Grouping or ungrouping template objects \(on page 53\)](#)

Importing AutoCAD and MicroStation files with no import method defined in Template Editor

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 again to define the opposite corner point 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 in Template Editor \(on page 23\)](#)

[AutoCAD and Microstation files in Template Editor \(on page 118\)](#)

[Importing AutoCAD and MicroStation files with import method defined in Template Editor \(on page 118\)](#)

[Limitations in AutoCAD and MicroStation file imports in Template Editor \(on page 121\)](#)

[AutoCAD and Microstation import errors in Template Editor \(on page 122\)](#)

[Grouping or ungrouping template objects \(on page 53\)](#)

Setting fill type and font import settings in Template Editor

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 Locations** 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 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 Locations preferences in Template Editor \(on page 27\)](#)

[File Locations preferences in Template Editor \(on page 26\)](#)

Limitations in AutoCAD and MicroStation file imports in Template Editor

Here are some of the limitations in AutoCAD and MicroStation file import operations you may experience in Template Editor.

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).
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

Limitation	Description
	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	<p>You may experience problems with imported grouped objects and receive error messages during the import operation.</p> <p>User-defined line styles and world unit line widths from AutoCAD files are not imported.</p>

See also

[AutoCAD and Microstation files in Template Editor \(on page 118\)](#)

[Importing AutoCAD and MicroStation files with import method defined in Template Editor \(on page 118\)](#)

[Importing AutoCAD and MicroStation files with no import method defined in Template Editor \(on page 120\)](#)

[AutoCAD and Microstation import errors in Template Editor \(on page 122\)](#)

AutoCAD and Microstation import errors in Template Editor

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.

**Tip:**

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 in Template Editor \(on page 118\)](#)

[Importing AutoCAD and MicroStation files with no import method defined in Template Editor \(on page 120\)](#)

[Limitations in AutoCAD and MicroStation file imports in Template Editor \(on page 121\)](#)

[Grouping or ungrouping template objects \(on page 53\)](#)

Modifying imported AutoCAD and MicroStation files in Template Editor

An imported AutoCAD or MicroStation file is a group of shapes in Template Editor.

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 \(on page 53\)](#)

[Trimming template object locations \(on page 60\)](#)

[Importing AutoCAD and MicroStation files with import method defined in Template Editor \(on page 118\)](#)

[Importing AutoCAD and MicroStation files with no import method defined in Template Editor](#) (*on page 120*)

[Limitations in AutoCAD and MicroStation file imports in Template Editor](#) (*on page 121*)

Chapter 10. Template attributes and Template Editor

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>\bin\applications\Tekla\Tools\TplEd\settings`, but the location may be different in your environment.

Template attributes are defined in the following files:

File name	Description
<code>contentattributes.lst</code>	<p>This is a container file listing all the files that contain the actual attribute definitions. The files are added with <code>INCLUDE</code> sentences. The order of the files included in <code>contentattributes.lst</code> defines the reading order of the files.</p> <p>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.</p> <p>Generally, there is no need to modify <code>contentattributes.lst</code>. Do not modify it if you are not an administrator.</p>

File name	Description
<code>contentattributes_global.lst</code>	This file contains attributes that are hard-coded into the program. Do not modify this file.
<code>contentattributes_userdefined.lst</code>	<p>This file contains user-defined attributes, the same as in the <code>objects.inp</code> file.</p> <p>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.</p>

Find out more:

[User-defined template attributes and Template Editor \(on page 126\)](#)

[Free attributes in Template Editor \(on page 131\)](#)

[Global attributes and Template Editor \(on page 134\)](#)

See also

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

[Value field formulas in Template Editor \(on page 97\)](#)

[Controlling row output with rules in Template Editor \(on page 76\)](#)

User-defined template attributes and Template Editor

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:

①	②	③	④
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 \(on page 128\)](#)

[Adding comments to user-defined template attributes in Template Editor \(on page 129\)](#)

[Adding hierarchy to user-defined template attributes in Template Editor \(on page 130\)](#)

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
<u>MY_ATTRIBUTE</u>	<u>CHARACTER</u>	<u>LEFT</u>	<u>TRUE</u>
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>`.

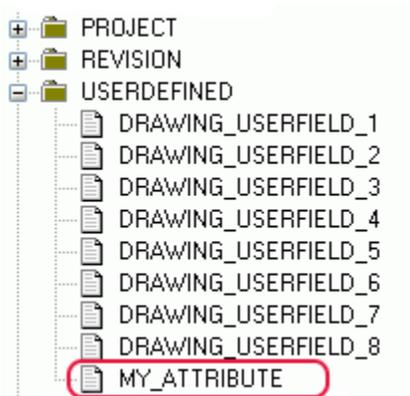
In this example, the content type is `DRAWING`.

```
// =====
// 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:`



See also

[User-defined template attributes and Template Editor \(on page 126\)](#)

Adding comments to user-defined template attributes in Template Editor

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`.

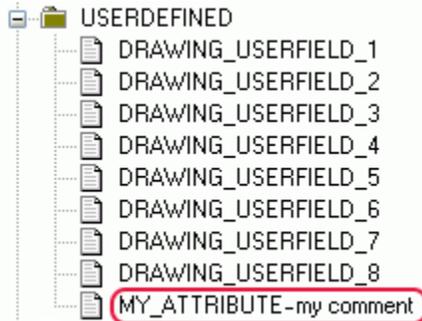
Do not modify the original `contentattributes_userdefined.lst` file.

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.



See also

[User-defined template attributes and Template Editor \(on page 126\)](#)

Adding hierarchy to user-defined template attributes in Template Editor

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`.

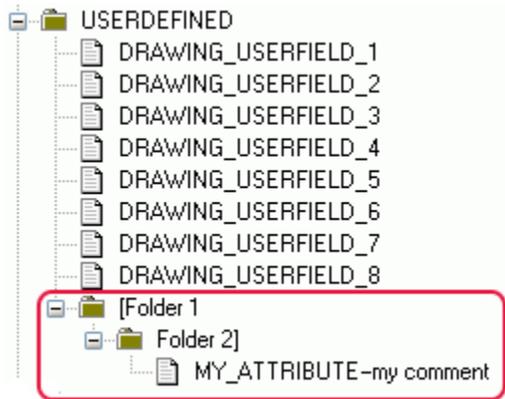
Do not modify the original `contentattributes_userdefined.lst` file.

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.



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 and Template Editor \(on page 126\)](#)

Free attributes in Template Editor

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 in Template Editor \(on page 131\)](#)

[Deleting free attributes in Template Editor \(on page 132\)](#)

[Modifying free attributes in Template Editor \(on page 133\)](#)

Adding free attributes in Template Editor

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 attribute 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 in Template Editor \(on page 133\)](#)

[Deleting free attributes in Template Editor \(on page 132\)](#)

Deleting free attributes in Template Editor

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

[Adding free attributes in Template Editor \(on page 131\)](#)

[Modifying free attributes in Template Editor \(on page 133\)](#)

Modifying free attributes in Template Editor

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.

There is now an option to remove the form feed character /f from output when a page break occurs during the output of a textual template. Click **Free Attributes**, which opens this dialog.

Free attributes

Application User

Name: Enable form feed

Add

Description: Add the form feed '\f' character to the output when a page break occurs. Default is Yes.

Modify

Type: Boolean

Delete

Value: No

Properties:

Name	Value	Type
------	-------	------

OK Cancel

1. In the **Application** tab, select **Enable form feed** from the **Name** field.
2. Change the default for **Value** from **Yes** to **No**.
3. Click **Add**.
4. Click **OK** to save the changes.

See also

[Deleting free attributes in Template Editor \(on page 132\)](#)

[Adding free attributes in Template Editor \(on page 131\)](#)

[Resizing value field fonts automatically in Template Editor \(on page 94\)](#)

Global attributes and Template Editor

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 the **File Locations** preferences.

See also

[File Locations preferences in Template Editor \(on page 26\)](#)

[Template rows \(on page 68\)](#)

[Template Editor objects \(on page 16\)](#)

[Value fields in Template Editor \(on page 85\)](#)

Chapter 11. Using templates in Trimble products

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

Find out more:

[Outputting a template created with Template Editor \(on page 135\)](#)

[Output options for templates created with Template Editor \(on page 136\)](#)

[Material listings created with Template Editor \(on page 137\)](#)

[Hierarchical listings created with Template Editor \(on page 138\)](#)

[Labels created with Template Editor \(on page 140\)](#)

[Creating an HTML report with Template Editor \(on page 141\)](#)

Outputting a template created with Template Editor

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 template headers and footers \(on page 61\)](#)

[Template row output properties \(on page 69\)](#)

Output options for templates created with Template Editor

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 Trimble products \(on page 135\)](#)

[Get a single piece of data from the database in Template Editor \(on page 101\)](#)

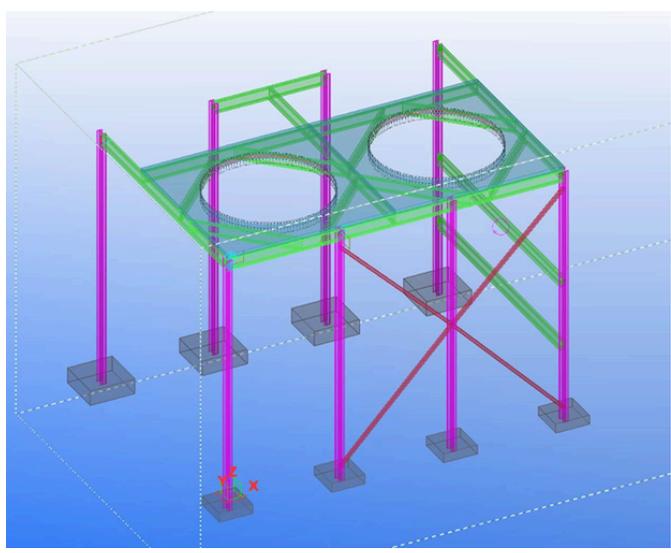
[Creating a listing of template objects from the database \(on page 102\)](#)

[Hierarchical listings created with Template Editor \(on page 138\)](#)

Material listings created with Template Editor

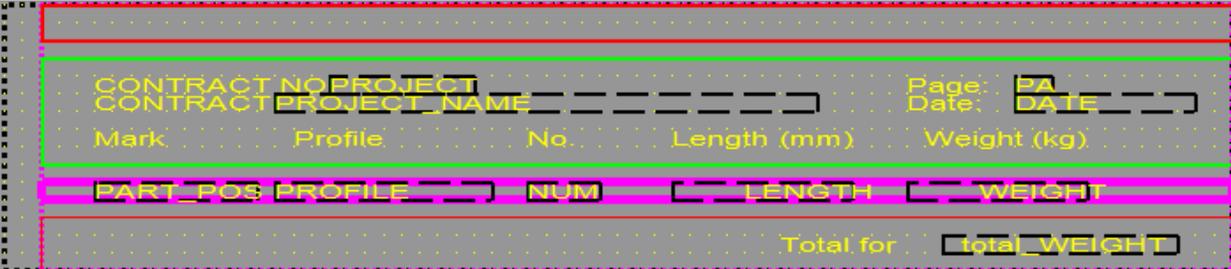
The following illustrates a textual template that creates a listing of selected parts of a building structure.

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 NO PROJECT _____ Page: PA
CONTRACT PROJECT_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

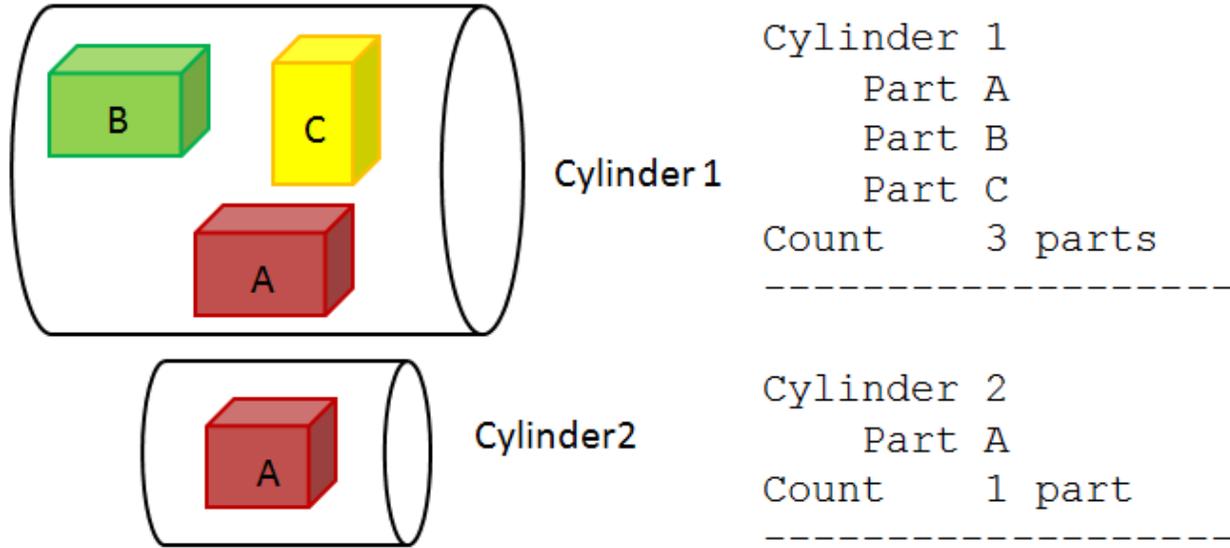
[Global attributes and Template Editor \(on page 134\)](#)

[Hierarchical listings created with Template Editor \(on page 138\)](#)

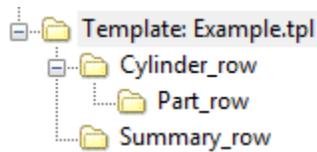
[Creating an HTML report with Template Editor \(on page 141\)](#)

Hierarchical listings created with Template Editor

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.



The listing is based on the following template:



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 template objects from the database \(on page 102\)](#)

Labels created with Template Editor

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 <input type="text"/>	Planner: <input type="text" value="Field_plan"/>		Author: <input type="text" value="Field_Auth"/>
	Drawing num. <input type="text" value="Field_num"/>		Job: <input type="text" value="Field_Proj"/>
	Plan: <input type="text" value="Field_plan"/>		
	Plan date: <input type="text" value="Field_plan"/>		
	Check date: <input type="text" value="Field_chec"/>		
Print date: <input type="text" value="Field_prin"/>			
Map scale: <input type="text" value="Field_scal"/>			
MAP SERIES <input type="text" value="NU"/>		MAP SHEET <input type="text" value="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
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 and Template Editor \(on page 134\)](#)

Creating an HTML report with Template Editor

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** box, 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 created with Template Editor \(on page 135\)](#)

Chapter 12. Template Editor 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.

Find out more:

[Adding comments to row rules and value field formulas in Template Editor \(on page 144\)](#)

[Offset for template objects \(on page 144\)](#)

[Conditional structure for row rules and formulas in Template Editor \(on page 145\)](#)

[Comparison and logical operators \(on page 146\)](#)

[Control functions in Template Editor \(on page 149\)](#)

[Attribute functions in Template Editor \(on page 152\)](#)

[Value field functions in Template Editor \(on page 154\)](#)

[String operations in Template Editor \(on page 159\)](#)

[Data type conversion functions in Template Editor \(on page 168\)](#)

[Mathematical operations in Template Editor \(on page 171\)](#)

See also

[Controlling row output with rules in Template Editor \(on page 76\)](#)

[Value field formulas in Template Editor \(on page 97\)](#)

[Arithmetic operators in Template Editor \(on page 172\)](#)

[Statistical functions in Template Editor \(on page 174\)](#)

[Trigonometric functions in Template Editor \(on page 173\)](#)

[Miscellaneous functions in Template Editor \(on page 177\)](#)

Adding comments to row rules and value field formulas in Template Editor

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> */`

```
Rule
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 in Template Editor \(on page 97\)](#)

[Creating and saving a template row rule \(on page 77\)](#)

[Modifying a template row rule \(on page 81\)](#)

[Modifying a value field formula in Template Editor \(on page 100\)](#)

Offset for template objects

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 in Template Editor \(on page 50\)](#)

[Moving a template object \(on page 54\)](#)

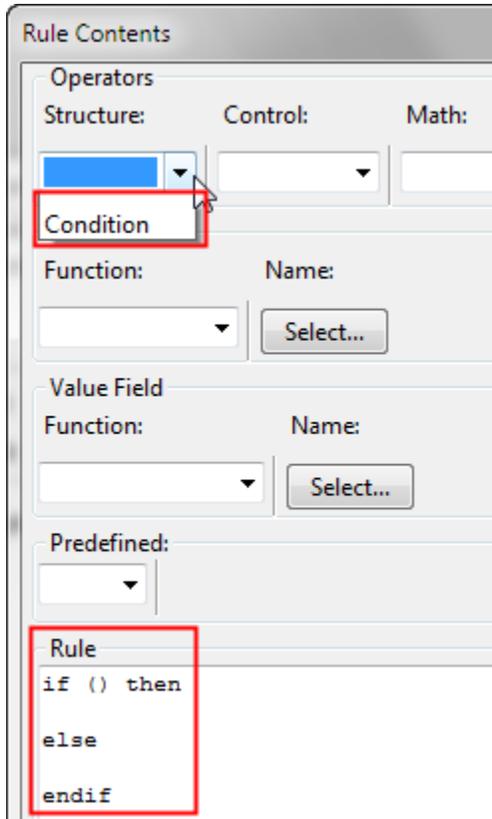
[find \(on page 160\)](#)

Conditional structure for row rules and formulas in Template Editor

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

Conditional structure can also be applied to value field formulas.

You can create rules and formulas by modifying the structure and conditions of the *if-else* statement. The example below is for a row rule.



See also

[Controlling row output with rules in Template Editor \(on page 76\)](#)

[Creating and saving a template row rule \(on page 77\)](#)

[Modifying a template row rule \(on page 81\)](#)

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.

Operator	Description	Example
		<pre> Rule if (GetValue("PROFILE_TYPE" == "B")) then Output() else StepOver() endif </pre>
!=	Sides are not equal	<p>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.</p> <pre> Rule if GetValue("ASSEMBLY_POS") !=NextValue("ASSEMBLY_POS") then Output() else StepOver() endif </pre>
<	Left side is smaller	<p>This rule finds all rows with the assembly position that is less than 10, anything equal to or greater than 10 is skipped.</p> <pre> Rule if GetValue("ASSEMBLY_POS") < 10 then Output() else StepOver() endif </pre>
<=	Left side is smaller or equal	<p>This rule outputs values with a LENGTH less than or equal to 30. If those values are not found, they are skipped.</p> <pre> Rule if GetValue("LENGTH") <= 30 then Output() else StepOver() endif </pre>
>	Right side is smaller	<p>This rule outputs values with a LENGTH greater than 50, anything equal to or less than 50 is skipped.</p>

Operator	Description	Example
		<pre> Rule if GetValue("LENGTH") > 50 then Output() else StepOver() endif </pre>
>=	Right side is smaller or equal	<p>This rule outputs values with a LENGTH equal to and greater than 30, anything less than 30 is skipped.</p> <pre> Rule if GetValue("LENGTH") >= 30 then Output() else StepOver() endif </pre>
&&	Logical AND, both conditions must be true	<p>If D1 is 200 and D2 smaller than 40, the row is output, otherwise it is skipped.</p> <pre> Rule if (D1==200 && D2<40) then Output() else StepOver() endif </pre>
	Logical OR, only one condition must be true	<p>If D1 is 200 or D2 is smaller than 40, the row is output, otherwise it is skipped.</p> <pre> Rule if (D1==200 D2<40) then Output() else StepOver() endif </pre>

See also

[Conditional structure for row rules and formulas in Template Editor \(on page 145\)](#)

[Value field functions in Template Editor \(on page 154\)](#)

[Control functions in Template Editor \(on page 149\)](#)

Control functions in Template Editor

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.

Some control functions can be used in formulas as well (`IsFirst`, `IsLast`, `PageBreak`). Access control functions in the **Formula Contents** dialog box from the **Control** list.

Find out more:

[IsFirst \(on page 149\)](#)

[IsLast \(on page 150\)](#)

[Output \(on page 150\)](#)

[PageBreak \(on page 151\)](#)

[StepIn \(on page 151\)](#)

[StepOut \(on page 152\)](#)

[StepOver \(on page 152\)](#)

IsFirst

This function checks if the row is the first one, and returns TRUE if it is. This function can be used in both row rules and value field formulas.

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.

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

See also[IsLast \(on page 150\)](#)

IsLast

This function checks if the row is the last one and returns TRUE if it is. This function can be used in both row rules and value field formulas.

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.

```

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

```

See also[IsFirst \(on page 149\)](#)

Output

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

This rule outputs rows if the **LENGTH** attribute is greater than 50. If the **LENGTH** attribute is less than or equal to 50, those rows are skipped.

```

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

```

You can also override the color of template objects with the **Output** function with the following syntax:

```
Output("#<key>=<color>;")
```

The following abbreviations are keys that refer to the template objects in question:

- VFC (value field color)
- TXC (text color)
- DRC (drawing object color)

You can combine these values in one row rule using a semicolon as a separator. This rule overrides the colors when the **LENGTH** attribute is longer than 1000, then value fields are orange, text objects are yellow and the drawing color is blue (i.e. lines are blue).

```

Rule
if (GetValue("LENGTH") > 1000) then
  Output("#VFC=ORANGE;TXC=YELLOW;DRC=BLUE")
else
  Output("#VFC=BLUE;COLOR=GREEN")
endif

```

See also

[Overriding template object colors with row rules in Template Editor \(on page 82\)](#)

PageBreak

This function inserts a page break. This function can be used in both row rules and value field formulas.

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

```

Rule
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.

```

Rule
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 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.

```

Rule
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.

```

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

```

Attribute functions in Template Editor

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.

Find out more:

[GetValue \(on page 153\)](#)

[NextValue \(on page 153\)](#)

[PreviousValue \(on page 154\)](#)

[IsSet \(on page 154\)](#)

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**.

```

Rule
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.

```

Rule
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.

```

Rule
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.

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.

```

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

```

All other rows are skipped.

Value field functions in Template Editor

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 (" ").

**Note:**

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.

Find out more:

[GetFieldFormula \(on page 155\)](#)

[CopyField \(on page 155\)](#)

[Sum \(on page 157\)](#)

[Total \(on page 157\)](#)

[Count \(on page 158\)](#)

[All \(on page 158\)](#)

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:

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

CopyField

This function is intended for copying summary fields from other locations in the hierarchy. **CopyField** always copies the source value from other locations, even if the row is hidden.

For all components, CopyField searches for the value field in the same component first, before searching in other components. For headers and page headers, it searches for the value field after it. For footers and page footers CopyField searches for the value field in the components before it.

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.



Note:

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")
```

Below are the cases how **CopyField** works to find the relevant values in sequential order.

Case	Scenario
1	CopyField searches for the source field in the same row.
2	CopyField searches for the source field in the same row hierarchy above where it has been placed (i.e. from child row to parent row).
3	CopyField searches for the source field in the rows after where CopyField has been placed.

Case	Scenario
4	CopyField searches for the source field outside of the row hierarchy in the rows before where CopyField has been placed.

See also

[Hiding template rows from output in Template Editor \(on page 75\)](#)

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 the 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 \(on page 157\)](#)

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 \(on page 157\)](#)

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**.

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

See also

[All \(on page 158\)](#)

All

Use this function when you want to calculate the total count of a certain value field.

This function is not available for row rules.

This formula calculates the total number of all value fields named **ValueField_1**.

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

See also

[Count \(on page 158\)](#)

String operations in Template Editor

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.

Find out more:

[find \(on page 160\)](#)

[findany \(on page 160\)](#)

[match \(on page 161\)](#)

[mid \(on page 161\)](#)

[length \(on page 161\)](#)

[reverse \(on page 162\)](#)

[getat \(on page 162\)](#)

[setat \(on page 162\)](#)

[replace \(on page 163\)](#)

[toupper \(on page 163\)](#)

[tolower \(on page 163\)](#)

[fvf \(on page 164\)](#)

[newline \(on page 166\)](#)

[join \(on page 167\)](#)

[join2 \(on page 167\)](#)

[asc \(on page 168\)](#)

find

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

Parameters: string, substring

For example, this formula looks for the first instance of a dash. (-):

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

The character count for the string starts at 0. The result equals 2.

Note that this function is case-sensitive and will search for, for example, "WI300", but not "wi300" if it is specified as such in a formula.

findany

This function returns the offset of a string or character and ends the search when any of the conditions are met. If the search does not reveal a match, -1 is returned. This works in both value field formulas and row rules.

Parameters: source string, string (for a character search) OR source string, substring, substring (for a substring search)

This example searches for the first occurrence of '=', ',' or '' in a string and returns the offset when the first of those is found. The character count for the string starts at 0.

```
Formula
findany(GetValue("DESCRIPTION"), "=; ")
```

This example searches for the first occurrence of the strings "X:", "Y:", "End" or "Start" and returns the offset when the first of those is found. Note that the search for the strings is case-sensitive, so it will not find instances of "x:", "y:", "end" or "start".

```
Formula
findany(GetValue("DESCRIPTION"), "X:", "Y:", "End", "Start")
```

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

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

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

- ? (represents 0 or 1)
- * (represents 0 or more characters)

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".

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

length

This function returns the number of characters in a string.

Parameter: string

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

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

reverse

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

Parameter: string

The following example prints "elbmirt" in a report.

```
Formula
reverse("trimble")
```

getat

This function returns the character in a given offset.

Parameters: string, offset

You can also specify a negative offset, which will calculate the start of the offset from the end of the string.

The following rule calculates the offset of the characters H, E and A, essentially filtering and searching for **PROFILES** that begin with HEA. The offset starts at 0.

```
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

You can also specify a negative offset, which will calculate the start of the offset from the end of the string.

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

Formula

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

replace

This function replaces all specified substrings with another string.

Parameters: source string, old substring, new substring

For example:

Formula

```
replace(GetValue("PROFILE"), " ", "")
```

This returns a string in which all of the space characters from the value returned by `GetValue("PROFILE")` have been removed.

toupper

This function changes parameter string to all uppercase and returns the string.

Parameter: string

For example:

Formula

```
toupper(GetValue("NAME"))
```

This returns the value of the NAME-attribute of the part in all uppercase letters, for example, COLUMN.

tolower

This function changes parameter string to all lowercase and returns the string.

Parameter: string

For example:

Formula

```
tolower(GetValue("NAME"))
```

This returns the value of the NAME-attribute of the part in all lowercase letters, for example, column.

fvf

Use this function to find a value in an external file.

Parameters: filename, key value of row, column number, separator(s), allow empty columns

Separators and allowing empty columns are optional parameters. The default separators are space and tab. The default for allowing empty columns is FALSE.

Where the format for reading files is:

```
fvf("filename", "key_value_of_row", column_number, "separator(s)", FALSE)
```

The path for the file can be included in the filename parameter.

Template Editor supports several separators, so ",\t" would make all the space, comma, tab and newline characters separators.

The function retrieves the value `P20(200x1200)` from column 3 of the row identified by the string "2" in the file "hollowcore_slab_dim.dat."

Row	Column 1	Column 2	Column 3
0	120		P15(150x1200)
1	160		P18(175x1200)
2	180		P20(200x1200)
3	0		P27(265x1200)
4	0		P32(320x1200)
5	0		P37(370x1200)
6	0		P40(400x1200)
7	0		P50(500x1200)

`fvf("hollowcore_slab_dim.dat", "2", 3)`

Below are several hypothetical examples that illustrate how the syntax in the fvf formula produces different results, including the use of a fifth parameter in the formula to produce empty columns.

This example specifies an input file called "sample.dat", with the line:

A; B;; C; D

In this case, the formula

Formula

```
fvf("sample.dat", "A", 3, ";", TRUE)
```

will return an empty string: ""

As each separator in the above example is considered meaningful, the third column is an empty column between the first ';' and the first ' ' separators. So the column values in this case are:

1: "A"

2: " B"

3: ""

4: " C"

5: " D"

and

Formula

```
fvf("sample.dat", "A", 3, ";", FALSE)
```

will return: "C"

A subsequent example would be when multiple separators are used in the input file, i.e. a mix of spaces and commas.

A, B, C, D

In this case, the formula:

Formula

```
fvf("sample.dat", "A", 3, ", ", TRUE)
```

will return: "B"

and

Formula

```
fvf("sample.dat","A",4, " ", " ", TRUE)
```

will return an empty string: ""

and

Formula

```
fvf("sample.dat","A",3, " ", " ", FALSE)
```

will output: "C"

and finally

Formula

```
fvf("sample.dat","A",4, " ", " ", FALSE)
```

will output: "D"

That is, when the last parameter is FALSE, consecutive separators are interpreted as one separator.

TRUE = allow empty columns, thus consecutive separators are treated as unique separators.

FALSE = do not allow empty columns, thus consecutive separators are treated as one separator.

For more information related to the support for this function, consult the Tekla User Assistance topic, "How to read data from a text file in Custom Components."

newline

This function inserts a new line in a formula. You can either select it from the list or add it manually by typing \n.

NOTE: This function is only available for value field formulas.

For example:

Formula

```
"Width: " + GetValue("WIDTH") + "\n" + "Height: " + GetValue("HEIGHT")
```

The result would be:

Width: 100

Height: 27

join

This function joins a list of strings separated with a separator into one string and trims away the empty strings.

Parameter: separator

For example:

```
Formula
join(";", CopyField("HOLE_4"), CopyField("HOLE_5"),
CopyField("HOLE_6"), CopyField("HOLE_8"))
```

If the `CopyField` functions return the values 7, 7, <nothing>, 10, the result is 7;7;10.

<nothing> in this example means that the `HOLE_6` value field is not output because its row is stepped over based on the row rule.

See also

[join2 \(on page 167\)](#)

join2

This function joins a list of strings separated with a separator into one string and retains the empty strings.

Parameter: separator

For example:

```
Formula
join2(";", CopyField("HOLE_4"), CopyField("HOLE_5"),
CopyField("HOLE_6"), CopyField("HOLE_8"))
```

If the `CopyField` functions return the values 7, 7, <nothing>, 10, the result is 7;7;;10.

<nothing> in this example means that the `HOLE_6` value field is not output because its row is stepped over based on the row rule.

See also

[join \(on page 167\)](#)

asc

This function returns a character code or an integer code, depending on how it is input.

Parameter: string, integer

If the parameter is a string it returns an integer representing the character code to the first letter in a string.

If the parameter is an integer, it returns characters representing the integer code.

```
Formula
asc("A")
asc(65)
```

Data type conversion functions in Template Editor

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:

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

Find out more:

[int \(on page 169\)](#)

[double \(on page 169\)](#)

[string \(on page 169\)](#)

[vwu \(Value with unit\) \(on page 170\)](#)

[format \(on page 171\)](#)

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 3.

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

If rounding is required, use the `round` function. In this example, if the attribute value is 3.5, it is first rounded to 4.0 and then the `int` function returns 4.

```
Formula
int (round (GetValue ("SCALE1")))
```

See also

[round \(on page 181\)](#)

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

```
Formula
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. 0 means the string output contains a number with no decimals.

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

Here are some other examples of using string and the results it produces. The first line of each example is the formula and the second line is the result it would produce.

```
string(3.14, 1)
3.1
```

```
string(3.14, 1, 0)
3.1
```

```
string(3.14, 0, 5)
3.14
```

```
string(3.1, 3, 3)
3.100
```

```
string(3.1, 3, 0)
3.100
```

```
string(3.1245, 0, 2)
3.12
```

The format in the brackets is always (value, MinDecimals, MaxDecimals)

If you have a string function with MinDecimals set at 3, and there are not enough decimals in the string, zeros will be added. For example: string(3.1, 3, 3) would equal 3.100, because both MinDecimals and MaxDecimals have been set at 3.

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 the length unit is set to mm and decimals are set to 2 in the **Value Field Properties** dialog box.

Formula

```
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.

Formula

```
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.

Formula

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

To change the unit of measure (i.e. inches or millimeters), modify the **Graphical units** option in the **General** tab of the **Preferences** dialog.

See also

[Assigning value field meanings in Template Editor \(on page 93\)](#)

[Value field meanings in Template Editor \(on page 90\)](#)

Mathematical operations in Template Editor

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.

Find out more:

[Arithmetic operators in Template Editor \(on page 172\)](#)

[Trigonometric functions in Template Editor \(on page 173\)](#)

[Statistical functions in Template Editor \(on page 174\)](#)

[Miscellaneous functions in Template Editor \(on page 177\)](#)

Arithmetic operators in Template Editor

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.	<p>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.</p> <div data-bbox="597 1312 1201 1428" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Formula</p> <pre>(GetValue("AREA") - (GetValue("AREA_PZ") + GetValue("AREA_NZ"))) / (GetValue("WIDTH"))</pre> </div>
-	Subtraction	See the formula above.
*	Multiplication	<p>This formula uses the multiplication to calculate areas and find the biggest value of similar fields.</p> <div data-bbox="597 1711 1201 1753" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Formula</p> <pre>max(GetValue("LENGTH") * GetValue("HEIGHT"), GetValue("LENGTH") * GetValue("WIDTH"))</pre> </div>

Operator	Description	Example
/	Division	<p>This formula uses division to transform imperial units to feet and the floor function to round down the result.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Formula</p> <pre>floor (GetValue ("LENGTH") /25.4/12)</pre> </div>

See also

[Value field formulas in Template Editor \(on page 97\)](#)

[Controlling row output with rules in Template Editor \(on page 76\)](#)

Trigonometric functions in Template Editor

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 in Template Editor

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 \(on page 175\)](#)[floor \(on page 175\)](#)[min \(on page 176\)](#)[max \(on page 176\)](#)[sqsum \(on page 176\)](#)[ave \(on page 177\)](#)[sqave \(on page 177\)](#)

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.

Formula
<code>ceil(Sum("NUMBER_field")+0.03)</code>

See also[floor \(on page 175\)](#)

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.

Formula
<code>floor(GetValue("LENGTH")/25.4/12)</code>

See also

[ceil \(on page 175\)](#)

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.

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

See also

[max \(on page 176\)](#)

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.

```
Formula
"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 \(on page 176\)](#)

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.

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

ave

This function calculates the average of parameters.

Parameters: x, y, z,...

Use the following syntax to calculate the average.

```
Formula
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.

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

Miscellaneous functions in Template Editor

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.

Find out more:

[fabs \(on page 178\)](#)

[exp \(on page 178\)](#)

[ln \(on page 179\)](#)

[log \(on page 179\)](#)

[sqrt \(on page 179\)](#)

[mod \(on page 179\)](#)

[pow \(on page 180\)](#)

[hypot \(on page 180\)](#)

[n! \(on page 180\)](#)

[round \(on page 181\)](#)

[and \(on page 181\)](#)

[or \(on page 181\)](#)

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.

```
Formula  
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.

```
Formula  
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.

```
Formula
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.

```
Formula
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.

```
Formula
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.

```
Formula
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.

```
Formula
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.

```
Formula
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.

```
Formula  
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.

```
Formula  
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.

```
Formula  
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.

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

Index

A

- acos
 - 173
- Adding
 - hierarchy to user-defined template attributes
 - 130
- Adding a value field
 - 85
- Adding comments to formulas
 - 144
- Adding comments to rules
 - 144
- Adding free attributes
 - 131
- Adding logos to template
 - 109
- Adding pictures
 - 108
- Adding symbols to template
 - 108
- Adding user-defined template attributes
 - 128
- Aligning template objects
 - 60
- Aligning value field texts
 - 96
- Arithmetic operators
 - 172
- asin
 - 173
- Assigning value field meanings
 - 93
- atan
 - 173
- atan2
 - 173
- Attribute functions
 - 152
 - GetValue
 - 153
 - IsSet
 - 154
 - NextValue
 - 153
 - PreviousValue
 - 154
- Attributes
 - user-defined
 - 126
- AutoCAD file imports
 - error messages
 - 122
 - limitations
 - 121
- AutoCAD files
 - 118
 - fill type import settings
 - 120
 - font import settings
 - 120
 - importing
 - 118, 120
 - modifying
 - 123

C

- Calculating hypotenuse
 - 180
- Changing symbol file
 - 113
- Changing symbols

113	47
Changing template component type	modifying properties
53	50
Color map	Conditional structure
26	145
Column settings	Content Browser
fill direction	20
72	hide template objects
fill policy	65
72	Content types
modifying	69
75	for rows
multiple columns	73
75	Continuous action
number of columns	62
72	Control functions
space between columns	149
72	IsFirst
starting point	149
72	IsLast
Combined rows	150
86	Output
Combining lines with the same information	150
84	PageBreak
Commenting	151
user-defined template attributes	StepIn
129	151
Comparison operators	StepOut
146	152
Components	StepOver
changing the type	152
53	Conversion attributes
cut, copy, paste	26
49	Conversion colors
inserting a page footer	26
47	Conversion fonts
inserting a page header	26

- CopyField
 - 155
- Copying value field values
 - 155
- cos
 - 173
- cosh
 - 173
- Creating a formula
 - 97
- Creating a pre-defined template
 - 36
- Creating a row rule
 - 77
- Creating a value field
 - 85
- Creating HTML reports
 - 141
- Creating material listings
 - 102
- Creating row rules
 - creating with Rule Wizard
 - 78
- Creating rows
 - 68
- D**
- Data type conversion functions
 - 168
 - double
 - 169
 - format
 - 171
 - int
 - 169
 - string
 - 169
 - vwu (value with unit)
 - 170
 - Data types for value fields
 - 92
 - Defining template output
 - formulas
 - 97
 - Definition file for row contents
 - 26
 - Definition files
 - 18
 - Deleting a pre-defined template
 - 36
 - Deleting free attributes
 - 132
 - Drawing a polyline
 - 43
 - Drawing objects
 - setting visibility
 - 45
 - snapping lines
 - 63
 - use grid
 - 64
 - Drawing operations
 - 62
 - Drawing with numeric pad
 - 64
 - Duplicate lines
 - 84
 - Duplicate objects
 - 50
 - E**
 - Extending template objects
 - 56
 - F**
 - File Location preferences
 - setting symbol directories

- 116
- File Locations preferences
 - modifying
 - 27
- Files
 - in templates
 - 16
- Fill import map
- 26
- Filtering object selection
- 65
- Filtering rows from output
- 75
- Find values in external file
- 164
- Finding text
- 67
- Fitting template in window
- 23
- Font properties
 - for template objects
 - 66
- Fonts
 - resizing
 - 94
- Formula Contents properties
- 98
- Formulas
- 143
 - adding comments
 - 144
 - arithmetic operators
 - 172
 - comparison operators
 - 146
 - conditional structure
 - 145

- creating
- 97
- data type conversion functions
- 168
- logical operators
- 146
- mathematical operations
- 171
- offset
- 144
- properties
- 98
- saving
- 97
- statistical functions
- 174
- Formulas and rules
 - attribute functions
 - 152
 - binary operations
 - 181
 - misc. functions
 - 177
- Free attributes
- 131
 - adding
 - 131
 - deleting
 - 132
 - modifying
 - 133

G

- Global attributes
- 134
- Global attributes definition file
- 26
- Graphical fields

- inserting
 - 44
 - setting visibility
 - 45
 - Graphical objects
 - offset
 - 144
 - Graphical templates
 - 15
 - Grid
 - activating
 - 28
 - aligning template objects
 - 28, 60
 - drawing template objects
 - 64
 - showing or hiding
 - 29
 - Grid settings
 - density
 - 29
 - Grouped objects
 - errors during import
 - 122
 - Grouping template objects
 - 53
- H**
 - Headers and footers
 - output options
 - 61
 - Hiding the grid
 - 29
 - Hiding value fields from output
 - 95
 - HTML reports
 - 141
- I**
 - Importing AutoCAD and MicroStation files
 - 118, 120
 - Inserting a graphical field
 - 44
 - Inserting a value field
 - 85
 - Inserting template components
 - 42
 - Inserting template objects
 - 42
 - Inserting text object
 - 44
 - Interface components
 - workarea
 - 19
- K**
 - Keep aspect ratio
 - 52, 110, 111, 118
 - Keyboard shortcuts
 - 30
- L**
 - Logical operators
 - 146
 - Logo
 - adding to template
 - 109
- M**
 - Main features
 - customize output
 - 14
 - Material listings
 - 137
 - creating
 - 102
 - MicroStation file imports
 - limitations
 - 121

MicroStation files	sqrt, square root
118	179
fill type import settings	Modifying a formula
120	100
font import settings	Modifying AutoCAD and MicroStation files
120	123
import errors	Modifying file locations preferences
122	27
importing	Modifying free attributes
118, 120	133
modifying	Modifying row rules
123	81
Misc. mathematical functions	Modifying value field properties
177	86
Miscellaneous functions	Moving template components
and	52
181	Moving template objects
exp	54
178	N
fabs	New features
178	13
hypot	O
180	Object properties
ln, logarithm	attributes
179	125
log, logarithm	Objects
179	cut, copy, paste
mod, modulo	49
179	duplicating
n!, factorial	50
180	modifying properties
or	50
181	Offset
pow	144
180	Output options for templates
round	136
181	Output policy

- page footer
 - 47
- page header
 - 47
- Outputting a template
 - 135
- P**
- Page footer
 - 47
- Page header
 - 47
- Panning
 - 22
- Partially pick objects
 - 48
- Pens and colors
 - 33
- Picking multiple objects
 - 48
- Picture libraries
 - 107
- Picture properties
 - 111
- Pictures
 - 107
 - adding to template
 - 108
 - assigning rules
 - 115
 - setting paths
 - 116
- Polyline
 - drawing
 - 43
- Pop-up menus
 - 32
- Pre-defined rules
 - 83
- Pre-defined templates
 - 36
- Presentation files
 - 26
- Presentations
 - in templates
 - 33
- Preview pane
 - 20
- Program preferences
 - file locations
 - 26
 - general
 - 23
 - workarea
 - 25
- R**
- Removing freeform character
 - 133
- Restoring original view
 - 23
- Rotating symbols
 - 59
- Rotating template objects
 - 59
- Rotating text objects
 - 59
- Rotating value fields
 - 59
- Row hierarchy
 - modify
 - 76
- Row output
 - combine rows
 - 74
 - distinct rows

74	73
Row rules	modifying hierarchy
creating	76
77	modifying sort type
modifying	74
81	output properties
overriding template object colors	69
82	outputting sub-rows
properties	151
79	rules
saving	76
83	user-defined free attributes
storing rules	154
77	Rule Contents properties
Rows	79
16, 68	Rule wizard
content types	78
69	Rules
controlling output with rules	143
149	adding comments
creating	144
68	arithmetic operators
filtering	172
152	comparison operators
filtering output	146
150	conditional structure
getting values of previous row	145
154	data type conversion functions
hide from output	168
75	for pictures
ignoring rows in output	115
152	for symbols
inquiring next output	114
153	logical operators
inquiring output	146
149, 150	mathematical operations
modifying content type	171

properties	79	Setting grid density	29
row output	76	Setting pens and colors in Template Editor	33
statistical functions	174	Setting picture file paths	116
S		Setting properties for objects	58
Saving a pre-defined template	36	Setting row output properties	69
Searching for text	67	Setting symbol file paths	116
Selecting objects		Setting text visibility	45
using toolbar	65	Setting value field meanings	26
with a filter	65	Setting work directory for Template Editor	26
Selecting template components	48	Showing the grid	29
Selecting template objects	48	sin	173
Selection toolbar	65	sinh	173
Setting a directory		Snapping line objects	63
definition files	27	Sorting order	
Setting drawing object visibility	45	value field output	91
Setting file for pre-defined rules	26	Special drawing operations	62
Setting folder(s) for fonts	26	Statistical functions	174
Setting folder(s) for pictures	26	ave	177
Setting folder(s) for symbols	26	ceil	175
Setting graphical field visibility	45		

floor	166
175	replace
max	163
176	reverse
min	162
176	setat
sqave	162
177	tolower
sqsum	163
176	toupper
Status bar	163
33	Summing information in value fields
Storing a rule	103
77	Symbol Editor
String operations	116
159	Symbol files
asc	changing
168	113
find	Symbol libraries
160	107
findany	Symbols
160	107
fvf	adding to template
164	108
getat	assigning rules
162	114
join	changing
167	113
join2	in templates
167	16
length	modifying
161	116
match	modifying symbol files
161	112
mid	rotating
161	59
newline	setting paths

116	Template definitions
symbol properties	35
110	Template Editor
viewing symbol files	about
112	13
T	file locations preferences
tan	26
173	general preferences
tanh	23
173	grid
Tekla fonts	28
66	interface overview
Tekla Structures	19
advanced options for value fields	keyboard shortcuts
105	30
language string in Template Editor	pop-up menus
105	32
Template attributes	preview pane
125	20
adding	setting up interface
128	23
Template components	status bar
16, 42	33
changing name	toolbars
51	19
cropping	user interface
53	19
inserting into template	workarea
42	19
modifying size	workarea preferences
52	25
moving	Template Editor components
52	42
selecting	Template Editor interface
48	modify components
Template definition files	21
18	Template Editor objects

42	
Template field objects	59
Graphical fields	selecting
16	48
Value fields	setting default properties
16	58
Template footers	text properties
16	57
Template grid	trimming
28	55
Template headers	trimming location
16	60
Template objects	ungrouping
16, 42	53
aligning	view or hide
60	65
aligning to grid	Template output
60	creating reports
changing name	141
51	customize
extending	14
56	duplicate lines
font properties	84
66	forcing a page break
formatting with painter	151
56	hierarchical listings
grouping	138
53	map labels
inserting into templates	140
42	material listings
modifying size	137
52	Template page
moving	modifying properties
54	39
overriding colors with row rules	Template page footers
82	16
rotating	Template page headers
	16

Template rows	output process
16, 68	135
creating a rule	outputting
77	135
creating a rule with Rule Wizard	page properties
78	39
pre-defined rules	reports and labels
83	135
rules for rows	saving
76	35
saving a rule	saving pre-defined
77	36
Template shapes	textual templates
16	15
Template type	Text in templates
modifying	16
40	Text objects
Template windows	57
21	inserting
Templates	44
35	rotating
changing type	59
40	setting visibility
closing	45
38	Text properties
creating	57
35	Textual templates
creating pre-defined	15
36	Tips
deleting pre-defined	template components
36	61
graphical templates	Template Editor interface
15	30
opening	template objects
38	61
output options	value fields
136	104

Toolbars	145
19, 34	modifying
Tooltips	100
34	Value field functions
Trigonometric functions	154
173	All
Trimming template object location	158
60	Count
Trimming template objects	158
55	GetFieldFormula
TrueType fonts	155
66	Sum
U	157
Ungrouping template objects	Total
53	157
User-defined attributes	Value field meanings
126	90, 93, 96
User-defined template attributes	Value field properties
adding comments	86
129	Value fields
adding hierarchy	85
130	aligning texts in output
Using format painter for objects	96
56	assigning meanings
Using presentations	93
in templates	creating
33	85
Using row rules to change template objects	formatting during output
colors	96
82	formatting output
V	90
Value field attributes	gather data
resizing fonts automatically	101
94	hiding from output
Value field formulas	95
97	language strings
conditional structure	105

modifying output properties

86

out a blank field

104

resizing fonts

94

rotating

59

setting data type

92

setting layout

94

setting priority

91

setting query on other fields

155

setting sort order

91

sorting priority

91

summary fields

103

TS advanced options

105

zero value in value field

104

Z

Zooming

22