software

b&w

SMART Annotate

Administration guide

Contents

1

Introduction

About SMARTAnnotate

Generally **SMARTAnnotate** is designed to
support **users and administrators** with
placing, harmonizing and managing Standard annotations
in **2D drawings** and **3D models**

The software provides the following functions:

- User dialog to easily select and place annotations
- Multi-language support for all kinds of annotation
- Drawing Format Management
- Meta Data management.
- Administrator dialog to comfortably organize the annotation elements and structure

The Ribbon UI

All **SMARTAnnotate** commands are accessible from the ribbon tab **SMARTAnnotate**. The ribbon is available in the modes: Part, Assembly and Drawing. The content of the ribbon is the same for all modes.
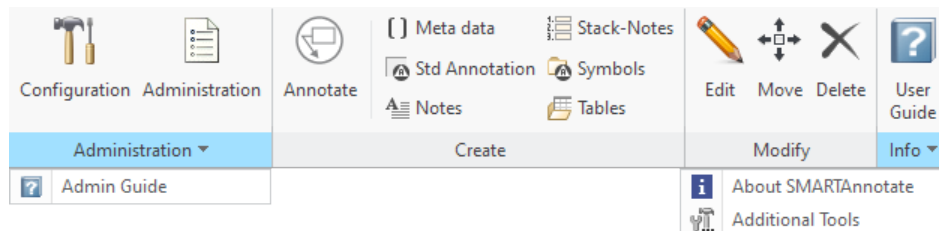
- **Administration:**
  - ○ Configuration — (Admin mode only) opens the configuration dialog
  - ○ Administration — (Admin mode only) opens the administrator dialog
  - ○ Font Alignment — (Admin & drawing mode only) auto-creates necessary data to properly align Stack-Notes vertically if a non-mono-spaces font is used
  - ○ Admin Guide — (Admin mode only) opens software documentation created for the administrators
- **Create**:

- ○ Annotate — opens the user dialog to initially place annotations
- ○ Shortcuts to all tabs within the user dialog
- **Modify**:
  - ○ Edit — Edit any item that originally was placed with **SMARTAnnotate**
  - ○ Move — Move any item that originally was placed with **SMARTAnnotate**
  - ○ Delete — Delete any item that originally was placed with **SMARTAnnotate**
- **Info**:
  - ○ User Guide— opens software documentation created for the users
  - ○ About SMARTAnnotate — shows copyright and version information
  - ○ Additional Tools — Check out the other great Creo tools provided by B&W Software GmbH

2

Preparation

Installation

1. Unzip the downloaded software package and save it to any file location

2. Run the `setup.exe` file

3. Install the license server if required

   - If you have a demo license key you need to put it into the `/bin` folder of the installation

4. Install the software to the intended installation path.

5. Specify the path to your **Configuration** folder

   - Use the environment variable `SAN_USER_CONFIG_PATH` to point to the location of the `/configuration` folder of the software.

   - You may copy the default configuration folder to a new name and point to the new folder.

   - Default is `<install_dir>/coonfiguration`

6. Add **SMARTAnnotate** to your protk.dat file

7. Start Creo Parametric and find the additional **SMARTAnnotate** ribbon tab

   - To start the software as an administrator you have to set the environment variable `SAN_ADMIN_MODE_ENABLED` to `TRUE`

   - You will detect two additional icons within the **SMARTAnnotate** ribbon tab: **Configuration** and **Administration**

8. Specify the language that will be used for content in the Admin dialog

   - Use the environment variable `SAN_UI_LANG_FOR_CONTENTS` to point to the name of the intended language folder in the `/translation` directory of the software. (i.e. 'german')

   - The default setting is the language of the Creo session, but if that language is not available in the `/translation` directory of the software 'English' will be used.

Important Information and Conventions

General information

As **SMARTAnnotate** is written with the Creo API Pro/TOOLKIT the Undo-buffer of Creo will be emptied whenever you access a **SMARTAnnotate** command.

Naming conventions and explanations

Configuration	A **Configuration** is the whole structure of annotations available in one Creo session.
Filter	A **Filter** is a single item of a whole tree structure. It is used to distinguish annotation items for different departments or organizations within an enterprise. The filter tree structure is unlimited.
Mode	There are three **Modes** that **SMARTAnnotate** can run in. ⬜ Assembly mode ⬜ Part mode ⬛ Drawing mode
Object	An **Object** is an item that can be assigned to any filter within the tree structure. It will be inherited throughout the tree branch if intended. An **Object** contains the definition of the annotation including the text definition. The following objects are available: [ ] Parameters ⁰⁰₀ Parameter Sets ⌐⌐ Formats 🔒 Std-Annotations ▦ Std-Annotation Sets A≣ Notes Stack Notes ◤ Flags 🔒 Symbols Tables (2D mode only)

Object attributes	Within the Object Edit dialog, the attributes have different icons to illustrate their relevance for the revising mechanism or to show a status
	○ Attribute not relevant for revising mechanism
	● Attribute relevant for revising
	◌ System attributes
	● Unpublished object
	⚠ Required entry that is still missing
Publishing	The publishing mechanism has the following
	⚙ New object —> Needs to be published, can be deleted
	☁ Modified object —> Needs to be published, can only be reset to latest published revision

Revision mechanism

SMARTAnnotate offers a sophisticated revising functionality that is designed to

- Restore drawing or model content to any time in the past for translation purpose
- Notify the user about changes in texts and / or in the configuration
- Distinguish between published revisions for final usage and temporary revisions for administrative purposes.

Please refer to the 'Revision objects on page 69' chapter to find a detailed description about revising objects.

Text and Multi-language syntax

To provide an environment to easily manage multiple languages **SMARTAnnotate** uses a placeholder mechanism for all texts that are used in annotation entities like **Notes**, **Symbol** or **Tables**. The syntax is as follows

`\[L1](1)[A]\\[V]\`

- [L1] — indicates the main language. Use L2, L3, ... for additional languages
- (1) — indicates the first line of a text. Use 2, 3, ... for additional rows

- [A] — is a placeholder and will be replaced with the respective text block. You can use multiple placeholders and any characters.
- [V] — is a placeholder for user input. You may use multiple value placeholders and any characters.
- \...\ — indicates a formatting information that separates from other texts.

Content Preparation

Before starting to create your **SMARTAnnotate** configuration you should provide basic items to be used within your configuration.

Creo Symbols

You will need Creo Symbols for the stand alone **Symbols**, **Formats**, **Std-Annotations** and **Flags**. Later on you even might use custom symbols within the **Note** definition.

Define your symbols in a 2D drawing. Use **Symbol Galery ▶ Define** to open the symbol definition mode.

Your symbols can contain any kind of graphic and variable text. The variable text needs to follow a special syntax to work correctly in multi-language environments.

Creo Drawing Format

Within the **Format** definition in the 2D mode, you will need to assign Creo Drawing Format files (.frm). It is a specific file type that you can create in the Creo system.

The best approach is to define one .frm file per paper size that will suit all situations and requirements. The layout of the drawing table will be done by placing tables and symbols.

Creo Tables

Creo tables are used in the **Format** definition and as standalone **Tables** in 2D mode. You can call out parameter values and use the special syntax for multi-language situations.

📝 **Note**

Use a Drawing Format (.frm) as Creo object to work on your tables. That keeps your call out definition in place.

3

Structure Setup

The Administration Dialog

The Administration dialog within **SMARTAnnotate** allows you to setup the configuration within a graphical user interface. It provides all the necessary tools to specify the required information and details.

Use the environment variable `SAN_ADMIN_MODE_ENABLED` and set it to `TRUE`. The **Administration** icon will appear in the **SMARTAnnotate** ribbon

1. Select **Administration** ▤ from the **SMARTAnnotate** tab

   • The **Administration** dialog opens
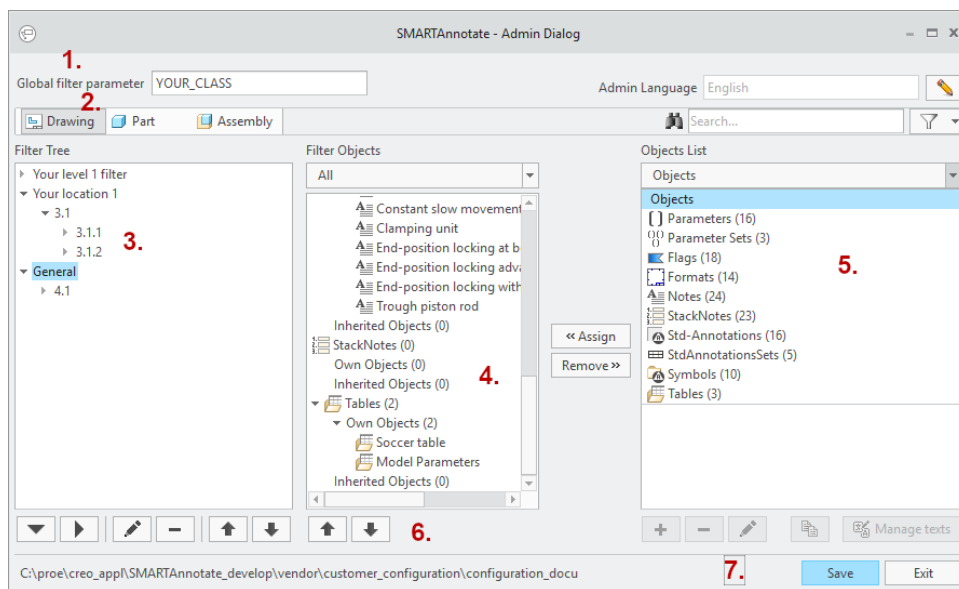
1. Global classification parameter
2. Mode
3. Filter structure
4. Assignment tree
5. Object list
6. Control area
7. Name of current configuration

> 💬 **Note**
>
> The environment variable `SAN_UI_LANG_FOR_CONTENTS` controls the display language of the content. (i.e. the note labels or the filter structure will be displayed in the specified language). Refer to the installation section for more details.

The Global Filter Parameter

SMARTAnnotate allows you to define a parameter to mark your drawings and models with a specific classification. You can use the parameter value for i.e. a customer designation, a location indication or a purpose indication. Or a combination of all.

You may use any parameter name. It is entered at the very top of the **Administration** dialog.
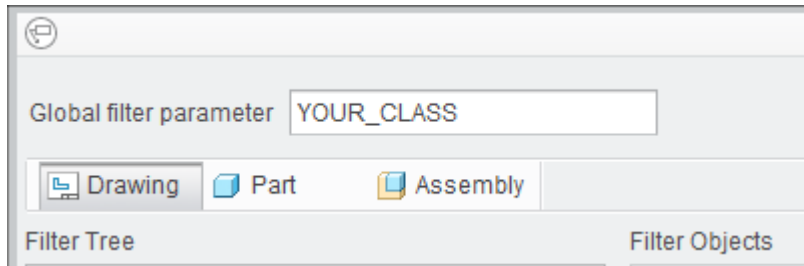
The corresponding values are entered during **Filter** definition.
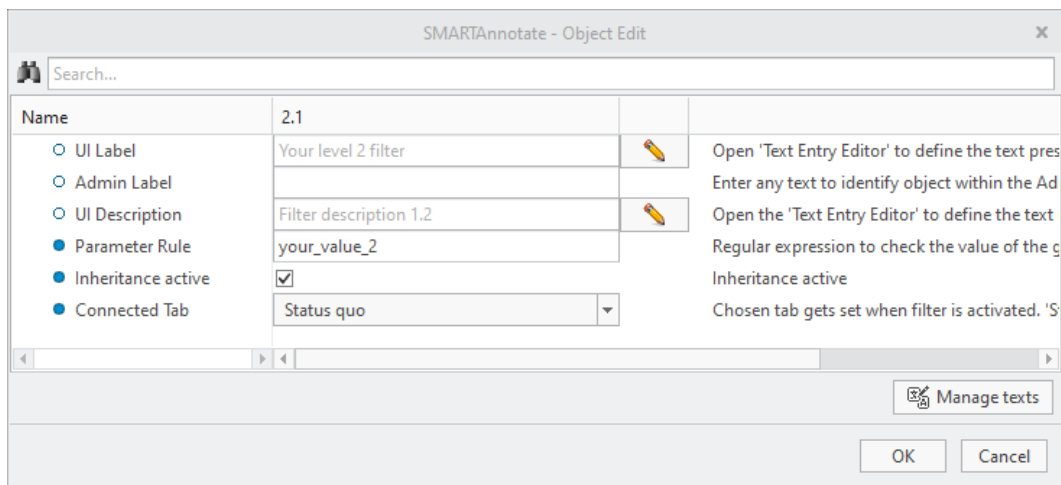
Important considerations:

- The value of the parameter needs to be <u>unique</u> throughout the whole filter tree.

- The entered value is a 'regular expression'

- A drawing does not have its own global filter parameter. It will look for this parameter within the active or designated drawing model.

- In the drawing filter tree you may add multiple values using the 'regular expressions' capabilities. (i.e. value_1|value_2|value_3 or value_\d+). This will allow you to have a single filter in the drawing tree supporting different model types.

If the user opens the **SMARTAnnotate** dialog, the software will look for this parameter and its value in the active or drawing model and will automatically set the correct filter tree location. This is indicated with a small star in front of the filter name. If the parameter is not found within the model or has an invalid value, the software will look for the last used setting from the AppData folder.

The Filter Structure

The filter structure is the main mechanism to distinguish different settings like notes, formats, symbols for user selection and display.

Your may assign any object to any tree position and inherit the object throughout the tree branch.

The tree is unlimited in its depth.
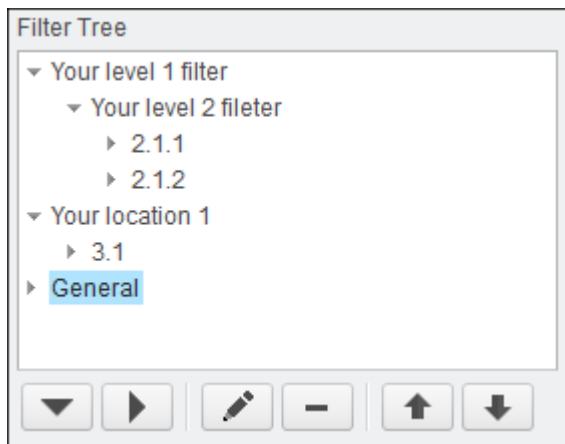
Create or edit a filter

1. Select **Add Sibling** to create a new filter node at the same levelorSelect **Add child** to create a new filter node below the current selectionorSelect **Edit** to modify any existing filter node

- The **Object Edit** dialog opens
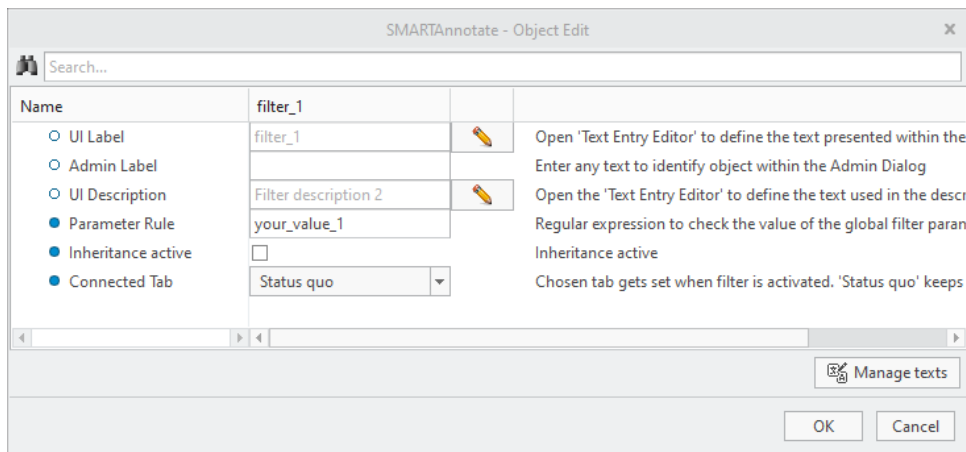
2. Select to enter the **UI Label** that is displayed in the user dialog.

3. Enter an **Admin Label** that is used for distinction within the administration dialog.

4. Select to enter a description that will be displayed in the user dialog. It is shown as a tooltip while hovering over the corresponding node within the filter tree.

5. Enter a **Parameter Value** for the classification parameter. See the for details.

6. Check **Inheritance active** if you want to inherit the assigned objects throughout the branch of your tree structure.

7. Select **OK** to close the dialog

Delete a filter

1. Mark any node within your tree structure and select **Delete**

   - A confirmation dialog appears.
   - Chose **OK** and the selected node including all children nodes will be removed from the structure

Reorder filter nodes

1. Mark any node within your tree structure and use or to reorder nodes or branches of your structure

4

Importing Content

Standard MS Excel Import

General

Besides creating **SMARTAnnotate** content from scratch you can use the **Import** functionality to bring in or manage your content. It can be used as initial bulk loader or as permanent process to manage texts and objects outside from Creo Parametric and **SMARTAnnotate**. Nevertheless the MS Excel interface has limitations.

Current limitations:

- Support for **Notes** and **Stack Notes** only
- A single hard-coded **Pattern file** can be used (default_pattern.txt) with a single prompt.
- The filter structure has to be built up within **SMARTAnnotate**
- Two level filter structure depth as maximum
- No **Flag** support

Workflow for initial bulk loading

1. Build up your intended filter structure with **SMARTAnnotate**
2. you want to import texts to
3. Fill the provided MS Excel template with your content

    - Make sure the columns for 'Catergory1' and 'Category2' match your **SMARTAnnotate** filter structure
    - Add columns for each language you want to support
    - Do not remove or rename the header cells
    - See the docu tab in the Excel sheet for more detailed information how to use variables in text

4. Make sure the configuration option *ENABLE_EXCEL_IMPORT* is set to *YES*.
5. Select from the Administration dialog

    - The **File Open** Dialog appears

6. Choose your prepared Excel Sheet

    - The **Import** will start immediately and can take some moments related to the amount of data you are importing.

SMART Annotate

Workflow for updating existing content

1. Add, modify our delete content in your MS Excel Sheet

> 📝 **Note**
>
> Revising objects is now your responsibility as the **Revision** information will be imported from the Excel sheet

---

2. Select from the Administration dialog
    *   The **File Open** Dialog appears
3. Choose your prepared Excel Sheet
    *   The **Import** will start immediately
    *   It will delete all existing content and rebuild the content from the Excel sheet.
    *   The process can take some moments related to the amount of data you are importing.

5

Creating Content

About Creating Content

Before assigning anything to the previously defined filter nodes, you need to create content. You can choose from the **Objects List** the type of content you want to create.
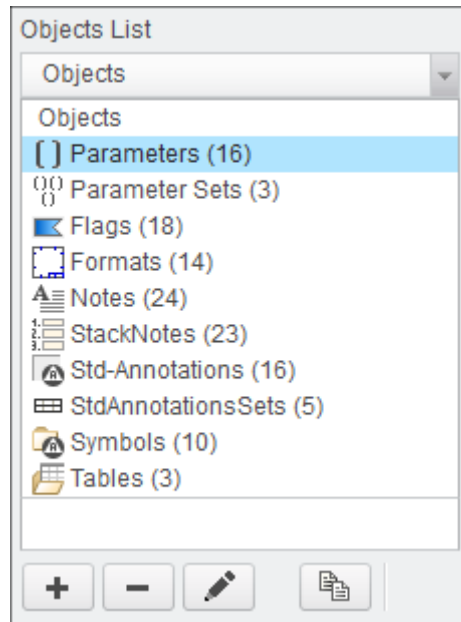
Manage Text Entries

To enter a new text entry or manage an existing one, the **SMARTAnnotate** provides a separate dialog. The dialog provides the ability to add texts in multiple languages and to use special formatting functions. This dialog is used for all text editing areas.

Add a new or modify an existing text

1. Select ⌞Manage texts⌟ from the main admininistration dialog or any object edit dialog, or select 🖉 next to a text input panel.
   - The **Text Entry Edit** dialog appears.
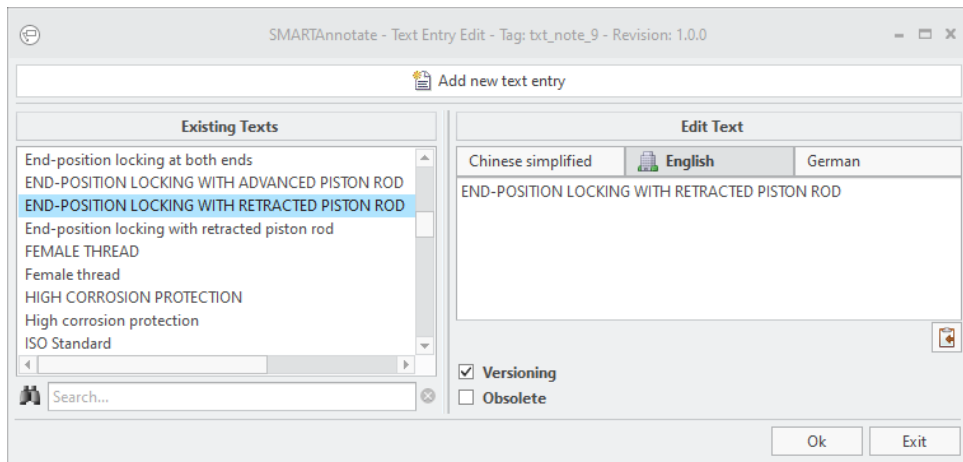
2. Select **Add new text entry** 📄 to create a new text or select an existing text entry from the **Existing Texts** list for modification.
3. (Optional) Switch language tabs to enter / modify the text in the current language.
4. Enter your Text.

Formatting text

- Return key — Creates a line break.

Options

1. **Versioning —** Marks the current object to be versioned by the software. How the versioning works on page 69.
2. **Obsolete** — Marks the current entry as obsolete. The entry will remain in the software, but not anymore selectable by the user.

Using variables within text

1. You may use the same syntax for adding a variable to your text as in symbols or formats.
2. Use \[V1]\ within your text at any position and the variable V1 will be called out during final placement. You need to have created an item in the **User Editable** section of your object.

Filter the Existing Texts list

- The list fills from different source files. Each object type has a separate list for **Existing Texts**.

Drawing Formats

The **Format** object contains a collection of a drawing format file (.frm) any number of tables (.tbl) and symbols (.sym) and Parameter Sets.
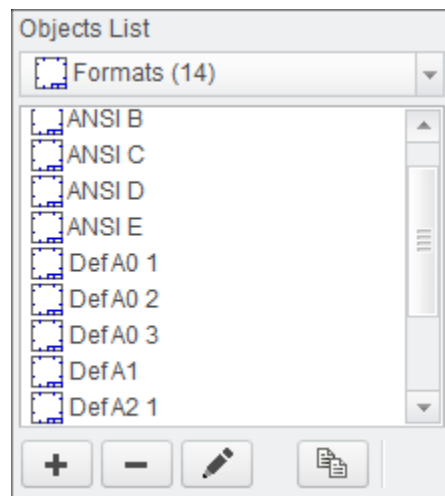
General Properties, Size and Drawing Format file

1. Select **Add new object** ➕ or select **Edit existing objects** ✏️ or use 📋 to copy an existing entry and modify from there.

    - The **Object Edit** dialog appears.
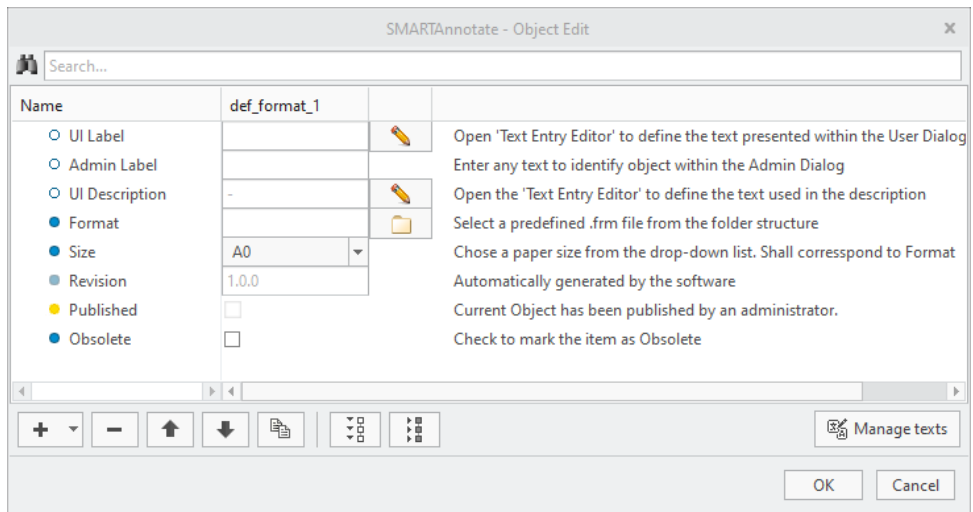
2. Select 🖊️ to enter the **UI Label** that is displayed in the user dialog.

3. Enter an **Admin Label** that is used for distinction within the administration dialog.

4. Select 🖊️ to enter a **UI Description** that will be displayed as tooltip in the user dialog, if you hover the mouse over the format name.

5. Select 📁 to choose a **Format**(.frm) from the directory /frm.

📝 Note

You may leave this field blank and work with symbols and tables only. Your drawing will not require a .frm file.

---

6. Choose an appropriate **Size** from the pull-down menu to match your frame file.

📝 **Note**

You can choose a **Custom** format size and enter the size values individually.

---

7. Open the **Add new elements** pull-down to add additional items (optional)

   • Add Format Table

   • Add Format Symbol

   • Add Selectable Symbol

   • Add Format Std-Annotation

   • Add Format Parameter Sets

Adding a Format Table

1. Choose **Add new element** ▶ **Add Format Table**

   • A new **Table** entry is created within the **Object Edit** dialog

▼ ⚠️ 🖿 Table		+ Table definition (Name)
⚠️ ● X-Position *		Enter X-Position placement ccordinates (drawing units)
⚠️ ● Y-Position *		Enter Y-Position placement ccordinates (drawing units)
⚠️ ● File Name *	📁	Select an existing Creo object
● Translate	☐	Check if this entity shall be translated
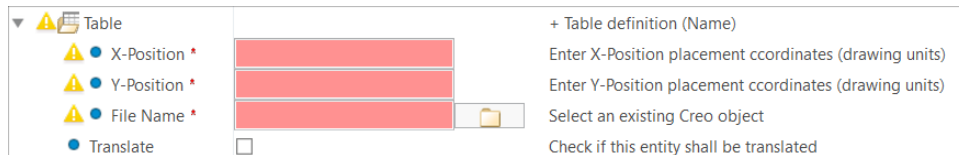
2. Enter **X and Y Position** of the table. The values are calculated in drawing units and are related to the origin of the table.

3. Select 📁 to choose a **Table** from the `/tbl` directory.

4. Check the **Translate** check mark, if you have text placeholders for multiple languages

Adding a Format Symbol

1. Choose **Add new element** ▶ **Add Format Symbol**

- A new **Symbol** entry is created within the **Object Edit** dialog

▼ ⚠ Symbol		+ Symbol definition (Name)
⚠ ● X-Position *		Enter X-Position placement ccordinates (drawing units)
⚠ ● Y-Position *		Enter Y-Position placement ccordinates (drawing units)
⚠ ● File Name *		Select an existing Creo object

2. Enter **X and Y Position** of the symbol. The values are calculated in drawing units and are related to the origin of the symbol.

3. Select 📁 to choose a **Symbol** from the /sym directory.

4. (Optional) Choose **Add new element ► Add Variable Text** while the symbol entry is highlighted

   - A new **Variable Text** entry is created within the Symbol entry. You may add as many **Variable Text** entries as you need.

▼ ⚠ Variable Text		+ Text definition
⚠ ● Prompt *		Enter the characters that were used in the symbol, table or pa
⚠ ● Text Entry *		Open 'Text Entry Editor' to create, chose or modify your text e

5. (Optional) Enter the **Prompt** that you have used as variable text in the symbol definition

6. (Optional) Select ✏ to enter a **Text Entry** that will be displayed within your symbol.

Adding a Selectable Format Symbol

1. Choose **Add new element ► Add Selectable Symbol**
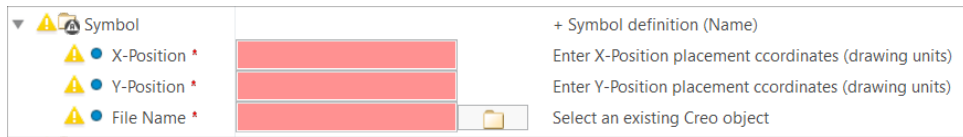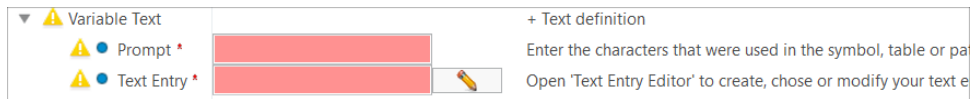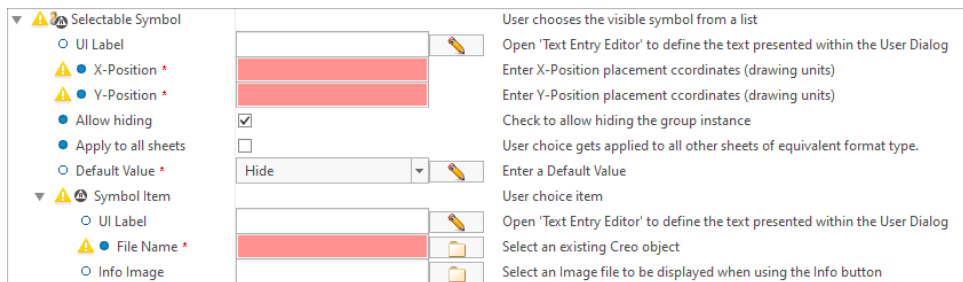
   - A new **Selectable Symbol** entry is created within the **Object Edit** dialog including a first Symbol instance definition.

▼ ⚠ Selectable Symbol		User chooses the visible symbol from a list
○ UI Label		Open 'Text Entry Editor' to define the text presented within the User Dialog
⚠ ● X-Position *		Enter X-Position placement ccordinates (drawing units)
⚠ ● Y-Position *		Enter Y-Position placement ccordinates (drawing units)
● Allow hiding	☑	Check to allow hiding the group instance
● Apply to all sheets	☐	User choice gets applied to all other sheets of equivalent format type.
○ Default Value *	Hide	Enter a Default Value
▼ ⚠ Symbol Item		User choice item
○ UI Label		Open 'Text Entry Editor' to define the text presented within the User Dialog
⚠ ● File Name *		Select an existing Creo object
○ Info Image		Select an Image file to be displayed when using the Info button

2. Select ✏ to enter or choose a **UI label**

3. Enter **X and Y Position** of the symbol. The values are calculated in drawing units and are related to the origin of the symbol.

4. Choose the **Allow hiding** option to allow the user to not show any symbol instance.

5. Choose the **Apply to all sheets** option to set this option for new drawings.

6. Enter a **Default Value** of the initial used symbol instance. You must use any of the **Symbol item** values of the subsequent symbol instances, or keep '-' for an empty default.

7. Expand the definition of the first **Symbol item**

8. Select ✏ to enter or choose a **UI label** for the Symbol item.

9. Select 📁 to choose a **Symbol** from the /symbols directory.

10. (Optional) Select 📁 to choose a **Info Images** from the /images folder.

11. (Optional) Choose **Add new element** ▶ **Add variable text**

   a. Enter the prompt used in the variable text section of the symbol.

   b. Select ✏ to enter or choose a text element.

12. (Optional) Choose **Add new element** ▶ **Add Symbol item** to add another symbol instance. Follow step 6 — 11.

Adding a Std-Annotation Set

1. Choose **Add new element** ▶ **Add Format Std-Annotation Set**

   • A new **StdAnnotationSet** entry is created within the **Object Edit** dialog

Std-Annotation Set			+ Std-Annotation Set definition
● Name	Manufacturing advices	▼	Chose a Name from the drop-down list
▼ Column			+ Column definition
● X-Position (Top-Down) *	1081		Enter X-Position placement coordinates (Top-Down) in drawing units
● Y-Position (Top-Down) *	700		Enter Y-Position placement coordinates (Top-Down) in drawing units
● Column break (Top-Down)	100		Enter the Column break coordinates (Top-Down) in drawing units.
● X-Position (Bottom-Up) *	1081		Enter X-Position placement coordinates (Bottom-Up) in drawing units
● Y-Position (Bottom-Up) *	80		Enter Y-Position placement coordinates (Bottom-Up) in drawing units
● Column break (Bottom-Up)	700		Enter the Column break coordinates (Bottom-Up) in drawing units.

2. Choose a **Name** of an existing Std-Annotation Set object from the pull-down list

3. Enter the **X and Y coordinates** for the stack placement (bottom left corner) and define a **Column break** coordinate for both stacking methods: Top-Down and Bottom-Up

📝 **Note**

You may also define the position and the column coordinate with your Std-Annotation object.

Adding a Parameter Set

1. Choose **Add new element** ► **Add Format Parameter Set**

   • A new **ParameterSet** entry is created within the **Object Edit** dialog

⊙⊙ Parameter Set		Parameter Set
● Name	Parameter Set 1 ▼	Chose a Name from the drop-down list

2. Choose a **Name** of an existing Parameter Set object from the pull-down list
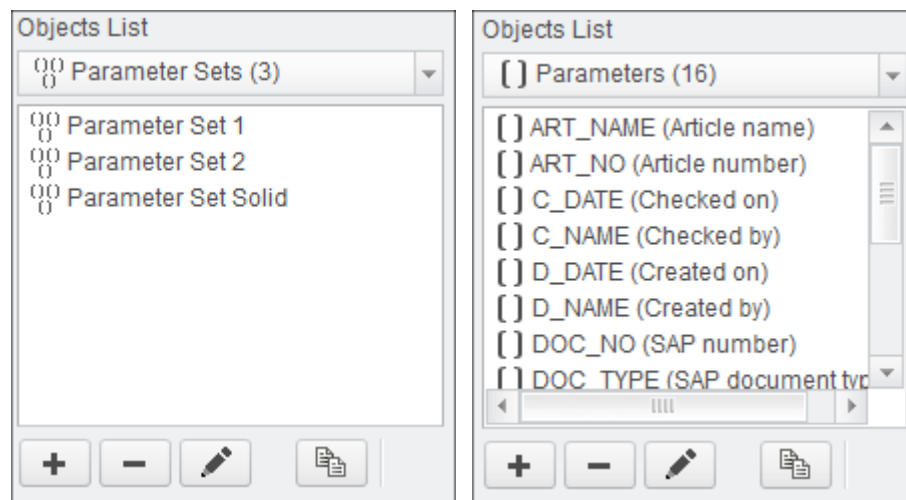

> 📝 **Note**
>
> This Parameter Set will be added to the Parameter Sets that you directly assign to a filter node. Duplicate entries will be filtered automatically.
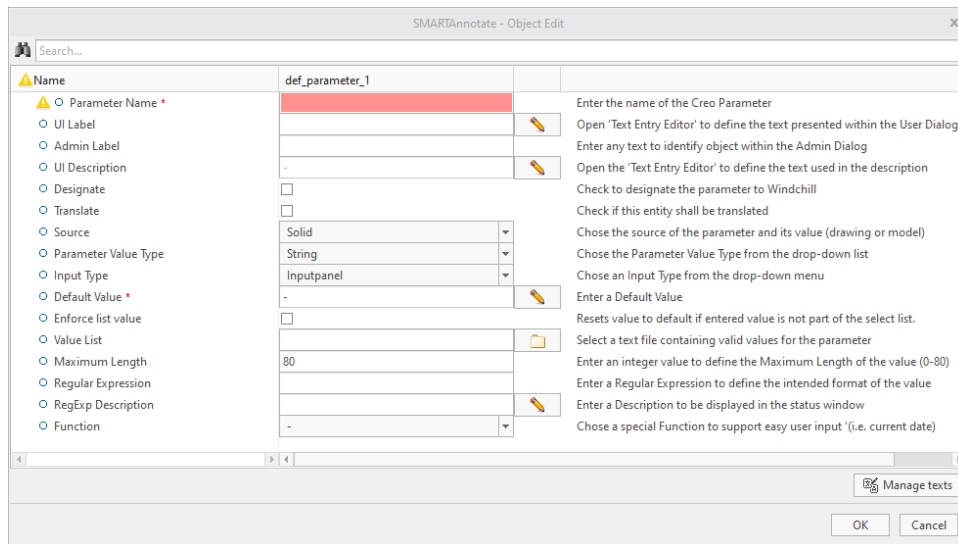
Parameters and Parameter Sets

The **Parameter** object contains the definition of a parameter that can be used in drawing mode or 3D mode. The **Parameter Set** is a collection of **Parameters**.

Objects List	Objects List
⊙⊙ Parameter Sets (3) ▼	[ ] Parameters (16) ▼
⊙⊙ Parameter Set 1	[ ] ART_NAME (Article name)
⊙⊙ Parameter Set 2	[ ] ART_NO (Article number)
⊙⊙ Parameter Set Solid	[ ] C_DATE (Checked on)
	[ ] C_NAME (Checked by)
	[ ] D_DATE (Created on)
	[ ] D_NAME (Created by)
	[ ] DOC_NO (SAP number)
	[ ] DOC_TYPE (SAP document typ
+ − ✏ 📋	+ − ✏ 📋

Adding Parameters

1. Select **Add new object** ![plus icon] or select **Edit existing objects** ![pencil icon] or use ![copy icon] to copy an existing entry and modify from there.

   - The **Object Edit** dialog appears.

2. Type in the **Creo Parameter** name that you want to call out.

3. Select ![pencil icon] to enter the **UI Label** that is displayed in the user dialog.

4. Enter an **Admin Label** that is used for distinction within the administration dialog.

5. Select ![pencil icon] to enter a **UI Description** that will be displayed in the **Description** column in the user dialog.

6. Check the **Designate** check box if this parameter shall be designated to Windchill.

7. Check the **Translate** check box if the value of this parameter shall be translated. For each configuration language an unique instance is automatically created using the parameter name as a basis and a language specific suffix. E.g. if the parameter name is set to "MY_PARAM" and the translate attribute is set, the parameters "MY_

PARAM_EN", "MY_PARAM_DE" and "MY_PARAM_CN" could be created and updated. If the user changes the parameter value via drop down list in the user interface, the corresponding translation for the chosen value is applied to all language instances of the parameter. The suffixes are defined for each language within the corresponding xml file (<configuration directory>\translations\<language name>\san_language_settings.xml) under the node "<param_suffix>".

8. Choose the **Source** of the parameter from the pull-down menu.Use **Solid** to call out parameter values from the 3D model or **Drawing** to call out values from the 2D drawing.

9. Choose the **Parameter Value Type** from the pull-down menu.

10. Choose the Input Type from the pull-down menu to define how the user can edit the parameter value. You can choose from the following options:

    • Read-Only — Value is displayed, but user cannot change

    • Inputpanel — User can type in a value

    • Optionmenu — User can select from pull-down menu only

    • Optionmenu (editable) — User can select from pull-down menu or can type in a value

11. Enter a **Default Value**. Use '-' for no value.

12. (Optional) You may select 📁 to specify a **Value List** (text file) from the `/lists` folder. You also can enter a semicolon separated string.

13. Enter the **Maximum Length** of the parameter value. The maximum length that Creo allows is 80 characters. The user will only be able to enter characters up to the specified length.

14. (Optional) Enter a **Regular Expression** to specify the format of the value. Refer to the Internet (i.e. Wikipedia) for explanations.

15. (Optional) Enter a **RegExp Description** to be displayed in the **Parameter Status Window**. This helps the user to understand when the entered value does not match the regular expression.

16. (Optional) Choose a **Function** from the pull-down menu. Currently two functions are available.

    • Login name — the value of the 'USERNAME' environment variable

    • Today's date — the current date is used (format is set in the configuration)

Adding Parameter Sets

1. Select **Add new object** ⊞ or select **Edit existing objects** ✏ or use 📑 to copy an existing entry and modify from there.

    • The **Object Edit** dialog appears.
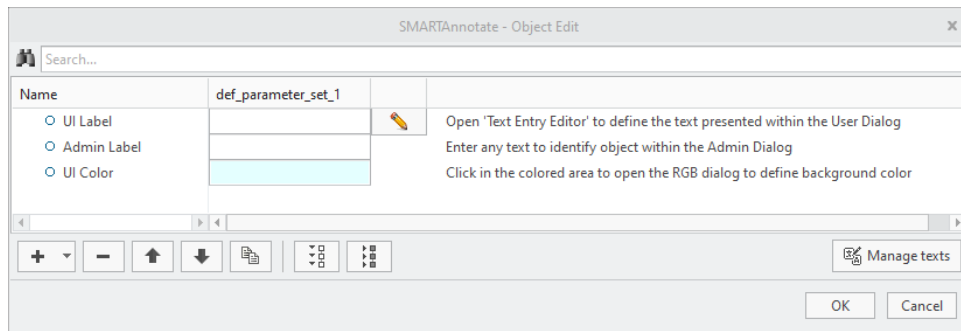
2. Select ✎ to enter the **UI Label** that is displayed in the admin dialog.

3. (Optional) Enter an **Admin Label** that is used for distinction within the administration dialog.

4. (Optional) Click in the color bar and the Color UI dialog opens. Define a background color for the current Parameter Set within the user dialog.
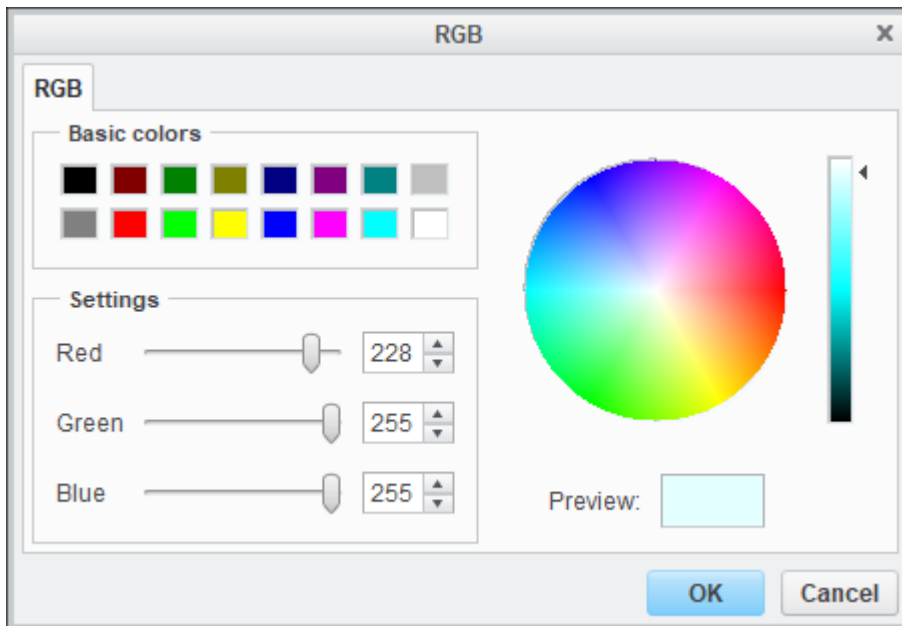
5. Choose **Add new element ▶ Add Parameter**

- A new **Parameter** entry is created within the **Object Edit** dialog

6. Choose an existing Parameter from the pull-down menu.
7. Repeat for all intended parameters in the set.

Std-Annotations and Std-Annotation Sets

The **Std-Annotation** object contains the definition of an annotation that can be used in drawing mode or 3D mode. The technical means is a Creo Symbol. Multiple symbols can be used in one Std-Annotation definition.The **Std-Annotation Set** is a collection of **Std-Annotations**.
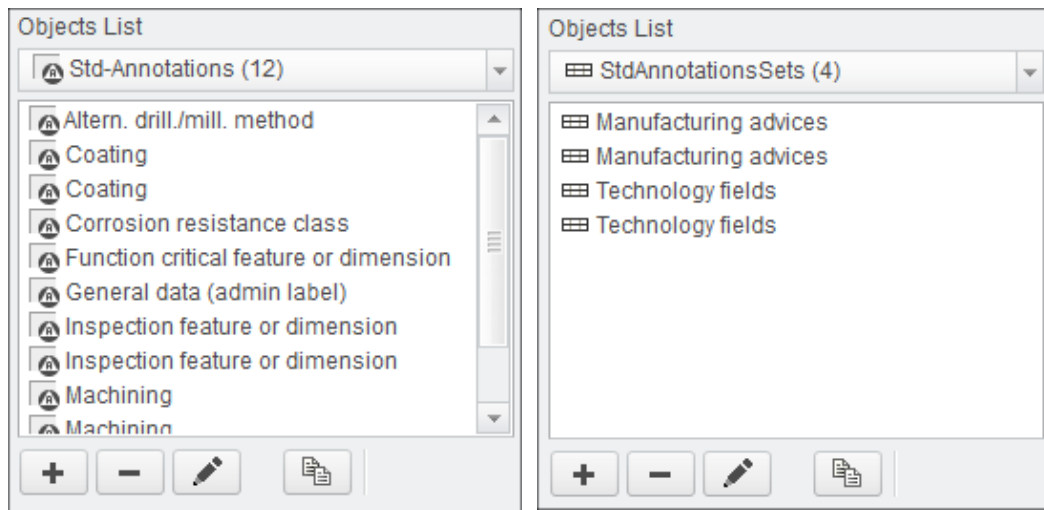
How it works

A **Std-Annotation** set is automatically placed on the drawing or within the 3D model. The placement coordinates are defined during the definition of the **Std-Annotation Set** or in the **Drawing Forma**t.

The selected **Std-Annotations** of an **Std-Annotation Set** will be stacked up at the defined location. You can use the Bottom-Up or Top-Down method for stacking.

You may use a **Separator** and/or a **Margin** between the single **Std-Annotation** definitions.

Any kind of Creo symbol is valid.

Adding Std-Annotations

1. Select **Add new object** ⊞ or select **Edit existing objects** ✎ or use ▤ to copy an existing entry and modify from there.

   - The **Object Edit** dialog appears.
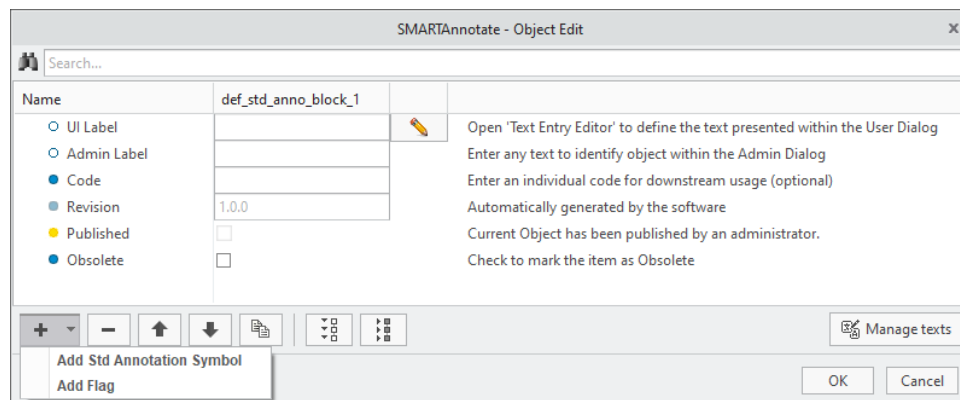
2. Select ✎ to enter the **UI Label** that is displayed in the admin dialog.

3. (Optional) Enter an **Admin Label** that is used for distinction within the administration dialog.

4. (Optional) Enter a **Code** (not implemented yet)

5. The **Revision** will be iterated automatically every time you leave the definition dialog with OK and save the configuration. See the **Versioning Objects** on page 69 section in the Appendix for more details-

6. Check the **Obsolete** check box if you want to disable the Std-Annotation. It will remain within the configuration, but not anymore available for user selection.

7. Choose **Add new element** ⊞ ▶ **Add Std Annotation Symbol**

   • A new **Std Annotation Symbol** entry is created within the **Object Edit** dialog

⚠🔲 Symbol		+ Symbol definition (Name)
⚠ ● File Name *	📁	Select an existing Creo object
● Code		Enter an individual code for downstream usage (optional)
User independent		+ Entry definition without user interaction
👤 User-Editable		+ Entry definition requiring user interaction

---

💬 **Note**

See the Symbol definition chapter on page 55 for a detailed description of all additional symbol definition options.

---

8. (Optional) Choose **Add new element** ⊞ ▶ **Add Flag**

   • A new **Flag** entry is created within the **Object Edit** dialog. For every flag symbol that you want to display in the UI pull-down you have to create a separate **Flag** entry.

▼ ◤ Flag		+ Flag definition (Symbol used)
● Name	def_flag_1 ▼	Chose a Name from the drop-down list
● Default	☐	Check to mark this flag to be selected by default
● Mandatory	☐	Check to mark this flag as mandatory for placement
○ Default Value *	- ✏	Enter a Default Value

9. Check the **Default** check box to indicate which flag is the default one.

10. Check the **Mandatory** check box to define if the user can deactivate the flag.

11. Enter the **Default Valu**e, if you have a variable text within the underlying symbol.

Adding Std-Annotation Sets

1. Select **Add new object** ✚ or select **Edit existing objects** ✏ or use 📄 to copy an existing entry and modify from there.

   • The **Object Edit** dialog appears.
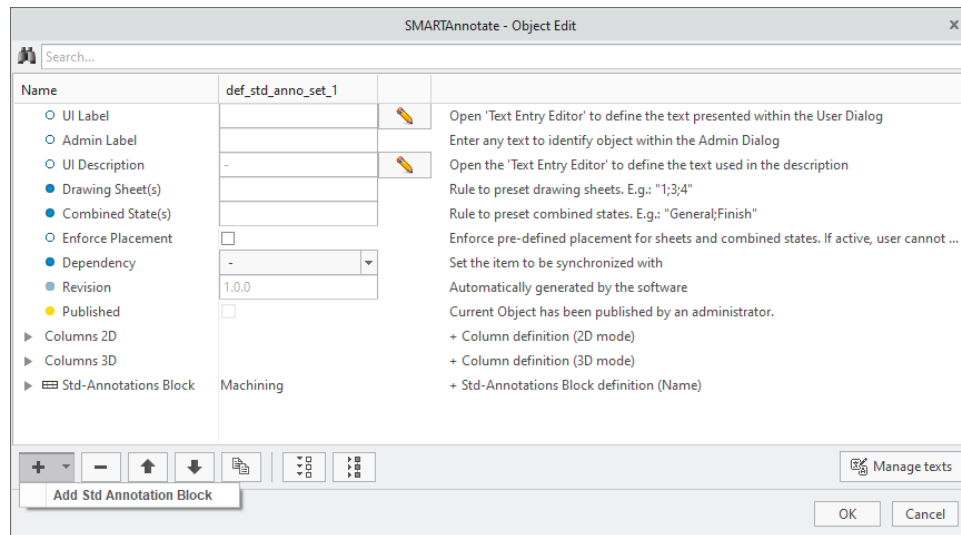
2. Select ✏ to enter the **UI Label** that is displayed in the user dialog.
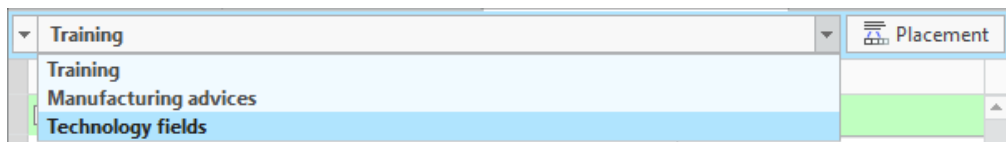
3. (Optional) Enter an **Admin Label** that is used for distinction within the administration dialog.

4. (Optional) Enter a **UI Description**.

5. Define a rule to auto-assign the Std-Annotation Set to **Drawing Sheet**s via index. The first sheet has index 1, second has index 2 etc. The Std-Annotation Set will be automatically placed on these sheets.

   Examples:

   • "1-3;5" = 1,2,3,5

   • "*" = ∞

   • "3-*" = 3 to ∞

   • "*-3" = 1,2,3

   • "0" = 1
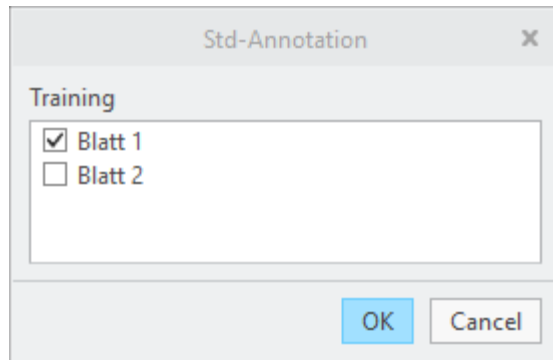
6. Enter **Combined State** names as semicolon (;) separated list which you automatically want to place the Std-Annotation Set in (3D mode only). To address all combined states use "*".
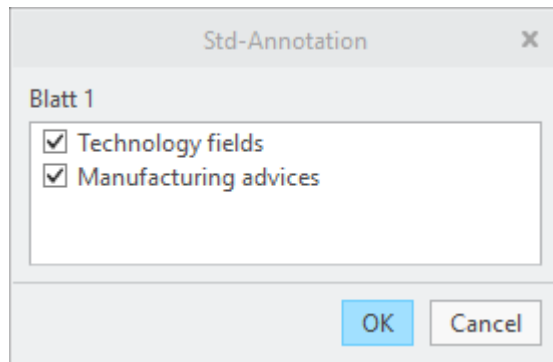
7. Define the placement of the Std-Annotation stack

   - The stack can be build using the top-down or bottom-up direction. The behavior is globally defined in the configuration file and can distinguish between 3D and 2D mode.
   - The stack will be broken into the next column, if the breakpoint coordinates are reached while building the stack.
   - The specified coordinate values refer to the origin of the used Creo symbol.

- 2D placement (**2D Columns**)

  - The values entered here only take effect, if there are no values entered in the format definition.
  - You can specify multiple columns for every used format size
  - The coordinates are entered in drawing units

- 3D placement (**3D columns**)

  - You can specify multiple columns for the use in a combined state.
  - The coordinates are entered in screen coordinates.

    The coordinates range vertically from 0 to 840 and horizontally from 0 to 1000 starting at the bottom left corner of the graphics area. They are independent from the screen resolution.

8. (Optional) If you have defined custom size formats, you can add those formats to the list for entering the placement coordinates.
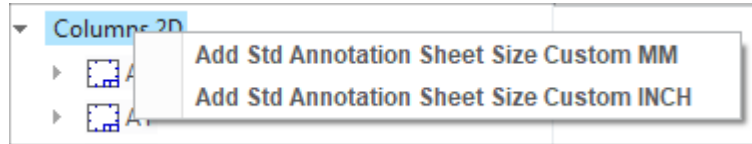
9. Choose **Add new element** ➕ ▶ **Add Std Annotation Block**

   - A new **Std Annotation Block** entry is created within the **Object Edit** dialog

10. Choose a **Name** of an existing Std-Annotation from the pull-down menu.

11. Choose a **Necessity** from the pull-down menu.

   - mandatory — this entry is pre-selected in the UI and cannot be deselected by the user

   - required — this entry is preselected in the UI, but may be deselected by the user

   - optional — this entry is not pre-selected in the UI

   You can define the background colors of the three options in the configuration dialog.

12. Check the **Set Flag** check box to allow a flag to the Std-Annotation definition.

13. Enter an **Offset** that is used to generate space between the current and the next definition. Default is 0.

14. Check the **Separator** check box to use a separator symbol between Std-Annotation definitions. The Separator symbol is defined by the config option `SEPARATOR_SYMBOL` and can contain any kind of entities.

15. Repeat for all intended Std-Annotation entities in the set.

Notes

The **Notes** object contains the definition of an annotation that can be used in drawing mode or 3D mode. The technical means is a Creo Note.
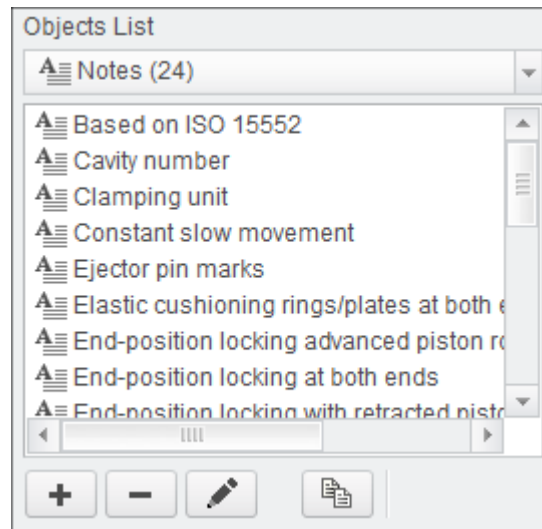
How it works

A **Note** is manually placed on the drawing or within the 3D model. The placement coordinates are defined by user placement. The selected note entities are combined to a single note and can be placed a single item.

Adding Notes

1. Select **Add new object** ➕ or select **Edit existing objects** ✏ or use 🗎
   to copy an existing entry and modify from there.

   • The **Object Edit** dialog appears.
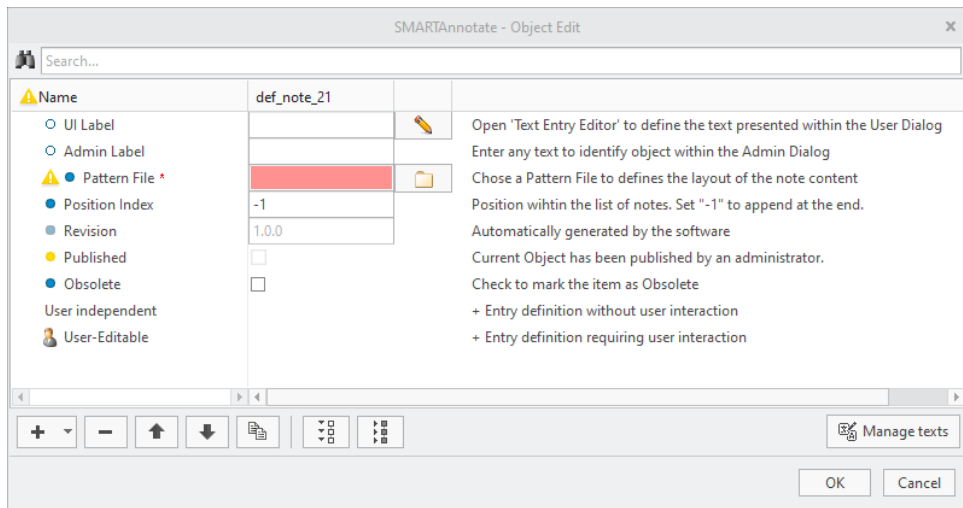
2. Select ✏ to enter the **UI Label** that is displayed in the admin dialog.

3. (Optional) Enter an **Admin Label** that is used for distinction within the
   administration dialog.

4. Select 📁 to chose a **Pattern File** from the /patterns folder.See the
   Pattern file section on page 159 to learn how to create and use it.

5. The **Revision** will be iterated automatically every time you leave the
   definition dialog with OK and save the configuration. See the
   Versioning Objects on page 69 section in the Appendix for more
   details-

6. Check the **Obsolete** check box if you want to disable the Std-
   Annotation. It will remain within the configuration, but not anymore
   available for user selection.

7. For a simple text note without any user interaction select the **User
   Independent** section and chose **Add new element** ➕ ► **Add Text**

   • A new **Text** entry is created within the **Object Edit** dialog
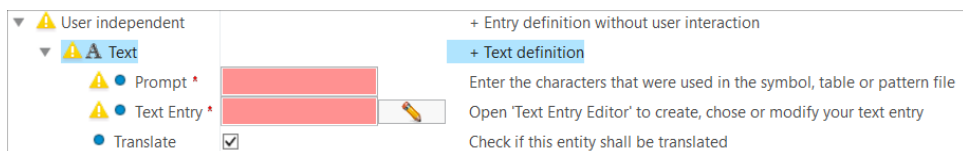
8. Enter the same **Prompt** that was used in the Pattern file.

9. Select ✎ to enter the **Text Entry** representing your final text.

10. Check the **Translate** check box, if this text shall be translated.

11. For a <u>text note that requires user interaction</u> (i.e. enter a value) select the **User Editable** section and chose **Add new element** ✚ ► **Add Value**

    • A new **Value** entry is created within the **Object Edit** dialog. For every value that you want to be display in the UI you have to create a separate **Value** entry.

▼ ⚠👤 User-Editable		+ Entry definition requiring user interaction
▼ ⚠ Value		Enter any text as a predefined value
⚠ ● Prompt *		Enter the characters that were used in the symbol, table or pattern file
○ UI Label	✎	Open 'Text Entry Editor' to define the text presented within the User Dialog
● Translate	☑	Check if this entity shall be translated
● Input Type	Inputpanel ▼	Chose an Input Type from the drop-down menu
○ Default Value *	- ✎	Enter a Default Value
● Dependent symbol g		List of symbol group names (";"-separated). Value is just visible to the user
● Maximum Length	80	Enter an integer value to define the Maximum Length of the value (0-80)
● Regular Expression		Enter a Regular Expression to define the intended format of the value
○ RegExp Description	✎	Enter a Description to be displayed in the status window

12. Enter the same **Prompt** that was used in the Pattern file.

13. Select ✎ to enter the **UI Label** that is displayed in the user dialog.

14. Check the **Translate** check box, if this text shall be translated.

15. Choose the Input Type from the pull-down menu to define how the user can edit the parameter value. You can choose from the following options:

    • Read-Only — Value is displayed, but user cannot change

    • Inputpanel — User can type in a value

    • Optionmenu — User can select from pull-down menu only

    • Optionmenu (editable) — User can select from pull-down menu or can type in a value

16. Enter a **Default Value**. Use '-' for no value.

17. If you have selected **Optionmenu** or **Optionmenu (editable)** from the **Input Type** pull-down, you need to add value items to be selected from the pull-down. Choose **Add new element** ✚ ► **Add Text**

    • A new **Item** entry is created within the **Object Edit** dialog

▼ Item		+ Item definition
● Value	✎	Enter any text as a predefined value
○ UI Label	✎	Open 'Text Entry Editor' to define the text presented within the User Dialog

18. Enter a **Value** and repeat until you have all intended values

Stack Notes

The **Stack Notes** object contains the definition of an annotation that can be used in drawing mode or 3D mode like a regular **Note**. The difference to a regular **Note** is that SMARTAnnotate will keep track on this type of notes. The technical means are multiple Creo Notes.
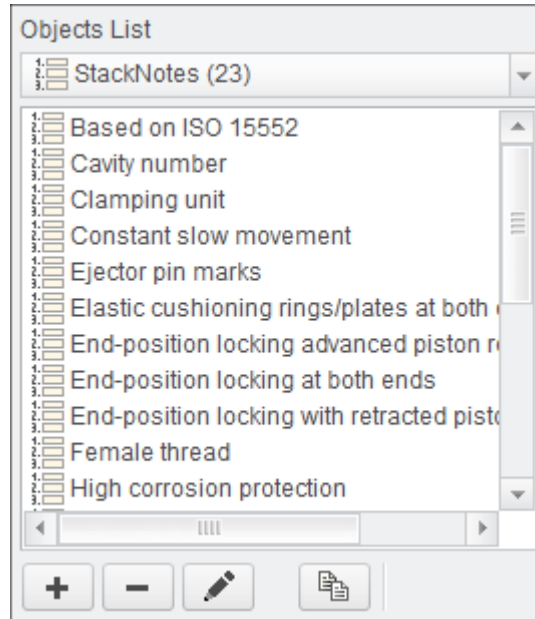
How it works

Stack Notes are manually placed as one or multiple columns on the drawing or within the 3D model. The placement coordinates are defined by user placement.

After selecting the **Stack Notes** from the user dialog and hit the **Add** button you will see a dynamic preview of the notes stack.

- Move down your mouse over the note entities and they will appear each by each if you keep moving the mouse
- Use LMB to finish the first column and use LMB again to place the second column.
- Proceed until you have placed all notes.

Adding Stack Notes

1. Select **Add new object** ⊞ or select **Edit existing objects** ✎ or use 🗎
    to copy an existing entry and modify from there.

    *   The **Object Edit** dialog appears.
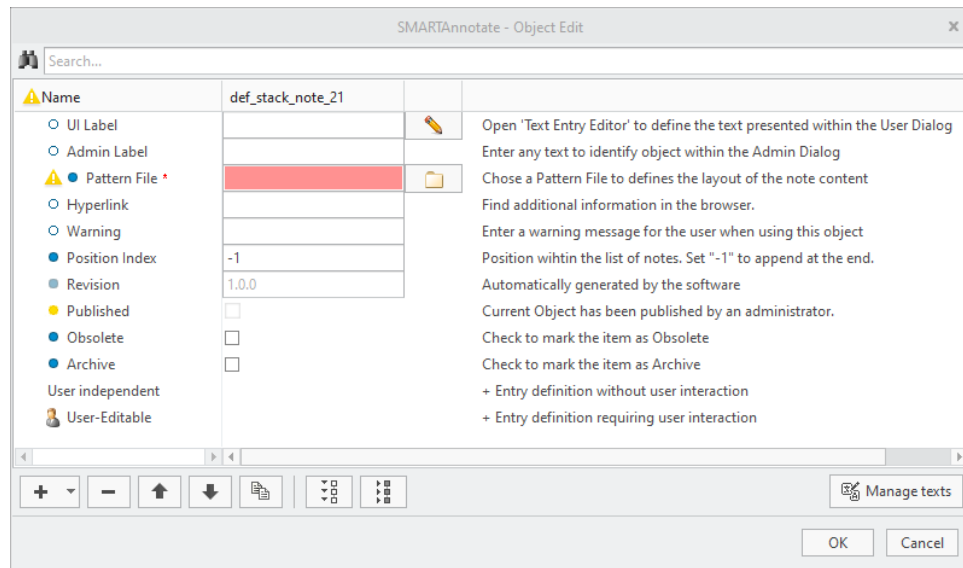
2. Select 🖉 to enter the **UI Label** that is displayed in the admin dialog.
3. (Optional) Enter an **Admin Label** that is used for distinction within the
    administration dialog.
4. Select 📁 to choose a **Pattern File** from the /patterns folder.See the
    Pattern file section on page 159 to learn how to create and use it.
5. Enter a **Hyperlink** here to any Website in the Internet or Intranet. From
    the **Info column** in the User dialog you can access this website and it
    will display in the internal Creo browser. This will not be a link from the
    graphics window.

    ### 📝 Note

    You must set the configuration option in the Stack Notes tab SHOW_
    HYPERLINK_COLUMN to YES.

6. Enter a fixed **Position** for this note in the stack here. Default is '-1' and
    places the note at the bottom of the stack in the User dialog.
7. The **Revision** will be iterated automatically every time you leave the
    definition dialog with OK and save the configuration. See the

8. Check the **Obsolete** check box if you want to disable the **Stack Not**e. It will remain within the configuration, but not anymore available for user selection. The user will be informed by the Notification center if an obsolete note is found on an existing drawing or model to take care of it. No new revisions are created from this object.

9. Check the **Archive** check box if you want to disable the the Stack Note. In this case it also will not be available for user selection but there will be no warning from the Notification Center. No new revisions are created from this object.

10. For a simple <u>text note without any user interaction</u> select the **User Independent** section and choose **Add new element** ➕ ▶ **Add Text**

   • A new **Text** entry is created within the **Object Edit** dialog
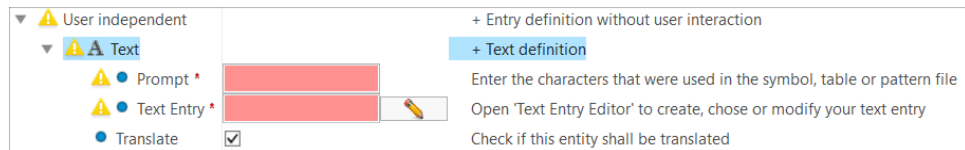
11. Enter the same **Prompt** that was used in the Pattern file.

12. Select 🖉 to enter the **Translation Tag** representing your final text.

13. For a <u>text note that requires user interaction</u> (i.e. enter a value) select the **User Editable** section and choose **Add new element** ➕ ▶ **Add Value**

   • A new **Value** entry is created within the **Object Edit** dialog. For every value that you want to be display in the UI you have to create a separate **Value** entry.
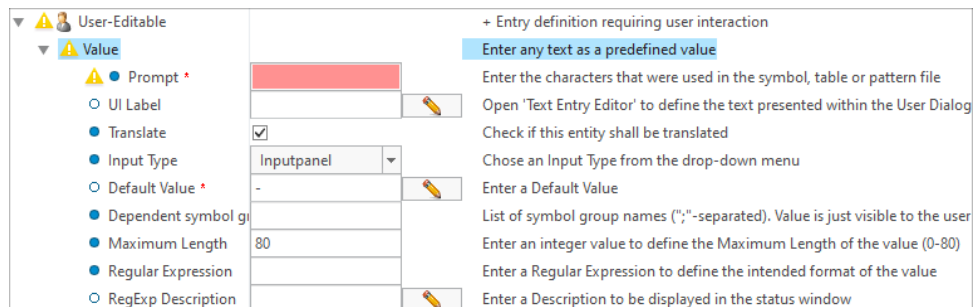
14. Enter the same **Prompt** that was used in the Pattern file.

15. Select 🖉 to enter the **UI Label** that is displayed in the user dialog.

16. (Optional) Enter an **Admin Label** that is used for distinction within the administration dialog.

17. Choose the Input Type from the pull-down menu to define how the user can edit the parameter value. You can choose from the following options:

   • Read-Only — Value is displayed, but user cannot change

   • Inputpanel — User can type in a value

   • Optionmenu — User can select from pull-down menu only

   • Optionmenu (editable) — User can select from pull-down menu or can type in a value

18. Enter a **Default Value**. Use '-' for no value.

19. If you have selected **Optionmenu** or **Optionmenu (editable)** from the **Input Type** pull-down, you need to add value items to be selected from the pull-down. Choose **Add new element** ➕ ▶ **Add Text**

   • A new **Item** entry is created within the **Object Edit** dialog
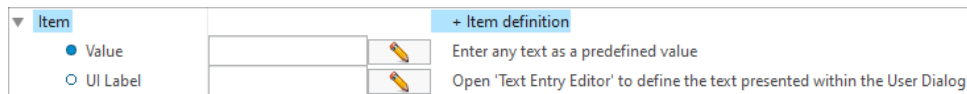
20. Enter a **Value** and repeat until you have all intended values

Flags

The **Flag** object contains the definition of a flag that can be used in drawing mode or 3D mode. The technical means is a Creo Symbol.
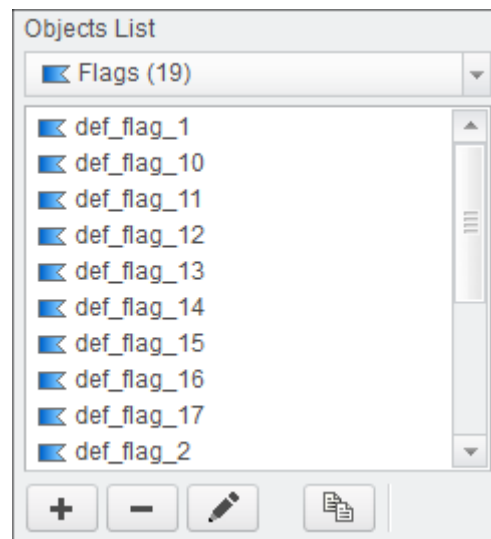
How it works

A **Flag** can be associated to **Std-Annotations** or **Notes**. It is usually used as a visual link between a text and a graphical entity. When adding a **Flag** to a **Std-Annotation** or a **Note** it will automatically appear as icon in the Flag column in the respective annotation list. The user will be prompted to place the flag.

Flags are monitored by the software to make sure that neither the text nor the flag is missing.

Adding Flags

1. Select **Add new object** ⊞ or select **Edit existing objects** 🖉 or use 📄 to copy an existing entry and modify from there.

   - The **Object Edit** dialog appears.
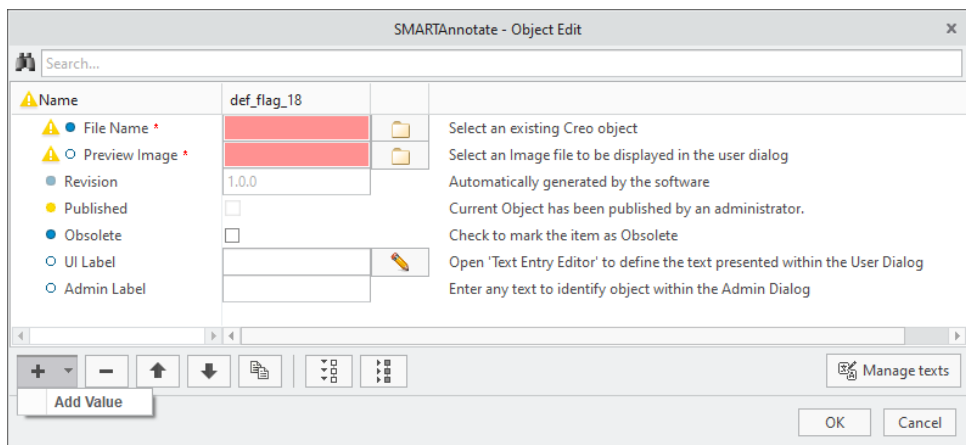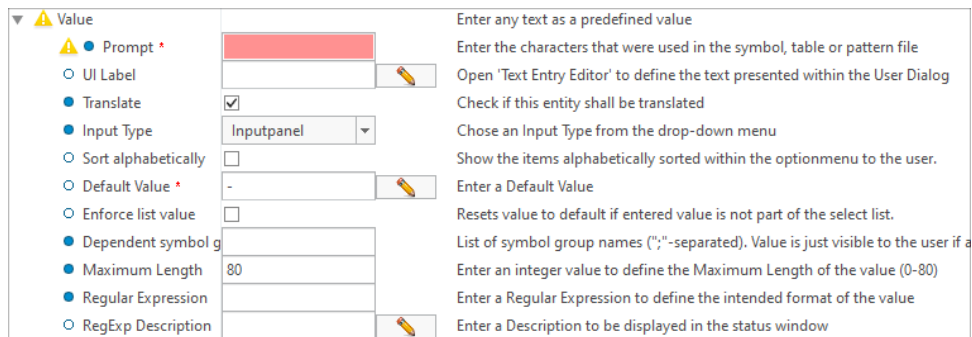
2. Select 📁 to choose a **File Name** of a symbol from the /sym folder.

3. Select 📁 to choose a **Preview Image** from the /images folder.This images will be displayed in the user dialog.

4. The **Revision** will be iterated automatically every time you leave the definition dialog with OK and save the configuration. See the **Versioning Objects** on page 69 section in the Appendix for more details-

5. Check the **Obsolete** check box if you want to disable the symbol. It will

remain within the configuration, but not anymore available for user selection.

6. (Optional) For a <u>flag that requires user interaction</u> (i.e. enter a value) choose **Add new element** ![+] ▶ **Add Value**

- A new **Value** entry is created within the **Object Edit** dialog. For every value that you want to be display in the UI you have to create a separate **Value** entry.

▼ ⚠ Value		Enter any text as a predefined value
⚠ ● Prompt *		Enter the characters that were used in the symbol, table or pattern file
○ UI Label	✎	Open 'Text Entry Editor' to define the text presented within the User Dialog
● Translate	☑	Check if this entity shall be translated
● Input Type	Inputpanel ▼	Chose an Input Type from the drop-down menu
○ Sort alphabetically	☐	Show the items alphabetically sorted within the optionmenu to the user.
○ Default Value *	- ✎	Enter a Default Value
○ Enforce list value	☐	Resets value to default if entered value is not part of the select list.
● Dependent symbol g		List of symbol group names (";"-separated). Value is just visible to the user if a
● Maximum Length	80	Enter an integer value to define the Maximum Length of the value (0-80)
● Regular Expression		Enter a Regular Expression to define the intended format of the value
○ RegExp Description	✎	Enter a Description to be displayed in the status window

a. Enter the same **Prompt** that was used in the variable text of the symbol.

b. Select ![pencil] to enter the **UI Label** that is displayed in the user dialog.

c. (Optional) Enter an **Admin Label** that is used for distinction within the administration dialog.

d. Check the **Translate** check box to allow text translation.

e. Choose the **Input Type** from the pull-down menu to define how the user can edit the parameter value. You can choose from the following options:

- Read-Only — Value is displayed, but user cannot change
- Inputpanel — User can type in a value
- Optionmenu — User can select from pull-down menu only
- Optionmenu (editable) — User can select from pull-down menu or can type in a value

f. Enter a **Default Value**. Use '-' for no value.

g. If you have selected **Optionmenu** or **Optionmenu (editable)** from the **Input Type** pull-down, you need to add value items to be selected from the pull-down. Choose **Add new element** ![+] ▶ **Add Item**

- A new **Item** entry is created within the **Object Edit** dialog

▼ Item		+ Item definition
● Value	✎	Enter any text as a predefined value
○ UI Label	✎	Open 'Text Entry Editor' to define the text presented within the User Dialog
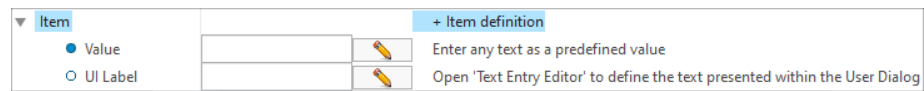
h. Enter a **Value** and repeat until you have all intended values

Symbols

The **Symbols** object contains the definition of a graphical symbol that can be used in drawing mode or 3D mode. The technical means is a Creo Symbol.

Objects List

Symbols (9)

- B&W Software GmbH
- Basic Geometries
- Battery location
- Force vector
- General tolerances for linear dimensions
- On / Off
- Power OFF
- Power ON
- Stand By

\+ − ✎ ⧉
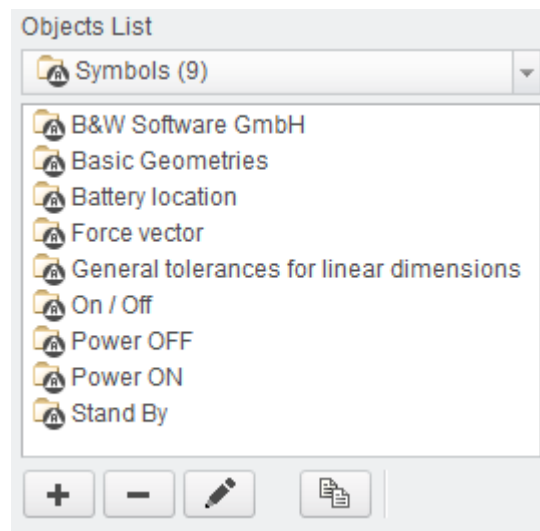
How it works

A **Symbol** can be graphically selected from the dialog and is manually placed on the drawing or within the 3D model. The placement coordinates are defined by user placement.

Adding Symbols

1. Select **Add new object** ➕ or select **Edit existing objects** ✏️ or use 📑 to copy an existing entry and modify from there.

   - The **Object Edit** dialog appears.

   ![SMARTAnnotate - Object Edit dialog]
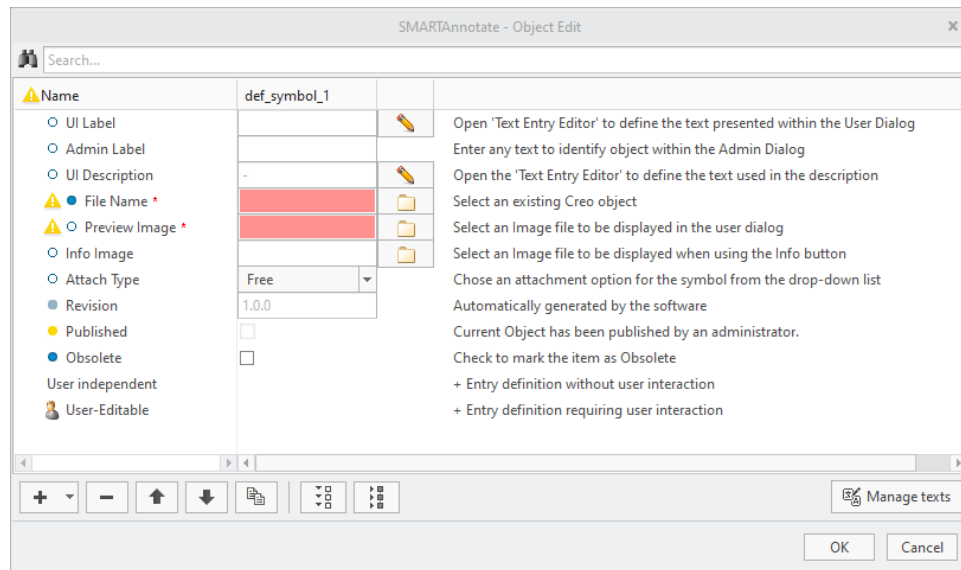
2. Select ✏️ to enter the **UI Label** that is displayed in the admin dialog.

3. (Optional) Enter an **Admin Label** that is used for distinction within the administration dialog.

4. Select ✏️ to enter the **UI Description** that is displayed in the user dialog.

5. Select 📁 to choose a **File Name** of a symbol from the `/sym` folder.

6. Select 📁 to choose a **Preview Image** from the `/images` folder.This images will be displayed in the user dialog for easy user selection.

7. Select 📁 to choose a **Info Image** from the `/images` folder.This image will pop up when using the **Info** button within the user dialog.

8. Choose an **Attach Type** from the pull-down menu. This will define the default behaviour while placing the symbol.

   - Free — Free placement with no attachment

   - Leaders — Multiple attachment points with leader lines

   - On Entity — will place the symbol directly onto a selected 2D item

   - Offset — will place the symbol next to a selected 2D item

9. The **Revision** will be iterated automatically every time you leave the definition dialog with OK and save the configuration. See the **Versioning Objects** on page 69 section in the Appendix for more details-

10. Check the **Obsolete** check box if you want to disable the symbol. It will remain within the configuration, but not anymore available for user selection.

11. For a simple <u>symbol without any user interaction</u> select the **User Independent** section and choose **Add new element** ![plus] ▶ **Add Text**

   • A new **Text** entry is created within the **Object Edit** dialog

▼ ⚠	User independent		+ Entry definition without user interaction
▼ ⚠ A	Text		+ Text definition
⚠ ●	Prompt *		Enter the characters that were used in the symbol, table or pattern file
⚠ ●	Text Entry *	✎	Open 'Text Entry Editor' to create, chose or modify your text entry
●	Translate	☑	Check if this entity shall be translated
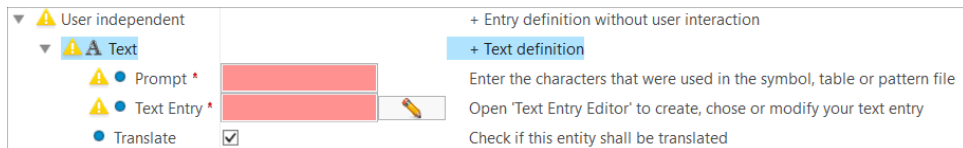
12. Enter the same **Prompt** that was used in the symbol definition.

13. Select ![pencil] to enter the **Translation Tag** representing your final text.

14. (Optional) If you intend to enter an untranslated text, de-select the **Translate** checkbox and enter your text in the **Text Entry** input panel.

15. For a <u>symbol that requires user interaction</u> (i.e. enter a value) you have three options when you select the **User Editable** section and choose **Add new element** ![plus]

   • **Add Value** — the user may enter a value for a variable text
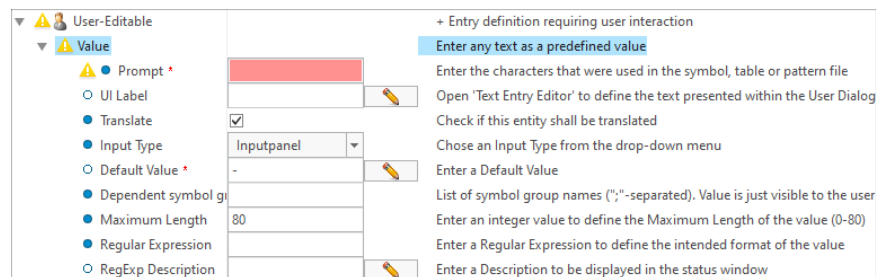
   a. Choose **Add new element** ![plus] ▶ **Add Value**

      ○ A new **Value** entry is created within the **Object Edit** dialog. For every value that you want to be display in the UI you have to create a separate **Value** entry.

▼ ⚠ 👤	User-Editable		+ Entry definition requiring user interaction
▼ ⚠	Value		Enter any text as a predefined value
⚠ ●	Prompt *		Enter the characters that were used in the symbol, table or pattern file
○	UI Label	✎	Open 'Text Entry Editor' to define the text presented within the User Dialog
●	Translate	☑	Check if this entity shall be translated
●	Input Type	Inputpanel ▼	Chose an Input Type from the drop-down menu
○	Default Value *	- ✎	Enter a Default Value
●	Dependent symbol gr		List of symbol group names (";"-separated). Value is just visible to the user
●	Maximum Length	80	Enter an integer value to define the Maximum Length of the value (0-80)
●	Regular Expression		Enter a Regular Expression to define the intended format of the value
○	RegExp Description	✎	Enter a Description to be displayed in the status window

   b. Enter the same **Prompt** that was used in the symbol definition.

   c. Select ![pencil] to enter the **UI Label** that is displayed in the user dialog.

d. Choose the **Input Type** from the pull-down menu to define how the user can edit the parameter value. You can choose from the following options:

   ○ Read-Only — Value is displayed, but user cannot change

   ○ Inputpanel — User can type in any value

   ○ Optionmenu — User can select from pull-down menu only

   ○ Optionmenu (editable) — User can select from pull-down menu or can type in a value

e. Enter a **Default Value** in case of 'Input panel'. Use '-' for no value.

f. If you have selected **Optionmenu** or **Optionmenu (editable)** from the **Input Type** pull-down, a first **Text Item** is created automatically.

   ○ A new **Item** is created within the **Object Edit** dialog
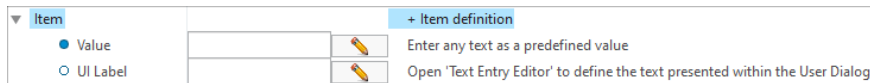
g. Enter a **Value** or select from the translated text dialog

h. (Optional) Select ✎ to enter the **UI Label** that is displayed in the user dialog.

i. Choose additional text items using **Add new element** ➕ ▶ **Add Item**

• **Group Choice** — the user can select a group instance

a. Choose **Add new element** ➕ ▶ **Add Group Choice**

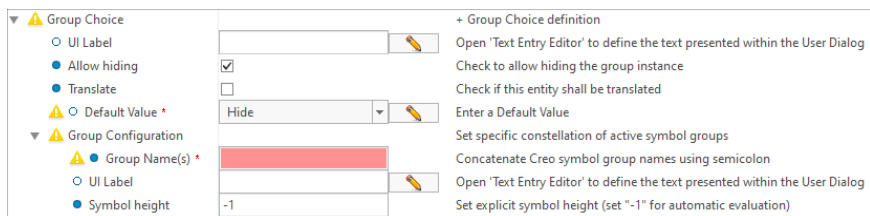   ○ A new **Group Choice** entry is created within the **Object Edit** dialog.

b. Select ✎ to enter the **UI Label** that is displayed in the user dialog.

c. Check the **Allow hiding** check box, if you want to allow this group to be hidden by the user

SMART Annotate

d. Check the **Translate** check box, if you want the variable texts in this group to be translated.

e. Select the **Default** group name that will be displayed in the user dialog.

f. To add multiple groups select the **Group Choice** section and choose **Add new element** ➕ ► **Add Group Configuration**

  ○ A new **Configuration** entry is created within the **Object Edit** dialog

g. Type in a **Group Name** that is available in the selected symbol.

h. Select 🖉 to enter the **UI Label** that is displayed in the user dialog.

i. (Optional) Enter a dedicated **Symbol height** or keep '-1' for automatic height calculation.

j. Add as many **Group Configuration Items** as you like.

• **Group Status** — the user can hide / show a group instance

a. Choose **Add new element** ➕ ► **Add Group Status**

  ○ A new **Group Status** entry is created within the **Object Edit** dialog.

b. Enter a **Group Name** that you want to show or hide

c. Select 🖉 to enter the **UI Label** that is displayed in the user dialog.

d. Select the **Default** status (Hide or Show)..

Tables

The **Table** object contains the definition of a 2D table that can be used in drawing mode only.
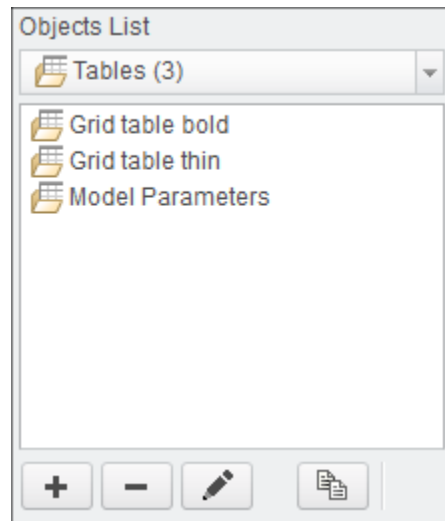
How it works

A **Table** can be graphically selected from the dialog and is manually placed on the drawing . The placement coordinates are defined by user placement. Snapping can be used to initially locate the placement relative to another 2D item.

Adding Tables

1. Select **Add new object** ➕ or select **Edit existing objects** ✎ or use 📄 to copy an existing entry and modify from there.

    •   The **Object Edit** dialog appears.
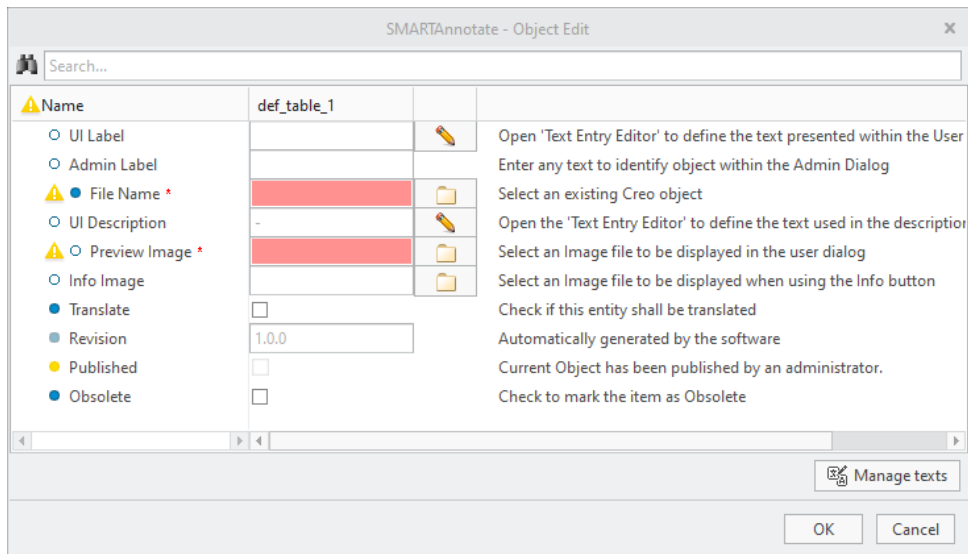
2. Select ✎ to enter the **UI Label** that is displayed in the user dialog.

3. (Optional) Enter an **Admin Label** that is used for distinction within the administration dialog.

4. Select 📁 to choose a **File Name** of a symbol from the `/tbl` folder.

5. Select ✎ to enter the **UI Description** that is displayed in the user dialog.

6. Select 📁 to choose a **Preview Image** from the `/images` folder.This images will be displayed in the user dialog for easy user selection.

7. Select 📁 to choose a **Info Image** from the `/images` folder.This image will pop up when using the **Info** button within the user dialog.

8. Check the **Translate** check box, if you want the variable texts in this table to be translated.

9. The **Revision** will be iterated automatically every time you leave the definition dialog with OK and save the configuration. See the **Versioning Objects** on page 69 section in the Appendix for more details-

10. Check the **Obsolete** check box if you want to disable the table. It will remain within the configuration, but not anymore available for user selection.

6

Working on Content

Search for Content

In configurations with a large number of entries the **Search** function helps you to find and to identify objects that you want to work on..

Search for objects

1. Type in a search string in the **Search** input panel. The filter option ▽
    Object name is active by default.

    *   The list of objects is dynamically filtered representing your search string.
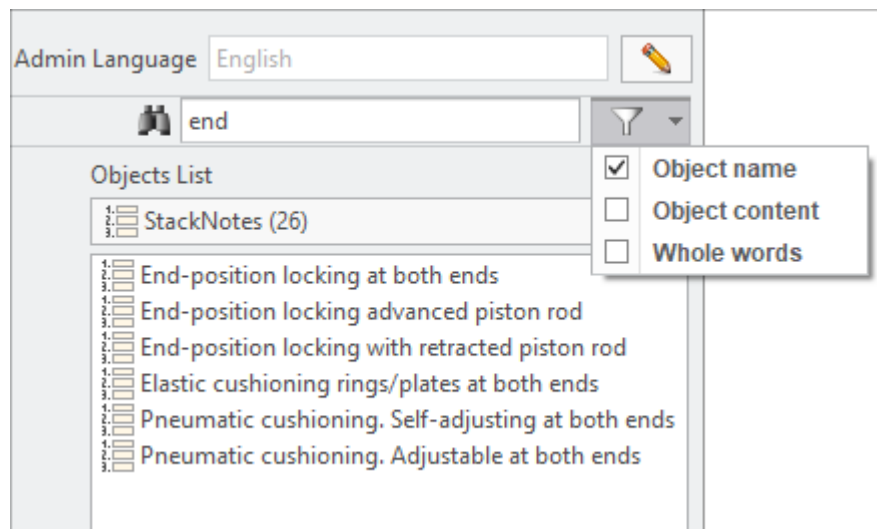
2. Continue typing or select the intended object for editing.

📝 Note

Switching tabs keeps your search active and the list of objects is automatically filtered

---

Search for content within objects

1. Type in your search string and activate the filter options ⧩ **Object name** and **Object content**

   - The list of objects is dynamically filtered representing your search string. It contains results that have a match within the content of the object.
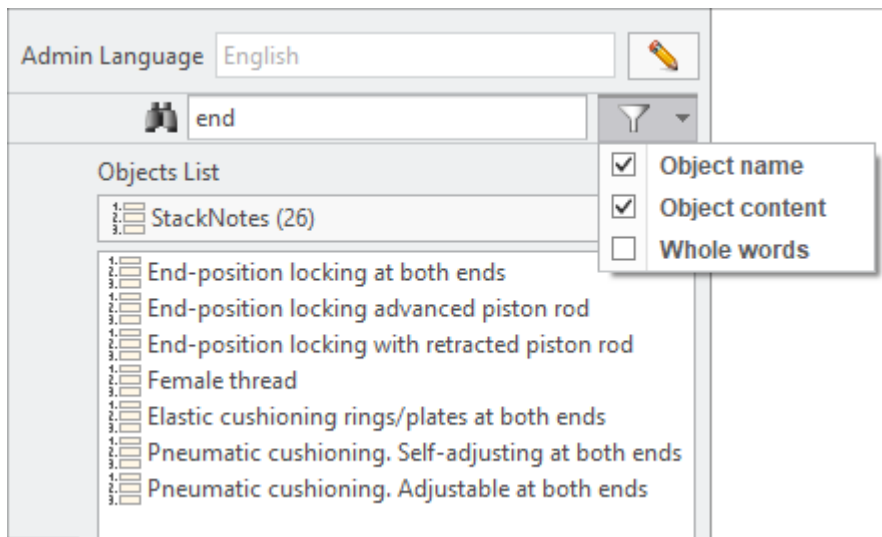
2. Select any of the search results for editing.

- The Object Edit dialog opens
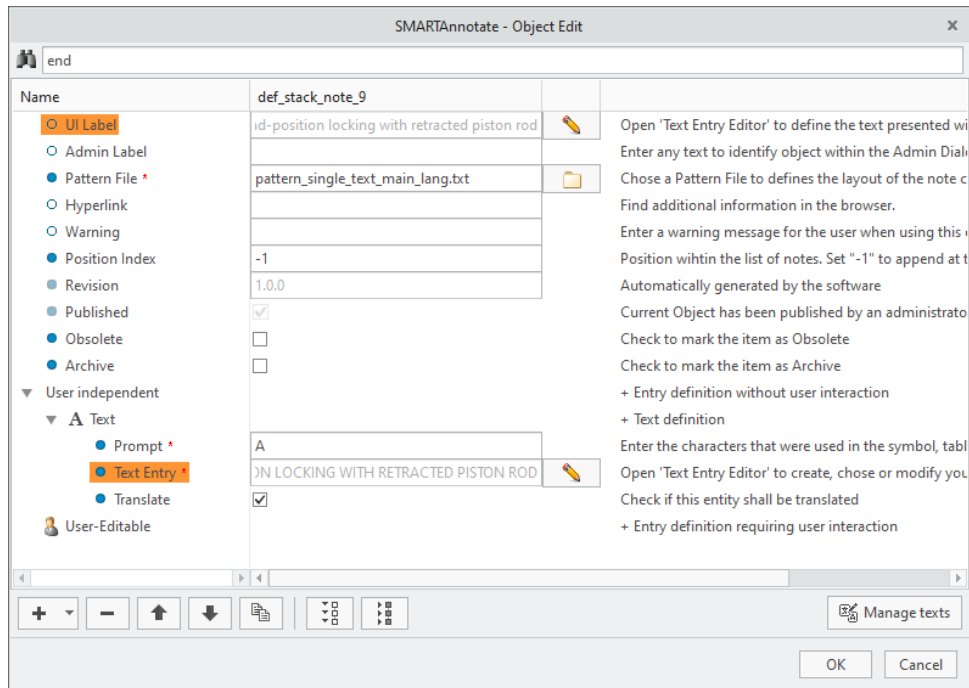
- The search string is taken from the admin dialog and entered in the local search field.
- The locations of the match are expanded in the tree and highlighted

Manage languages

SMARTAnnotate allows you to manage the language that is used as main language as well as adding or deleting complete languages from the tool.
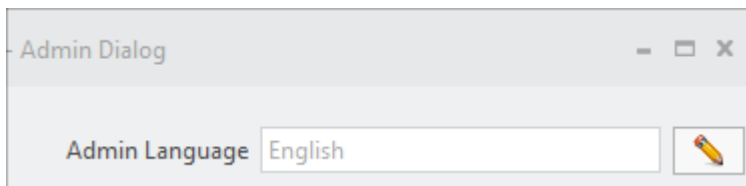
🗩 Note

The display language of the content in the admin dialog can be controlled by the environment variable `SAN_UI_LANG_FOR_CONTENTS`. Refer to the installation section for more details.

Set main language

1. Select Manage language ![pencil icon] from the admin dialog.

   - The **Manage Languages** dialog opens.
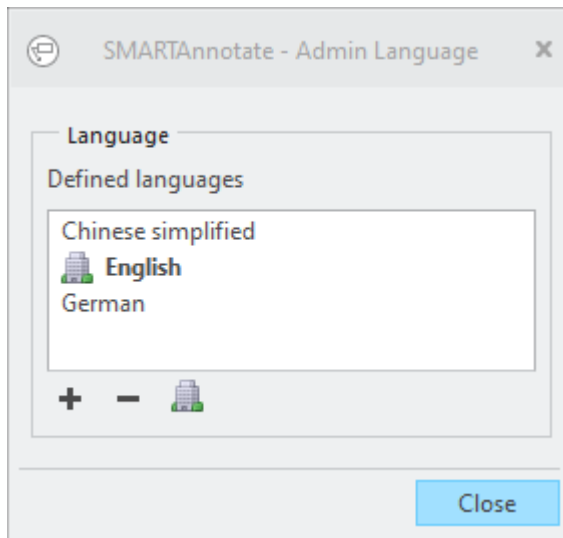
2. Select any language from the list and select the ![icon] icon to set the main language.

3. Confirm that the main language is set to all translation files.

Add a new language

1. Select ![plus icon] from the **Manage Languages** dialog

   - The **Add Language** dialog will open
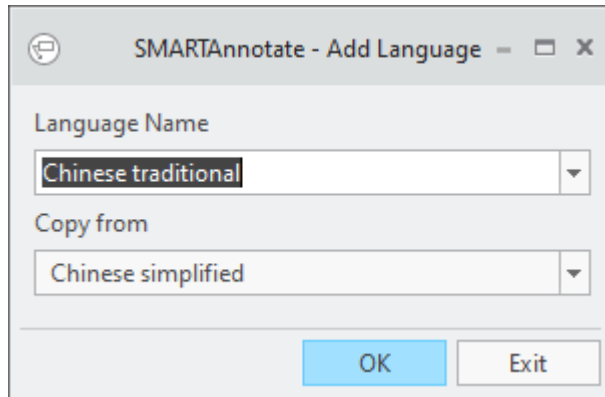
2. Select any of the available languages. The selection is based on Creo language support.

3. Select an existing language from your configuration to start from.

   - This helps to send out files for translation from a specified language.

📝 **Note**

The new texts are not translated automatically.

Delete a language

1. Select any language from the list

2. Select the ▬ to delete the language

   - The complete language folder will be removed from the configuration.

3. Select **SAVE** from Admin dailog.

📝 Note

If this language was not used on any drawing or model, no user will notice the change. If this language was used on drawings or models, the user will be notified once that the used language has been removed by the admin and will disappear.

Revising objects

SMARTAnnotate offers a sophisticated revising functionality that is designed to

- Restore drawing or model content to any time in the past for translation purpose
- Notify the user about changes in texts and / or in the configuration
- Distinguish between published revisions for final usage and temporary revisions for administrative purposes.

Revising texts and object definitions

1. Use the **Object Edit** or **Text Edit** dialog to modify any of the relevant objects on page 72 or texts.

   - Leaving the **Edit** dialog with **OK**, a new **Revision** will be created and indicated as 'unpublished' automatically.

   - You can verify the revision increase and the change to 'unpublished' when you open the object definition again.

● Revision	1.0.3
● Published	☐

   - Modified objects will be also marked as **Unpublished** in the object list.
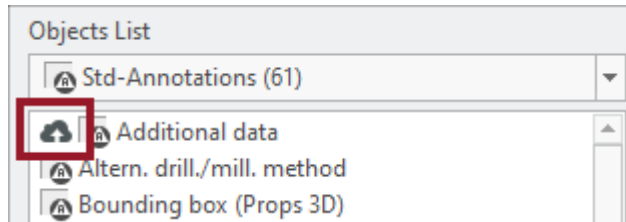
- To discard all unsaved modifications use **Exit** from the Admin dialog.

Note

Revisions of texts will be synchronised between all languages.

Saving modifications

Saving objects or texts stores the data in the respective .xml files.

1. Click **Save** to store your modified objects within Admin mode.

   - All new revisions of modified objects will be stored in the respective .xml files within the configuration, but <u>not</u> marked as published for user mode. That applies to texts and SMARTAnnotate objects definitions.

     o If running SMARTAnnotate Admin mode, you will see the changes and may verify your modifications.

     o In user mode you will not see these changes.

   - You still can delete object definitions from the configuration that never have been published.

   - All text revisions are kept in the configuration and can only be deleted, if they are not used in any revision of a SMARTAnnotate object definition.

Note

Revisions of object definitions and texts will be stored within the drawing or model for future software access.

Publishing object definitions for user visibility

The publishing mechanism is designed to keep your configuration clean and tidy. It is relevant for objects definitions only except for **Parameters** and **Parameter Sets** .
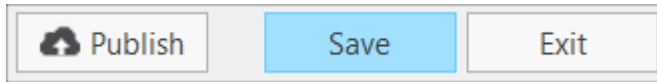
1. Click ☁ **Publish** to mark modified or new object definitions for the usage in User mode

   • The **Publish** dialog opens and shows a list of all unpublished object definitions.
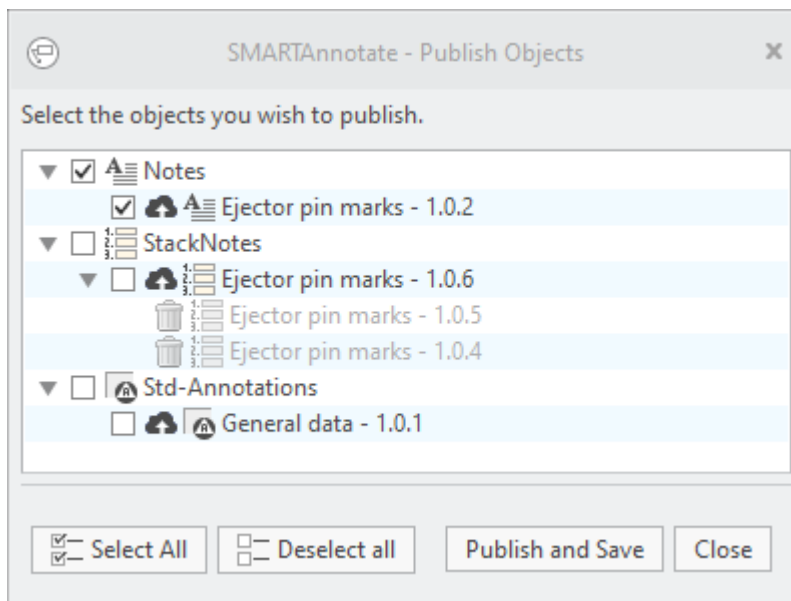
   • Select those definitions that you want to publish.

   • Click **Publish and Save**

      ○ The selected object definitions will be marked as 'published' and saved. They cannot be deleted from the configuration anymore. You can set these definitions to 'Obsolete' instead.

      ○ All 'unpublished' revisions of the selected object definitions, that you may have saved in the meantime, will be deleted from the . xml files.

○ You will see the published revisions of the object definition in user and admin mode.

Relevance for revising and exceptions

In general all objects and its internal entities are relevant for revising. Nevertheless there are some exceptions as these objects have no or irrelevant impact on how a drawing or model may looks like or has looked like in the past.

Excepted objects are:

- **Parameter**
- **Parameter Sets**

Within all object definitions there are some entities that also do not have an impact on the content of the drawing or model. If you modify these attributes, no revision will be created.

Excepted attributes are indicated in the **Object Edit** dialog with an unfilled circle ○ . Examples are:

- **UI label** and **UI description**
- **Admin label**
- **Hyperlink**
- **Position index**

Workflow to manually restore a legacy drawing or model for translation purpose

1. Open the legacy drawing or model
2. Open **SMARTAnnotate** dialog and switch the language

   • The language is switched to the new settings and reflects the respective texts and layouts.

   • In case a legacy object or text revision of an entry is missing the user will get the following information that he may send to the administrator for attention and repair.
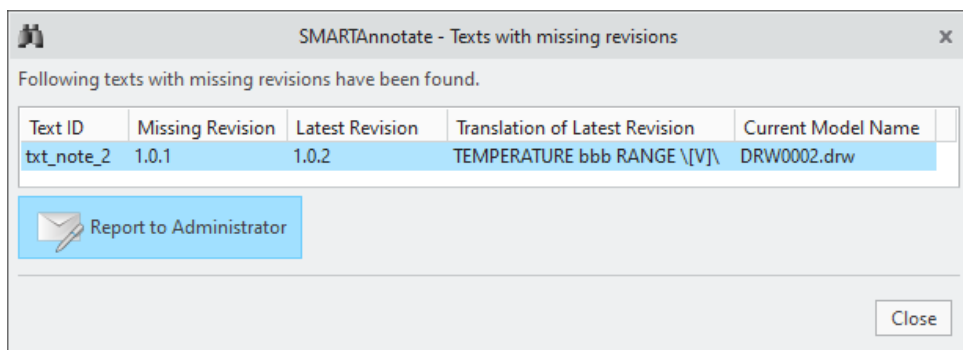
Automatic translation process

1. The worker opens a released legacy drawing or model
2. SMARTAnnotate is opened and the language switched to the intended setting.
3. The worker exports a 2D-PDF file

   • In case a requested revision of a text or object definition a log-file is written to the pre-defined folder location.

   You can use the config options `missing_revisions_output_path` and `missing_revisions_placeholde_text` to tweak the log-file

User notification
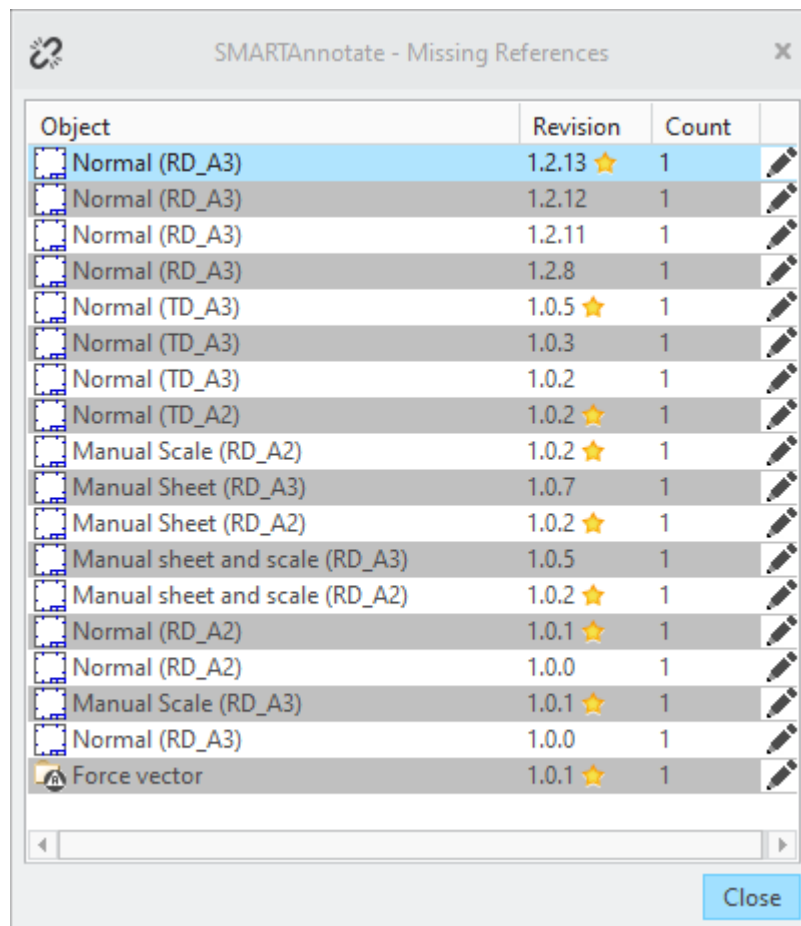
1. Whenever a user opens a drawing or models and activates the **SMARTAnnotate** dialog, he will be notified on changed objects and texts that are used in his Creo drawing or model.
2. He may use the **Notification Center** to update or repair the issues. More details on this you can find in the User guide.

Repair existing content

SMARTAnnotate keeps track of your existing content and provides a list with 'Missing references' when you **Save** the configuration or want to **Publish** object definitions. Reason for missing references are, when you rename or delete supporting files from your configuration, like symbols, images, pattern files or formats.

Worklow

1. Select **Save** or **Publish** from the admin dialog.

   • The **Missing References** dialog opens, if any references within your configuration are broken.

   • All published and the latest revision of an object are listed and need to be repaired.

   • The list contains the following information of the broken object.

      ○ Object name

- o Revision of the object that is affected (the star indicates if this is the latest revision)
- o Number of missing references within this object (hovering over the object will list all missing references)
- o Link to **Edit** the object

2. Select **Edit** ✏️ to directly open the **Object Edit** dialog of the broken object.

- The Object Edit dialog opens and the missing references of this revision are highlighted in red.

- All other attributes are disabled to avoid unwanted modifications to this revision.

3. Select new valid reference(s) and click **OK**

- Your changes are saved automatically.

- **The Missing References** dialog will open again until all references are repaired.

4. Select **Save** to open the list of missing references again and continue with the next missing reference.

Cleaning up the revision history

SMARTAnnotate automatically adds a new revision to every object definition or text entry when it is modified by the administrator. Over time this can create a large number of revisions for every object definition or text entry. To avoid to store all unnecessary revisions in the configuration, a mechanism has been build in to get rid of unnecessary revisions.

This **Clean-up** process is closely connected to the **Publish** mechanism of the object definitions.

Two options for cleaning up are available:

- Cleaning up <u>individual</u> object definitions
- Cleaning up <u>all</u> object definitions and text entries of one object type or all types

Clean up of individual object definitions
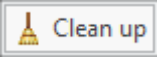
1. Open the intended **Object List** within the administrator dialog.

    *   New and modified objects will be marked
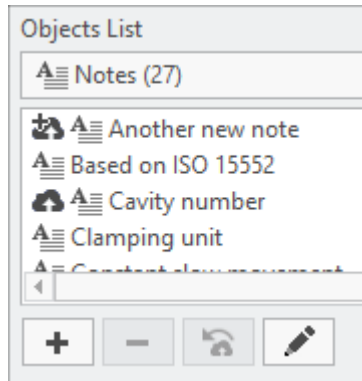
2. Use **Delete** ⊟ to completely remove <u>all</u> revisions of an object that has never been marked as published (indicated by the ♻ icon)

3. Use **Reset** ↺ to delete <u>all unpublished</u> revisions of an object and keep only the published revisions (indicated by the ☁ icon)

Clean up all object definitions and text entries

1. Make sure the general configuration option *ENABLE_REVISIONS_ CLEANUP* is set to *YES*.

2. Use [Clean up] from the Admin dialog to configure the action.

    • The **Clean Up** dialog opens
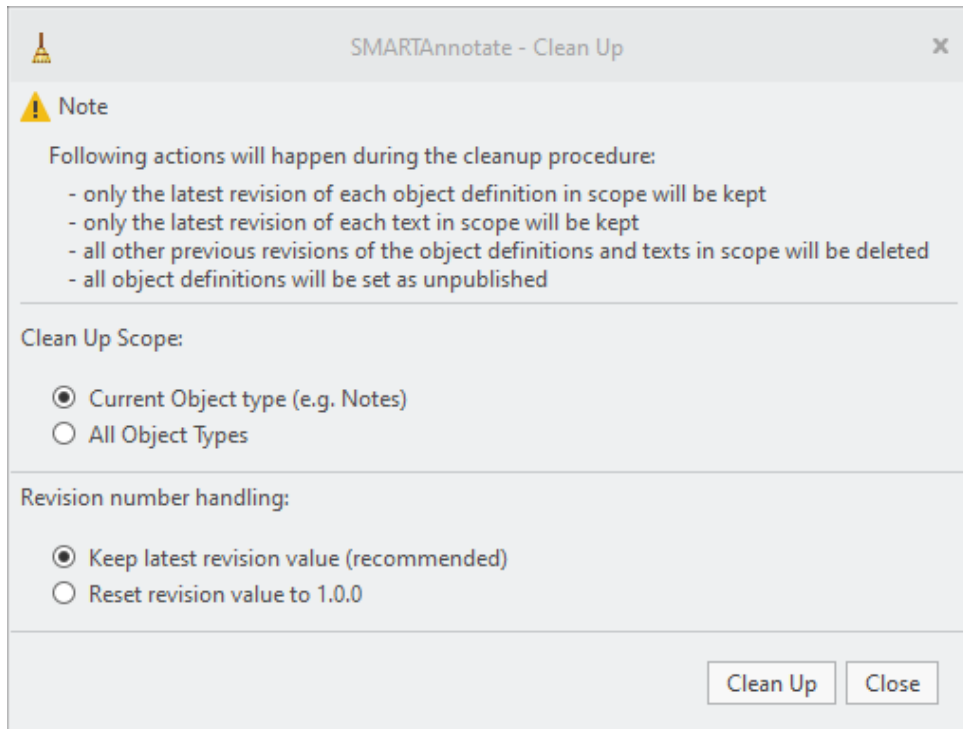
3. Define the scope of the clean up

    • **Current Object type** — Perform the action only for the current active object type and its text entries. (i.e. Notes)

    • **All Object types** — Perform the action for all relevant object types and its text entries.

4. Choose the type of the revision number handling

    • **Keep latest revision value** — the latest published revision will be kept and also its revision number (recommended).

    • **Reset revision value to 1.0.0** — the latest published revision will be kept, but the revision number will be reset to 1.0.0.

> **📝 Note**
>
> Be very careful with this option, as you may have used higher revisions in production

5. Click **Clean Up** to start the process that will do the following:

   - Keep the latest revision from the object definitions and texts in scope.

   - Delete all previous revisions from the object definitions and texts in scope.

   - Either keep the revision number of the remaining revision or reset the to 1.0.0

   - Mark the remaining revision as unpublished

> **📝 Note**
>
> This modification will only take effect when you save the configuration

From here you can continue as follows:

1. **Delete** those object definitions that you permanently want to get rid of.
2. **Publish** those object definitions that you want to use in production

Multi-User Administration

In configurations with a large number of entries there appears the need to work simultaneously with a team of administrators on the same configuration. **The Multi-User Administration** function helps you to lock the files you are working on, so that they are not edited by mistake by other colleagues in the administrator team.

Activate the functionality

1. To activate the functionality you need to set the configuration option `ENABLE_MULTIUSER_ADMINISTRATION` on `YES`:

2. After activating the option you can continue working as usual, while the functionality shows its effects seamlessly while working.

> **Note**
>
> To use this functionality in a team, you need to share the configuration folder on a network drive with your team colleagues.

Parallel working in team on the configuration

1. Assuming you start working on the **Flags** and your colleague e.g. John on the **Formats**, and each of you add a new object to the configuration which creates a change. The following will happen:

   - The XML configuration file of the **Flags** (san_flags.xml) will be locked with your user and computer name from the moment you initiated the change. It will be exclusively available for you to edit until you push the **Save** button, which releases the lock of the XML file.

   - While the XML file of the **Formats** (san_formts.xml) has been locked by your colleague John, you will see this information in the object

list. You will not be able to make any change on the formats until John saves the configuration on his side.
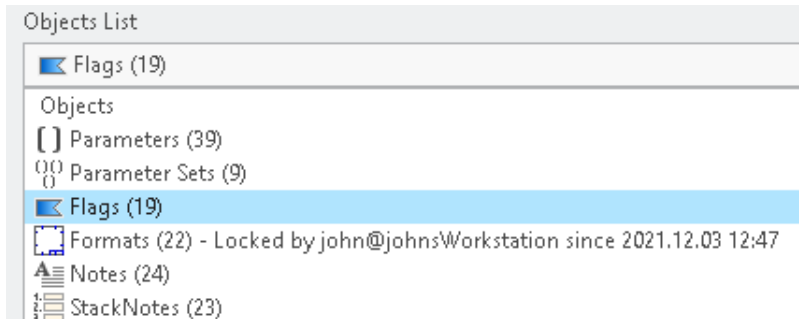
- When John has saved his modifications, you may start making changes to the **Formats** yourself.

2. If your colleague John now starts working on the **Filters**, you will not be able to make any change to the filters until he saves his changes.
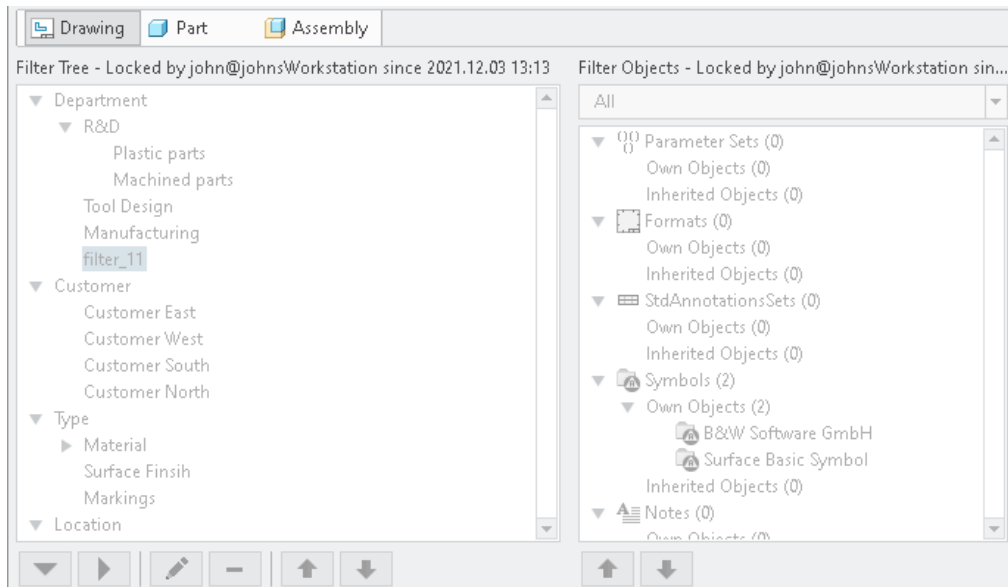
> **Note**

Even if it's possible to start working on the formats right after the colleague John has saved his changes, it still may lead to inconsistencies in rare cases. A good alternative is to designate configuration files responsible persons and distribute the XML configuration files among the members of the administrators team.

7

Content Assignment

Assigning Content to Filters

After setting up the **Filter Structure** and defining **Objects** you finally have to assign your objects to the filter structure.

The status of your assignments is displayed in the **Filter Objects** tree.

For each **Object** type you have two sections:

- **Own Objects** — Objects belonging to the selected filter node.
- **Inherited Objects** — Objects belonging to a parent filter node.
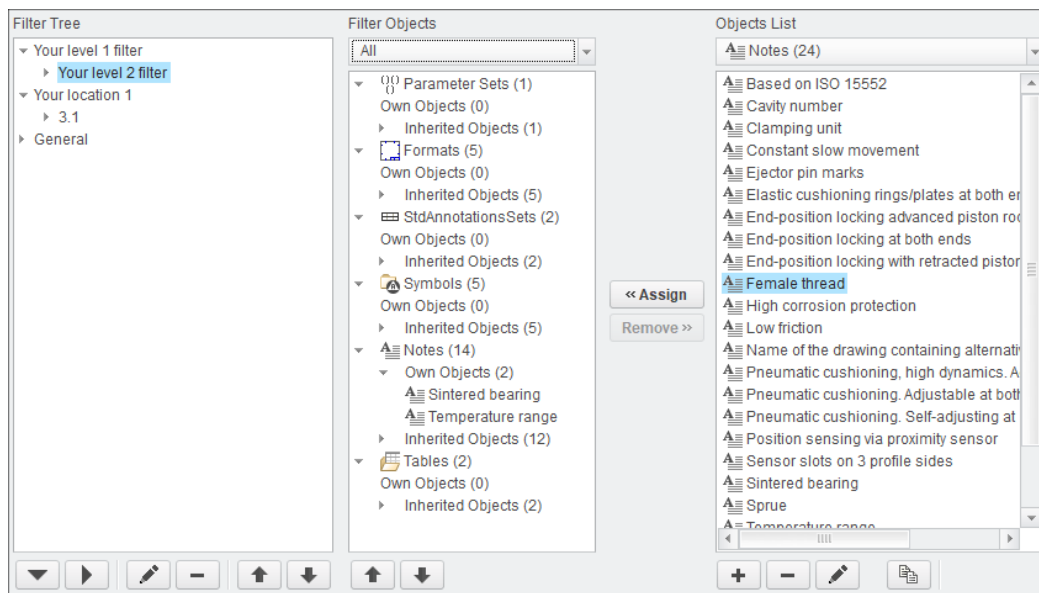
Assign objects

1. Select any node within your filter structure you want to assign objects to.

   - The **Filter Objects** tree will update and display the content of the selected filter node.

2. Browse to the intended object in the **Objects List**.
3. Click the **Assign** button.

- The selected object will be assigned to the **Own Objects** section in the **Filter Objects** tree.

📝 **Note**

An **Object** can only be assigned once to a filter node.

---

4. Use 🔼 or 🔽 to reorder the entries.

Remove objects

1. Browse to the intended object in the Filter Objects tree and click the **Remove** button.

   - The object is removed from the **Own Objects** section.

8

Using SMARTAnnotate

The User Dialog

The User dialog within **SMARTAnnotate** allows you to easily select objects and place them on the drawing or within a 3D model.

1. Select **Annotate** ⊕ from the **SMARTAnnotate** tab
   - The **User** dialog opens
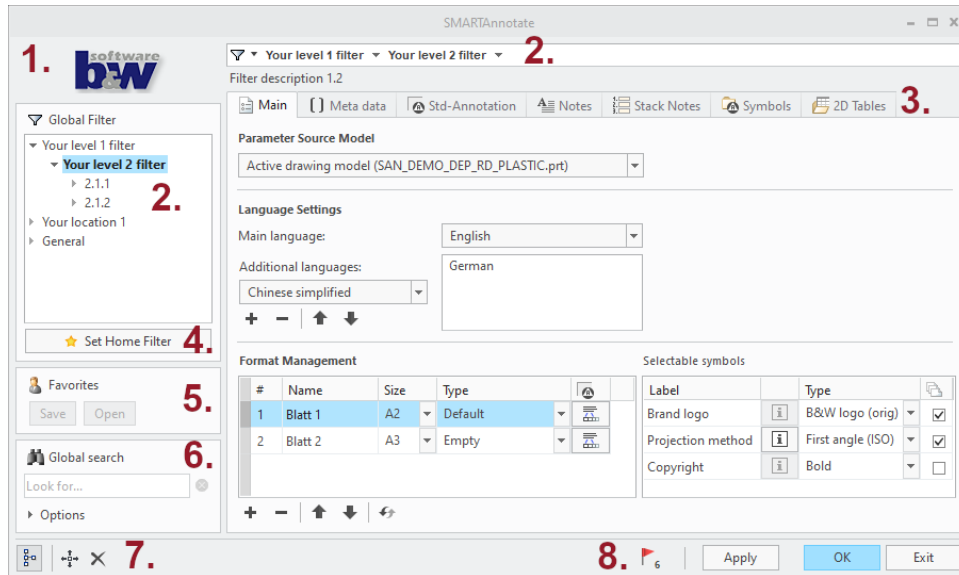
1. Company logo
2. Global filter tree
3. Content tabs
4. Home-Filter control
5. User Favorites
6. Global search
7. Control bar
8. Notification center

Working with the Filter Structure

The filter structure is the main mechanism to distinguish different settings like notes, formats, symbols for user selection and display.

You can browse through the structure using the **Global Filter** tree:
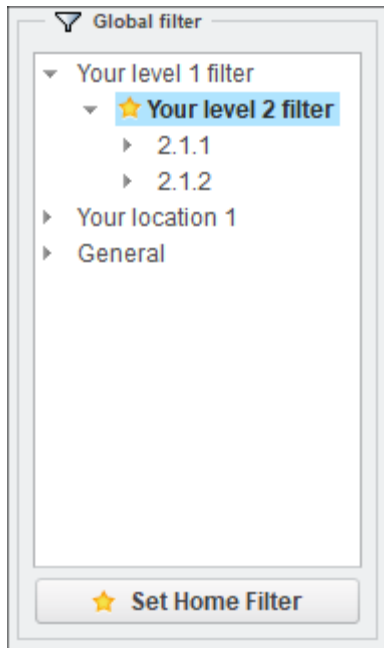
The selected filter node is highlighted and indicates that you are now looking at content related to this node.

The **Home Filter** ⭐ is related to the **Global Filter Parameter** and its value defined in the administration dialog. When opening the user dialog the software looks for this parameter and its value in the related 3D model. If the parameter exists and the value matches with a filter definition the structure is expanded up to this node and the yellow star is placed next to it.

You may use the **Set Home Filter** button to apply the home filter to any node. This will write the respective parameter value back to the active Creo model.

User Favorites

The **Favorites** capability provides the ability to save and retrieve user related settings manually. You may save multiple selections including its values.
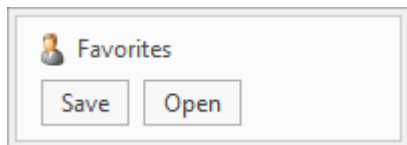
It is available for the following areas:

- Std-Annotations
- Stack-Notes
- Symbols

Saving Favorites

1. Activate the **Std-Annotations** tab or the **Stack-Notes** tab
2. Select **Save** from the **Favorites** layout

    - In case of Std-Annotations you will be prompted to select the intended annotation sets.

3. Enter a name and save the .xml file to any folder location

    - The .xml file is saved and contains the current selection of your Std-Annotations or Stack Notes.

Retrieving Favorites

1. Activate the **Std-Annotations** tab or the **Stack-Notes** tab
2. Select **Open** from the **Favorites** layout
3. Browse to the previouly saved .xml file and open the file

    - In case of **Std-Annotations** you will be inforemd about the status of the import and prompted to select the sheet(s) where the annotations shall go to.

    - In case of **Stack-Notes** you will be informed if any entities are already in your selection list and not imported. Place the stack as usual.

Additionally, the software saves the latest selection and the latest used languages by default.

It is available for the following areas:

- Std-Annotations
- Stack-Notes
- Symbols
- Languages in Main tab

Latest used settings

1. When leaving the **User dialog** with **OK**, latest used settings are saved to your local machine automatically.

    - A structure of .xml files is created in `C:\Users\<name>\AppData\Roaming\SmartAnnotate`

    - Recently used **Languages**, **Std-Annotations, Symbols** and **Stack-Notes** are stored

2. Retrieve these selections with the ⟳ icon from the Std-Annotation or Stack-Notes tab.

3. For new objects the latest used languages are resumed from the settings.

Language and Format Management

When opening **SMARTAnnotate** the **Main** ▤ tab is displayed.
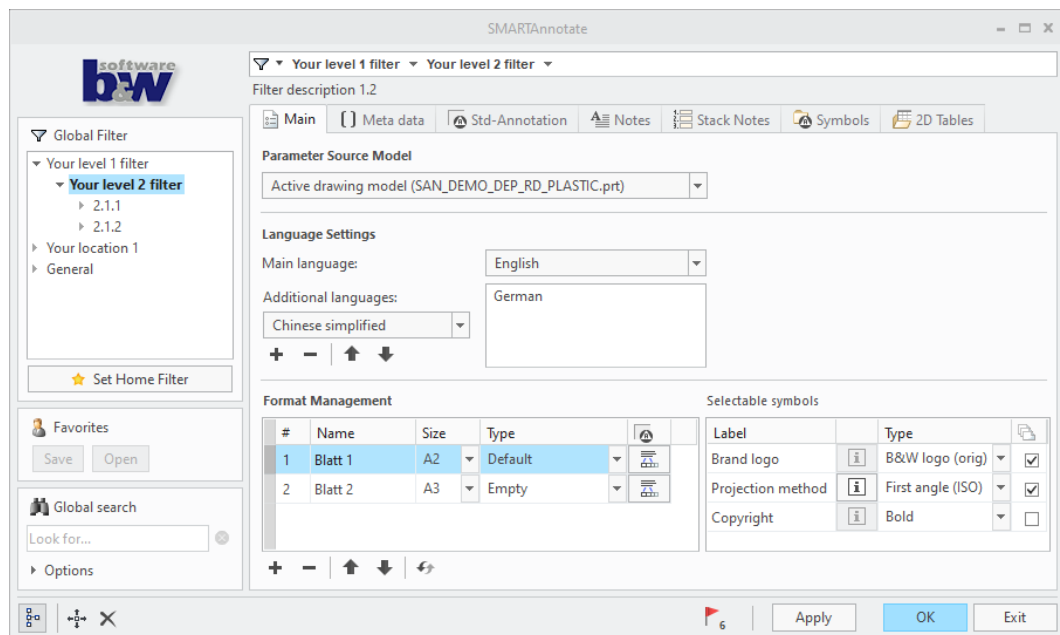
Parameter Source Model

The displayed model name indicates where parameter values are called out from.

Parameter Source Model
Active drawing model
Active drawing model
DEMO_PART.prt
DEMO_PLATE.prt
DEMO_PART_METAL.prt
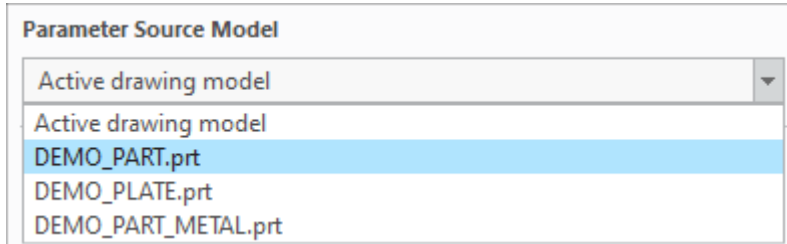
- In part or assembly mode the current active solid model name is displayed.
- In drawing mode you may have multiple drawing models and you can choose from the two options:
  - **Active Solid Model** (default) — the parameter values will switch when you change the active drawing model and update the format tables.
  - Any selected drawing model — The parameter values are permanently called out from the selected model.

Entry first row first column	Entry first row second column
Entry second row first column	Entry second row second column

Language Settings

To use the language settings the following requirements need to be valid:

- You must have a valid license for the multi-language option
- The config option to enable multi-language must be set
- Other languages need to be configured

Switch the **Main language** to another language by selecting another entry from the pull-down menu. The language will be exchanged in all your texts, symbols, formats as far it is configured for language change.

Add or remove **Additional languages** from the pull-down menu. This will add a 'Translation' to your existing text using the main language.

Note

This needs to be configured within your symbols, pattern files and formats to take effect.

Note

For non-roman language types you may need to install the necessary font.

Format Management

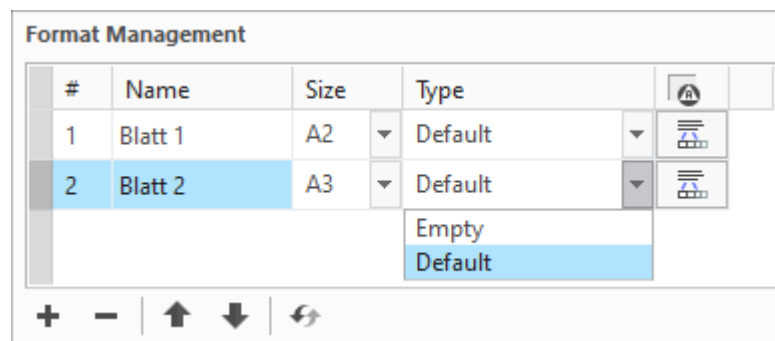Available in **Drawing mode** only!

- Select a paper **Size** from the pull-down menu.
- Select a format **Type** from the pull-down menu. **Empty** removes the format from your drawing or indicates that no format is used. This option is not available for **Custom** format sizes.

- Use the **Sheet Placement** ⬚ icon to define on which sheet the software will apply the Std-Annoations on.

  ○ The **Placement** dialog opens.
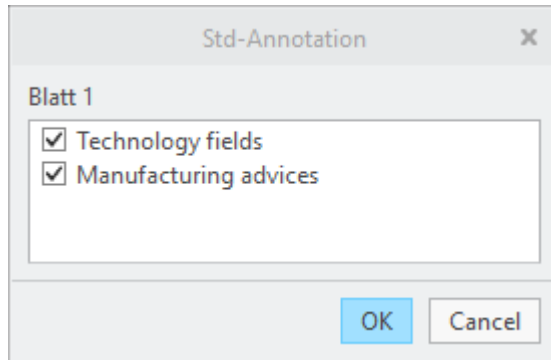
  ○ Select or deselect the Annotation-Sets from the list.
- You can **Add** ➕ or **Remove** ➖ Sheets.
- Sort the Sheets using ⬆ and ⬇ .
- The **Update** 🔄 button updates all information related to the formats.

Selectable Format Symbols

In order to provide a maximum on flexibility your administrator has enabled the ability to choose from a symbol instance in the **Selectable Symbols** table.

- Select a symbol instance from the drop-down list
- Use the adjacent check box to specify if this symbol instance shall be applied to

  ○ All drawing sheets (box is checked)

  ○ This particular sheet (box is unchecked)

- If an image of the symbol is provided form the administration, you may click on the **Info** button and see a preview image of the respective symbol instance.
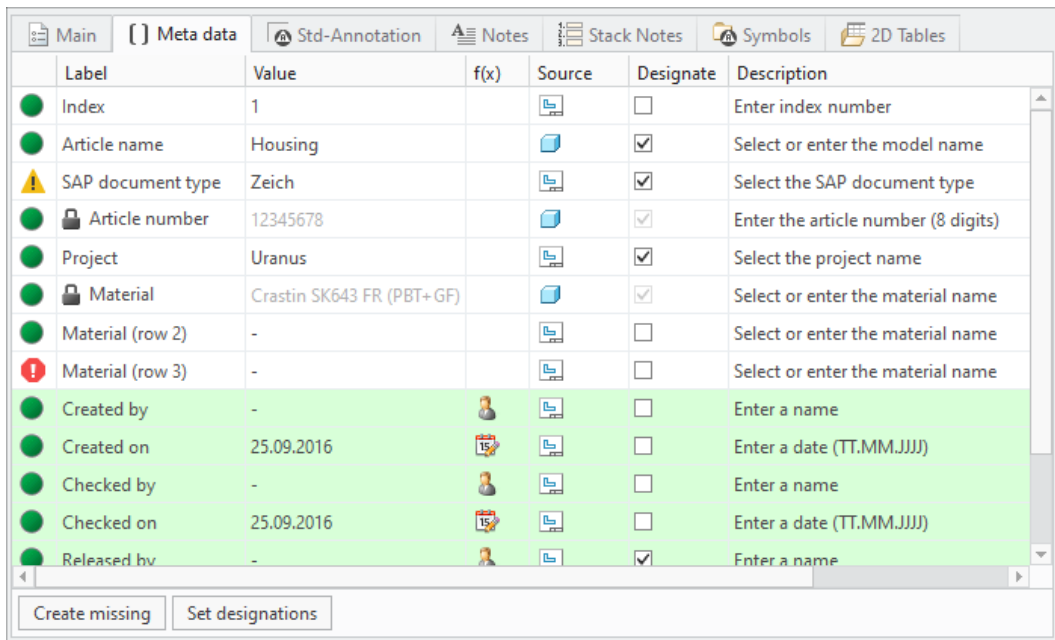
> **📋 Note**
>
> When adding a new sheet to your drawing the settings of the previous symbols will be evaluated to set the symbol instance. If you select a different format type the default from the configuration is used.

Using the Meta Data Management

The **Meta data** [ ] tab displays all **Parameter Set** definitions that are associated with the current filter node or the drawing format. The sets may have different background colors.

The complete tab can be hidden via a configuration setting if you do not want to use this functionality.

- The **Label** column indicates the parameter label. It may not be equal to the parameter name.
- Use the **Value** column to enter a value or chose from a pull-down list
- The function column **f(x)** provides special functions to fill the value.
  - 👤 — Inserts your current Login-Name
  - 📅 — Inserts Today's date

- The **Source** column indicates the source of the parameter value..

    ○ 🖼 — Drawing

    ○ 🔷 — Solid model

- The **Description** column provides a little extra information for filling the value.

- Use the **Create missing** button to create all missing parameters automatically.

- Use the **Set designations** button to correct all wrong designation states.

> 🗨 **Note**
>
> When leaving the dialog with **OK**, all missing parameters are created and all designations are set.

- The **Status** column provides a quality status. Click on the bullet and a status window will open.
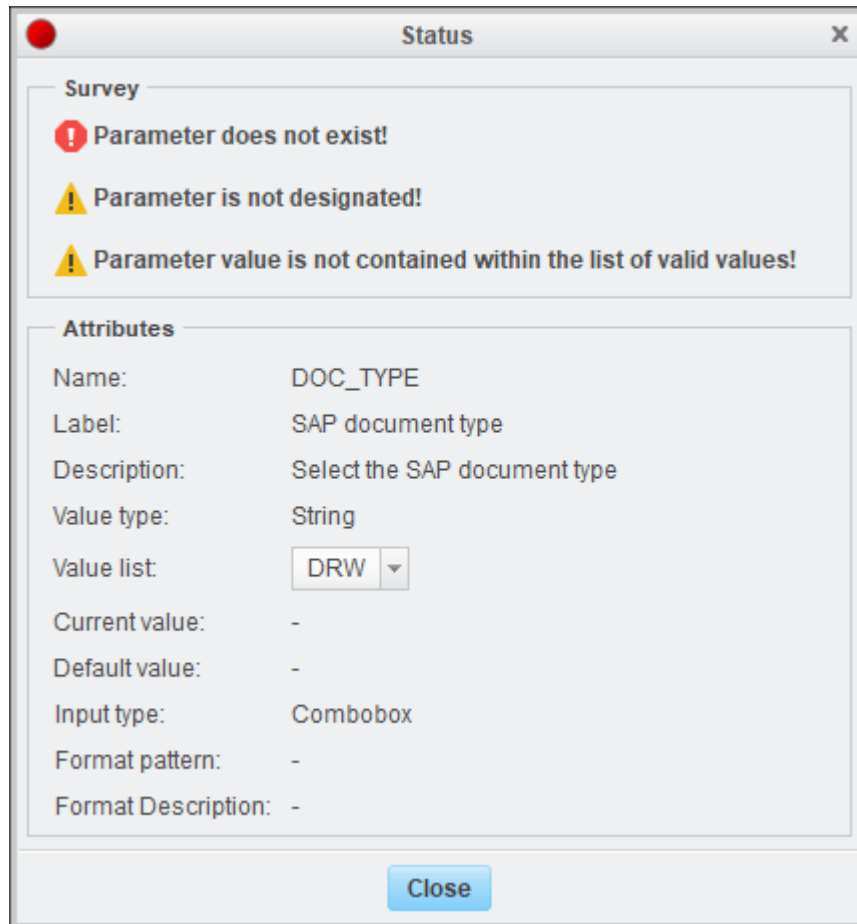
It lists the current issues and the attribute definition. It will help to correct the issues quickly.

- Use the **Apply** button or the **OK** button of the dialog to see the values appear in the call out areas.

Applying Std-Annotations

The **Std-Annotation** ⓐ tab provides a pull-down menu for all **Std-Annotation Set** definitions that are associated with the current filter node or the drawing format. The content of the current set is displayed.

The complete tab can be hidden via a configuration setting if you do not want to use this functionality.
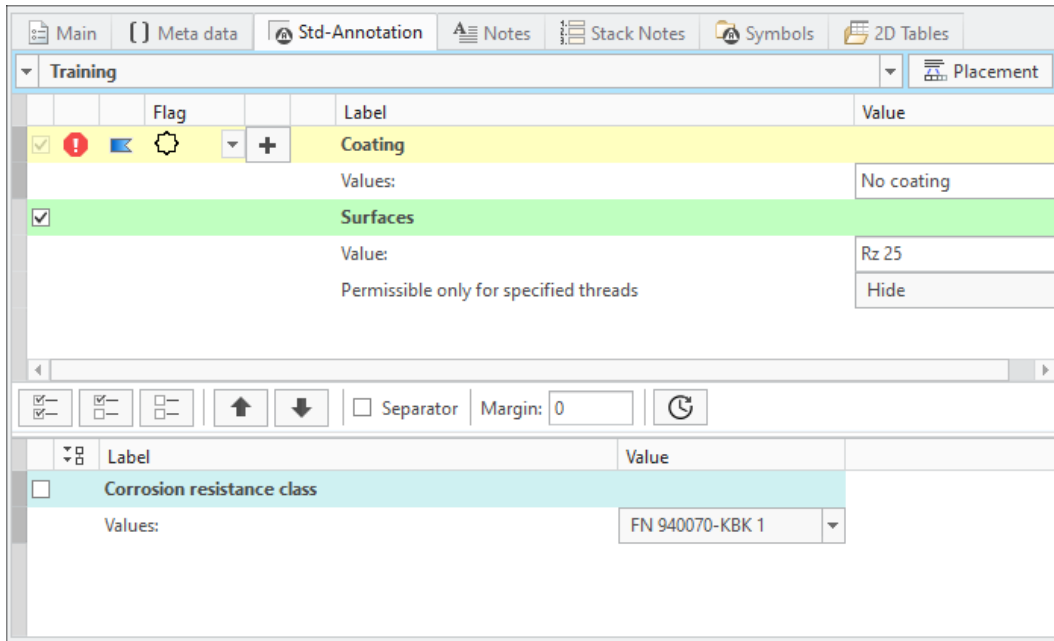
- The display is split into two areas.
  - **Pool area** (Lower display) — Available entries associated with the filter node are listed here
    - You may expand ⌖ or collapse ⌖ the value rows and column to improve the visibility
  - **Collection area** (Upper display) — All selected items and its values are displayed here
- The **Label** column displays the Std-Annotation label itself and the label of the corresponding values.
- The **Value** column provides the ability to enter values or choose from pull-down lists.
- The **Flag** column indicates if a flag is assigned to the annotation definition.
- The **Flag Placement** column provides the **Plus** ✚ button to place one or multiple flags for the selected row. It appears when the flag is set to active only.
- The **Flag Status** column shows a Warning symbol, if no flags have been placed for an active flag symbol.
- The **Flag symbol** column allows to choose the flag symbol.

Selection

1. Chose the **Std-Annotation Set** from the pull-down at the very top of the display.

2. Select entries from the **Pool area** one-by-one, all ▣ or mandatory and required items ▣ to add them to the preview area.

   - The selected Std-Annotations are displayed in the **Collection area**.

   - You can sort the entries with the ⬆⬇ buttons.

   - To separate the entries you may use a **Separator** or a **Margin**.

   - One entry may contain multiple rows. A headline and several value or choice rows. The complete entry is marked with a dark bar when selected.

   ### 💬 Note

   'Mandatory' items cannot be deselected from the **Collection area**. Only the administrator can change this setting.

(Optional) Flag Activation and Placement

1. Click on the little flag symbol to activate the **Flag** indication

   - ⚑ — Flag inactive

   - ⚑ — Flag active

     ○ A new entry will be created automatically within the **Flag** ⚑ tab and the **Stack Notes** sub-tab. Start flag symbol placement from there.

   - ⚑ — Flag indication (option `LISTING_FLAG_ENABLES_MIXED_ STATE = YES`)

     ○ No entry will be created automatically within the **Flag** ⚑ tab and the **Stack Notes** sub-tab.

2. Choose the **Flag symbol** from the pull-down list.

3. Use the **Plus** ➕ in the respective row to start placing the flag.

- The **SMARTAnnotate** dialog will disappear and the **Settings** tab in the Ribbon will open.

Attachment Type	With Leaders ▾	Leader Type	On Surface ▾	✓	↻	⇌ˣ	✕
		Arrowhead	⚡ Automatic ▾				
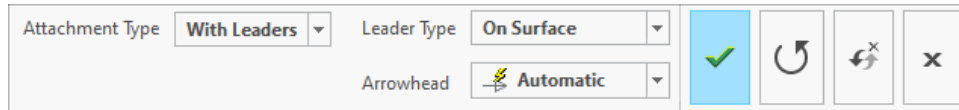
4. Choose the **Attachment Type** from the pull-down menu.

   - Free — Unattached placement
   - At vertex — Select a vertex
   - Normal to entity — Placement normal to selected geometry
   - With Leaders — Select one or multiple attachment points
   - On Entity — Select a geometry item that the note will placed on
   - Offset — Select a 2D item and chose an offset from there.

💬 **Note**

The available options are defined during symbol definition and might vary.

---

5. Choose the **Leader Type** from the pull-down menu
6. Choose the **Arrowhead** style from the pull-down menu
7. Select the attachment references (if required) and place with LMB.

   - Select **OK** from the little popup menu or use MMB to finish the placement.

8. Finish the placement

   - Select **Redo** ⟳ from the dashboard UI to redo the current placement.
   - Select **OK** ✓ from the dashboard UI or MMB to finalize the placement.
     - The **SMARTAnnotate** dialog will re-open automatically and you can continue with another flag placement.
   - Select **Cancel** ✕ from the dashboard UI to discard the placement.
     - The **SMARTAnnotate** dialog will re-open automatically and you can continue with another flag placement.

Placement

1. Select the **Placement** ⊞ button to define or verify the drawing sheets where the Annotation set goes to.

   - The Placement dialog opens.
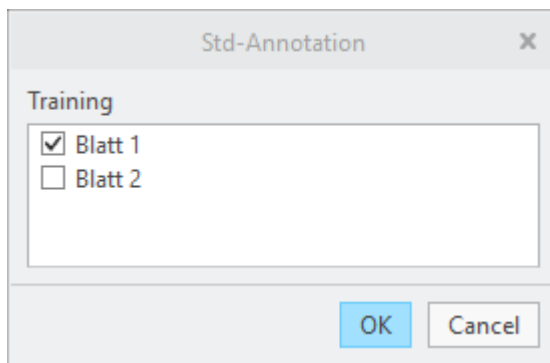
   - Select or deselect from the list of Sheets.
   - Close the dialog with **OK**.

2. Select the **Apply** button or the **OK** button of the dialog.

   - For every **Std-Annotation Set** a stack of **Std-Annotations** is build up automatically.

> **Note**

Do not forget to set the check box for Std-Annotations in the **Main tab ▶ Format Management** in the intended sheet!

Placing Notes

The **Notes** ▲≣ tab provides a selection UI for all **Note** definitions that are associated with the current filter node.

The complete tab can be hidden via a configuration setting if you do not want to use this functionality.
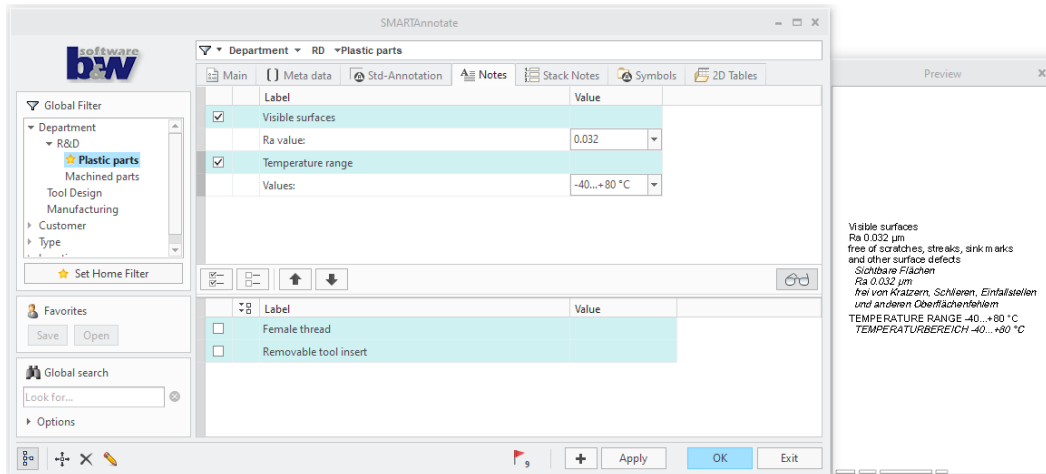
- The display is split into three areas.

  ○ **Pool area** (Lower display) — Available entries associated with the filter node are listed here

    ◆ You may expand ⊞ or collapse ⊟ the value rows and column to improve the visibility

  ○ **Collection area** (Upper display) — All selected items and its values are displayed here

  ○ **Preview area** (Attached to the right) — Live preview of your Note configuration including all values and choices (2D mode only)

- The **Label** column indicates the Notes label. It may not be equal with the containing text. When you move the mouse over the text label, a tooltip with the complete text will appear.

- The **Value** column is for entering values or choosing from pull-down lists.

Selection

1. Select entries from the **Pool area** one by one or all ⊟ .

   - The selected notes are displayed in the **Collection area**.

   - You can sort the entries with the ⬆ ⬇ buttons.

- (Optional) Open the **Preview area** and view the live status of your note configuration (2D mode only)
- One entry may contain multiple rows. A headline and several value or choice rows. The complete entry is marked with a dark bar when selected.

2. (Optional) Enter a value or chose from a pull-down list if necessary.

- The Preview is updated according to your selection and choices. Use the return-key to finish an entry.

Placement

1. Use the **Plus** to start placing the notes.

- The **SMARTAnnotate** dialog will disappear and the **Settings** tab in the Ribbon will open.
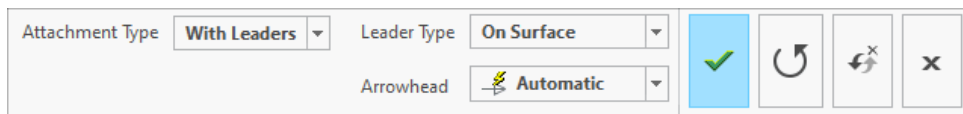
2. Chose the **Attachment Type** from the pull-down menu.

- Free — Unattached placement
- With Leaders — Select one or multiple attachment points
- On Entity — Select a geometry item that the note will placed on
- Offset — Select a 2D item and chose an offset from there.

3. Chose the **Leader Type** from the pull-down menu
4. Chose the **Arrowhead** style from the pull-down menu
5. Finish the placement

- Select **OK** or use MMB to finish the placement.
  - The **SMARTAnnotate** dialog will re-open automatically and you can continue with another set of notes.
- Select **Cancel** to cancel the placement.
  - The **SMARTAnnotate** dialog will re-open automatically and you can continue with another set of notes.
- Select **Redo** to start over with the placement.
- Select **Repeat** to keep the current instance of the note and add another one

---
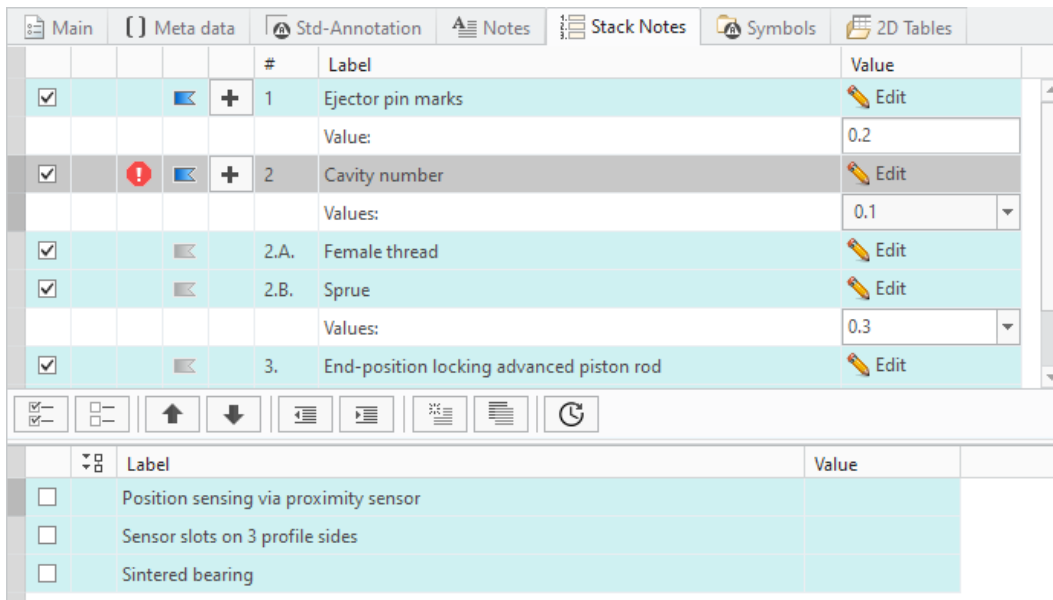
> 📋 **Note**
>
> If you have selected multiple note items, they are merged into a single note during placement.

Placing Stack Notes

The **Stack Notes** ⁞☰ tab provides a selection UI for all **Stack Note** definitions that are associated with the current filter node.

The complete tab can be hidden via a configuration setting if you do not want to use this functionality.

- The display is split into two areas.

  - **Pool area** (Lower display) — Available entries associated with the filter node are listed here

    - You may expand ⊟ or collapse ⊞ the value rows and column to improve the visibility

SMART Annotate

- ○ **Collection area** (Upper display) — All selected items and its values are displayed here
- The **Label** column indicates the Stack Notes label. It may not be equal with the containing text. When you move the mouse over the text label, a tooltip with the complete text will appear.
- The **Value** column is for entering values or choosing from pull-down lists.
- The **# (Number)** column applies an automatic numbering scheme. You can deactivate numbering using the configuration option `NOTE_LISTING_TYPE`. There are several more options to control the formatting of **Stack Notes.** Please see the Configuration Chapter for more details.
- The **Flag** column indicates whether an entry is marked as flagged. Find more information here on page 104.
- The **Flag Placement** column provides the **Plus** button to place one or multiple flags for the selected row. It only appears when the flag is set to active.
- The **Flag Status** column shows a warning notification symbol, if no flags have been placed for an active flag symbol.

Selection

1. Select entries from the **Pool area** one by one or all .

   - The selected notes are displayed in the **Collection area**.
   - You can sort the entries with the buttons.
   - One entry may contain multiple rows. A headline and several value or choice rows. The complete entry is marked with a dark bar when selected.

2. (Optional) Enter a value or chose from a pull-down list if necessary.

Placement

1. Use the **Plus** to start placing the notes.

- The **SMARTAnnotate** dialog will disappear and the notes stack will stick to your cursor for placement.

2. Click LMB to drop the stack

3. Move your mouse over the notes stack.

   - Watch more and more notes appear as you hover down the stack.

   - All notes will appear immediately if you move the mouse over the top of the stack.

4. Use LMB again to drop those notes that are currently displayed.

   - The stack will break at this point and allows you to continue with the next column the same way until all notes are dropped.

   - A horizontal snap line will help aligning the columns will placing the notes.

5. The placement finishes automatically when all notes are dropped.

   - The **SMARTAnnotate** dialog will re-open automatically.

   - The selected notes will remain in the **Collection area**.

📋 **Note**

You can only place <u>one</u> stack of notes. The notes will remain separate notes after placement.

(Optional) Flag Activation and Placement

1. Click on the little flag symbol to activate the **Flag** indication

   - ◤ — Flag inactive

```
1. END-POSITION LOCKING WITH ADVANCED PISTON
   ROD
   ENDLAGENVERRIEGELUNG BEI AUSGEFAHRENER
   KOLBENSTANGE
2. PNEUMATIC CUSHIONING. ADJUSTABLE AT BOTH
   ENDS
   PNEUMATISCHE DÄMPFUNG. BEIDSEITIG
   EINSTELLBAR
3. PNEUMATIC CUSHIONING. SELF-ADJUSTING AT
   BOTH ENDS
   PNEUMATISCHE DÄMPFUNG. BEIDSEITIG
   SELBSTEINSTELLEND
```

- — Flag active

  - The note number will be marked as flagged within the note text in the model/drawing.

```
1  END-POSITION LOCKING WITH ADVANCED PISTON ROD
   ENDLAGENVERRIEGELUNG BEI AUSGEFAHRENER
   KOLBENSTANGE
2. PNEUMATIC CUSHIONING. ADJUSTABLE AT BOTH
   ENDS
   PNEUMATISCHE DÄMPFUNG. BEIDSEITIG
   EINSTELLBAR
3  PNEUMATIC CUSHIONING. SELF-ADJUSTING AT BOTH
   ENDS
   PNEUMATISCHE DÄMPFUNG. BEIDSEITIG
   SELBSTEINSTELLEND
```

- — Flag indication (`option LISTING_FLAG_ENABLES_MIXED_STATE = YES`)

  - Same as active state, except that no warning notification will be shown if there was no flag note placed yet within the model/drawing.

2. Use the **Plus** in the respective row to start placing the flag.

   - The **SMARTAnnotate** dialog will disappear and the **Settings** dashboard within the ribbon is shown.
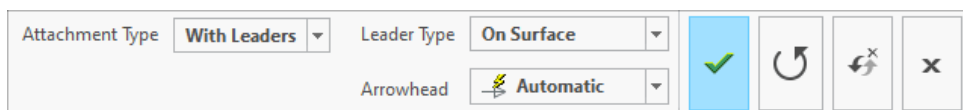
3. Chose the **Attachment Type** from the pull-down menu.

   - Free — Unattached placement
   - At vertex — Select a vertex

- Normal to entity — Placement normal to selected geometry
- With Leaders — Select one or multiple attachment points
- On Entity — Select a geometry item that the note will placed on
- Offset — Select a 2D item and chose an offset from there.

4. Chose the **Leader Type** from the pull-down menu
5. Chose the **Arrowhead** style from the pull-down menu
6. Select the attachment references (if required) and place with LMB.

    - Select **OK** from the little popup menu or use MMB to finish the placement.

7. Finish the placement

    - Select **Redo** from the dashboard UI to redo the current placement.

    - Select **Repeat** from the dashboard to keep the current instance and add another one.

    - Select **OK** from the dashboard UI or MMB to finalize the placement.

        ○ The **SMARTAnnotate** dialog will re-open automatically and you can continue with another flag placement.

    - Select **Cancel** from the dashboard UI to discard the placement.

        ○ The **SMARTAnnotate** dialog will re-open automatically and you can continue with another flag placement.

The created flags may look like this:
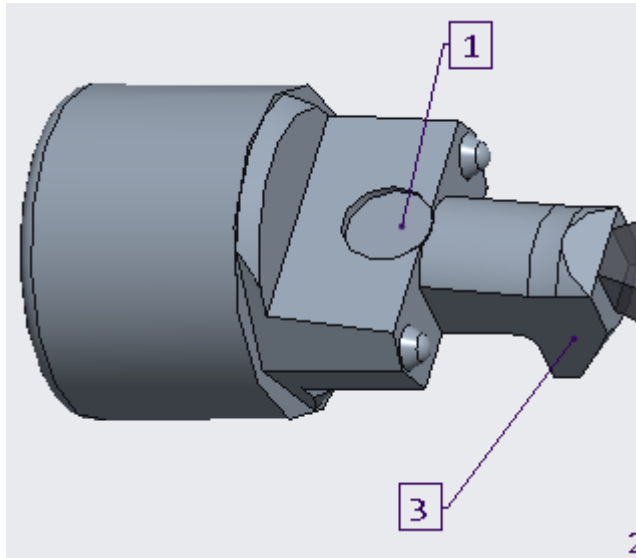
In 3D mode, the user-placed flags will be stored as elements within an annotation feature. The following parameters are created for these elements:

- **FLAG_NOTE**: Value is the name of the annotation element that corresponds to the note within the stack with the same number which the flag is referencing.

- **BUW_FLAG_TEXT**: Value is the actual number for instance "1" or "3".

- **BUW_PREFIX**: This parameter gets called-out within the flag note text. If its value is non-empty, the value text is shown in front of the flag text.

- **BUW_SUFFIX**: This parameter gets called-out within the flag note text. If its value is non-empty, the value text is shown behind the flag text.

The prefix and suffix text can be changed using the edit function . After a flag note was selected, the following dialog will appear:

If the configuration option ENABLE_FLAG_NOTE_PROTECTION is set to NO, then the flag notes can also be edited interactively. On the one hand, the annotation element parameters can be changed by opening the

corresponding Creo parameters dialog. On the other hand, the note text can be selected and by double-clicking on the parameter call-outs, Creo shows an input panel that allows to edit the parameter values as well.

'Custom Notes' for individual user modifications

Custom Notes are a possibility for the user to add individual notes to the stack. This adds a lot of flexibility and individualism to this kind of note type, but also has its limitations.

- **Custom Notes** do not support automatic translation
- Converting an existing **Stack Note** entry into a **Custom Note** is a point of no return
- No support for drop-down menus or Input fields.

Custom notes can be created in single or multi language manner in respect to the corresponding configuration setting. In single language mode, there's just one text area for text input while in multi language mode, there's a text area for every language that was selected. For instance if three languages are chosen, the dialog will appear like this:
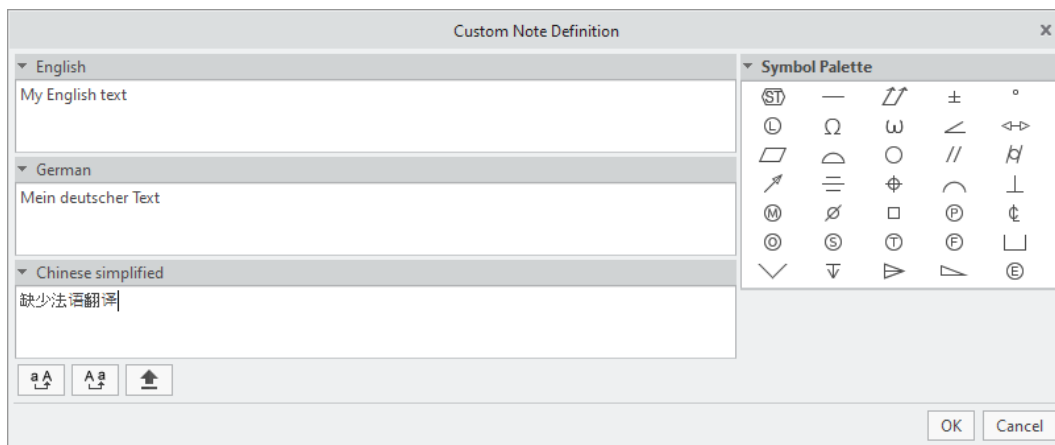
The following illustrations will show the custom note dialog used in single language mode.

There are different options to create a **Custom Note** that you can turn on or off individually through the configuration.
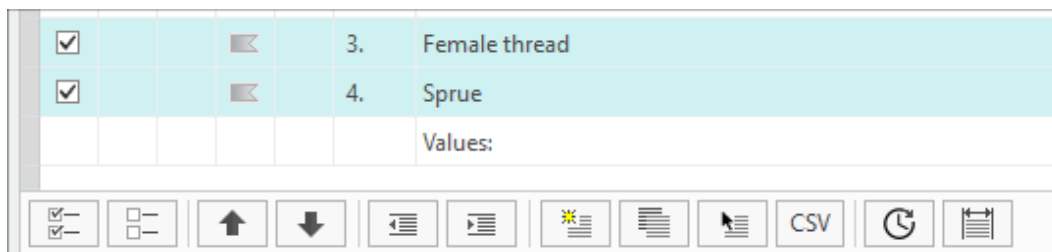
1. Create from scratch

   a. Select ⚎ from the icon bar within the **Stack Notes** tab

      • The Custom Note Definition dialog opens
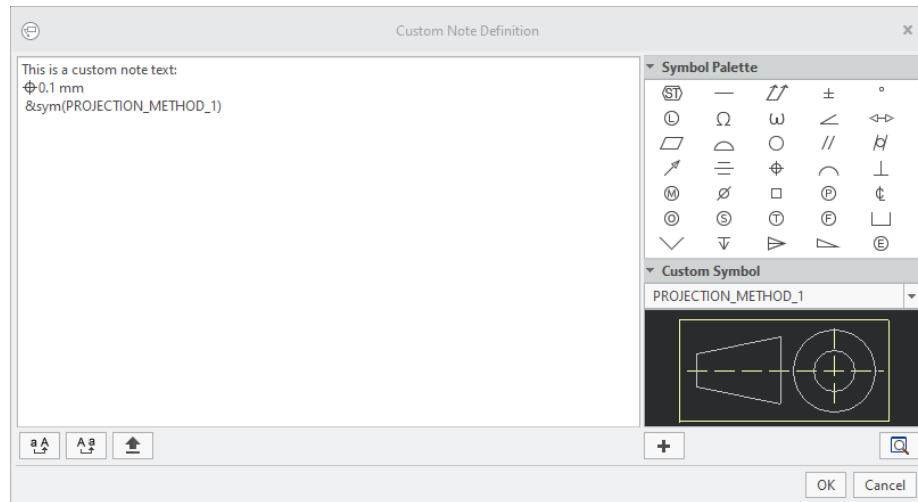
   b. Enter any text into the large text area. You will have to enter your own translations here, if intended. Use the following buttons to control your input.

      • ⬚ — Converts all selected characters to **Upper Case**

      • ⬚ — Converts all selected characters to **Lower Case**

      • ⬚ — Caps lock key

   c. (Optional) Select a symbol from the symbol palette.

   d. (Optional) Select a custom symbol to be called out within the text from the drop-down list.

   > 📝 **Note**
   >
   > The symbols appearing in the list are the symbols that are already included in the drawing or model

e. Click **OK** and the new **Custom Note** ![icon] will be added to notes stack.
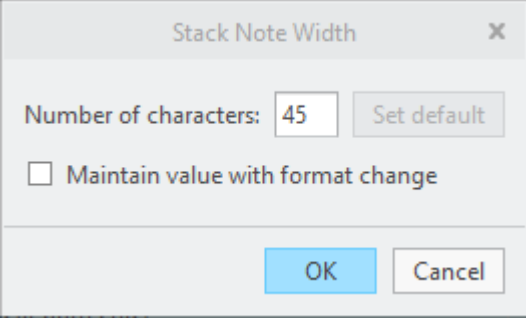
☑		◄	3.	Female thread		✎ Edit
☑		◄	4	Sprue		✎ Edit
				Values:	0.3	▾
☑	▤	◄	5.	New custom note		✎ Edit

2. Create by copying an existing note entry

   a. Select ![icon] from the icon bar within the **Stack Notes** tab

      • The **Custom Note Definition** dialog opens

Custom Note Definition	✕
Choose Source: End-position locking advanced piston rod ▾	
END-POSITION LOCKING WITH ADVANCED PISTON ROD ENDLAGENVERRIEGELUNG BEI AUSGEFAHRENER KOLBENSTANGE	Symbol Palette ⓢ — ⫽ ± ° Ⓛ Ω ω ∠ ↔

   b. Choose an existing notes entry from the drop-down list. The original text appears within the text area.

   c. Modify the existing text as you like

   d. Click **OK** and the new **Custom Note** ![icon] will be added to notes stack.

3. Create by selecting an existing text from the drawing or model

   a. Use option *CUSTOM_NOTES_CHOP_RULE* to define a regular expression that will cause the selected note to be split up into separate notes at each occurrence of specific characters.

   b. Select ![icon] from the icon bar within the **Stack Notes** tab

      • The Custom Note Definition dialog opens

   c. Modify the existing text as you like

   d. Click **OK** and the new **Custom Note** ![icon] will be added to notes stack.

4. Create by importing from a CSV file

   a. Use option *CUSTOM_NOTES_CHOP_RULE_CSV* to define a regular expression that will split up the text from CSV file into separate notes at each occurrence of specific characters.

   b. Select **CSV** from the icon bar within the **Stack Notes** tab

- The **File Open** dialog opens

c. Select the intended CSV file from your folder structure

- The content gets loaded into the **Custom Note Definition** dialog

- According to the definition within your Configuration, the content is split into separate entities that you can walk through with the little arrows at the bottom.

d. Modify the existing text of each entity as you like

e. Click **OK** and the new **Custom Notes** will be added to notes stack.

5. Use the **Recent** button to load the custom notes from your local 'recent notes' stack.

6. Define the character width of the stack notes column.

a. Use the **Width** button to define the width of the stack

- A dialog opens to enter an integer value for the maximum characters in a row of the stack.

- Check the box to maintain the width even in the event of a format change.

7. Edit an existing note entry

a. Select **Edit** from within the existing line item in the list

- You will be prompted if you really want to convert this entity into a **Custom Note**.

- The **Custom Note Definition** dialog opens showing the original text.

b. Modify the existing text as you like

c. Click **OK** and the existing note will turn into a **Custom Note** .

💬 Note

There are multiple Configuration options that control the display and formatting of the **Custom Notes**.

By default all empty lines will be removed automatically. Please contact your administrator for changing the respective option.

Placing Symbols

The **Symbols** 📁 tab provides a graphical selection UI for all **Symbol** definitions that are associated with the current filter node.

The complete tab can be hidden via a configuration setting if you do not want to use this functionality.
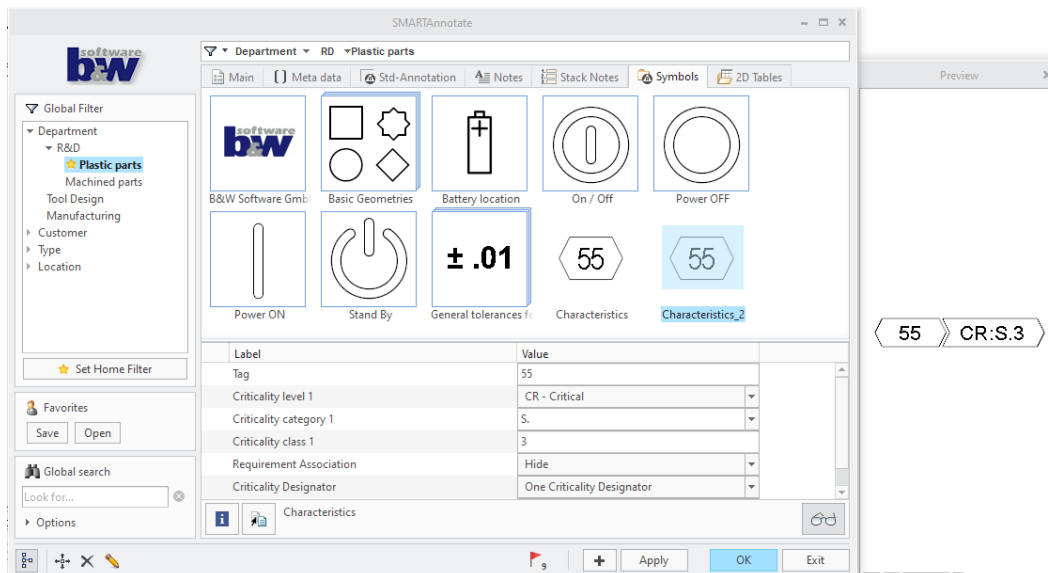
- The display is split into four areas.

  ○ **Graphical Selection area** — Available symbols associated with the filter node

  ○ **Value area** — Value and Group choices

  ○ **Description area** — Link to a large image and description of the symbol
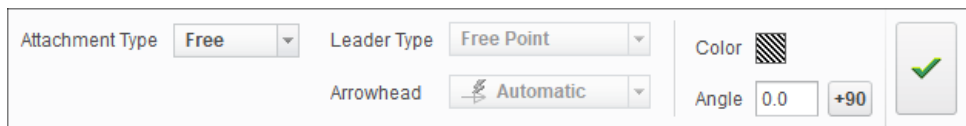
○ **Preview area** — (attached to the right) Live preview of your symbol instance including the chosen groups and values

Selection and configuration

1. Select one entity from the **Graphical Selection area** .

   - A description is displayed in the **Description area**.
   - Use the **Info** icon to open up a larger preview or description image.
   - Open the **Preview area** 👓 to receive a live preview of your symbol instance configuration.

2. (Optional) Enter a value or chose a group from a pull-down list if necessary. Use the Return-key to finish your input.

   - The live preview will be updated accordingly

Placement

1. Use the **Plus** ➕ to start placing the symbol.

   - The **SMARTAnnotate** dialog will disappear and the **Settings** tab in the Ribbon will open.

| Attachment Type | Free | ▼ | Leader Type | Free Point | ▼ | Color | ▨ | ✓ |
| | | | Arrowhead | 🖋 Automatic | ▼ | Angle | 0.0 | +90 |

2. Chose the **Attachment Type** from the pull-down menu.

   - Free — Unattached placement
   - At Vertex— Select a geometry vertex that the symbol will be placed next to.
   - Offset — Select a 2D item and chose an offset from there.

   💬 **Note**

   The available options are defined during symbol definition and might vary.

3. (Optional) Chose the **Leader Type** from the pull-down menu

4. (Optional) Chose the **Arrowhead** style from the pull-down menu

5. (Optional) Chose a **Color** for the symbol instance.

6. (Optional) Rotate the symbol instance by an **Angle** or in **+90** degree steps

7. Select the attachment references (if required) and place with LMB.

8. Finish the placement

    - Select **Confirm** ✔ from the little popup menu or use MMB to finish the placement.

        o You are in a loop and can continue to place more instances.

    - Select **Discard** ✖ from the little popup menu to redo the current placement.

    - Select **OK** ✔ from the ribbon UI or use RMB to finalize the placement.

        o The **SMARTAnnotate** dialog will re-open automatically and you can continue with another symbol placement.

Reusing a configured symbol instance

There are two options to reuse your final configured symbol

- Saving as a **Favorite**

    1. When your symbol definition is ready and placed in your model or drawing, use the **Save** button from the **Favorites** 👤 area

    2. Pick the symbol instance from your model or drawing

        o You will be prompted to save the XML file to your local machine

    3. To retrieve the saved instance, use **Open** from the **Favorites** 👤 area

        o Select the intended XML file

        o The saved symbol instance with all settings is brought into the dialog

        o Modify any values ore groups to your intend and place the instance.

- **Copy** function

    1. Use **Copy symbol** 📋 from the dialog

    2. Select any symbol that has been placed by SMARTAnnotate

        o You will be automatically put into placement mode and can place the new symbol instance

○ Once placed, you will be brought back to the symbol dialog

📝 **Note**

Both options will work in 2D and 3D mode of **SMARTAnnotate**

Placing Tables

The **Table** 📊 tab provides a graphical selection UI for all **Table** definitions that are associated with the current filter node.

The complete tab can be hidden via a configuration setting if you do not want to use this functionality.

- The display is split into three areas.
  - ○ **Graphical Selection area** — Available symbols associated with the filter node
  - ○ **Description area** — Link to a large image and description of the symbol
  - ○ **Preview area** — Live preview of your filled table

Selection

1. Select one entity from the **Graphical Selection area** .

- A description is displayed in the **Description area**.
- Use the **Info** icon to open up a larger preview or description image.
- (Optional) Open the **Preview area** to display the selected table including all variable texts.

Placement

1. Use the **Plus** ⊞ to start placing the symbol.

    - The **SMARTAnnotate** dialog will disappear and the **Settings** tab in the Ribbon will open.
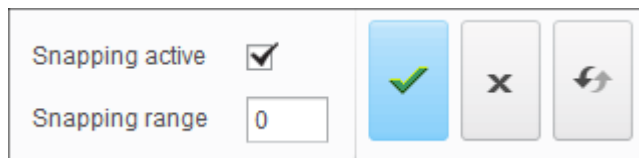
2. Chose the **Attachment Type** from the pull-down menu.

    - Free — Unattached placement
    - At Vertex— Select a geometry vertex that the symbol will be placed next to.
    - Offset — Select a 2D item and chose an offset from there.

> 🗩 **Note**
>
> The available options are defined during symbol definition and might vary.

3. (Optional) Uncheck the **Snapping Active** check box to disable table snapping.
4. (Optional) Enter a **Snapping range**.
5. Place the table with LMB.

    - Use RMB to switch to the next snapping origin while dragging the table.

6. Finish the placement

    - Select **OK** ☑ or use MMB to finish the placement.

o The **SMARTAnnotate** dialog will re-open automatically and you can continue with another table.

- Select **Cancel** ![X] to cancel the placement.

  o The **SMARTAnnotate** dialog will re-open automatically and you can continue with another table.

- Select **Redo** ![icon] to start over with the placement.

Manipulating existing Objects

By default **SMARTAnnotate** does not allow you to select objects that previously have been created via the software.

To manipulate these objects you have to use the Edit functions from the **SMARTAnnotate** ribbon or from within the dialog.
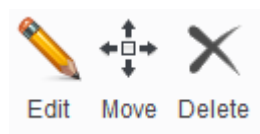
Edit objects

1. Click the **Edit** ![icon] icon and select an object that was created via **SMARTAnnotate**.

   - The **SMARTAnnotate** dialog will open to allow any kind of editing.

2. Select **Apply** or **OK** to see the changes in the graphics area..

Move Objects

1. Click the **Move** icon and select an object that was created via **SMARTAnnotate**.

   - The object is attached to your cursor for dragging.

   - In case of **Stack Notes** you will get prompted if you want to move only the selected column or the entire stack. You will have to re-place the column or entire stack.

2. Drop the object at its new location with LMB.

> 📝 **Note**
>
> This is not available for **Formats** and **Std-Annotations**.

Delete objects

1. Click the **Delete** icon and select any object.

    - The object will be deleted.
    - There is not multiple selection for this command for safety reasons.

9

Notifications

Using the Notification Center

The Notification center within **SMARTAnnotate** is designed to point out Errors and Warnings that the software detects while working in the dialog. It will help correcting the issues as far as possible.

Displaying Warning and Errors

1. Select **Open Notification Center** ⚑ from the bottom panel of the user dialog.
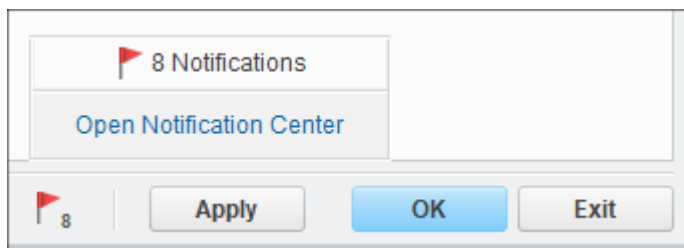
- The **Notification Center** dialog opens
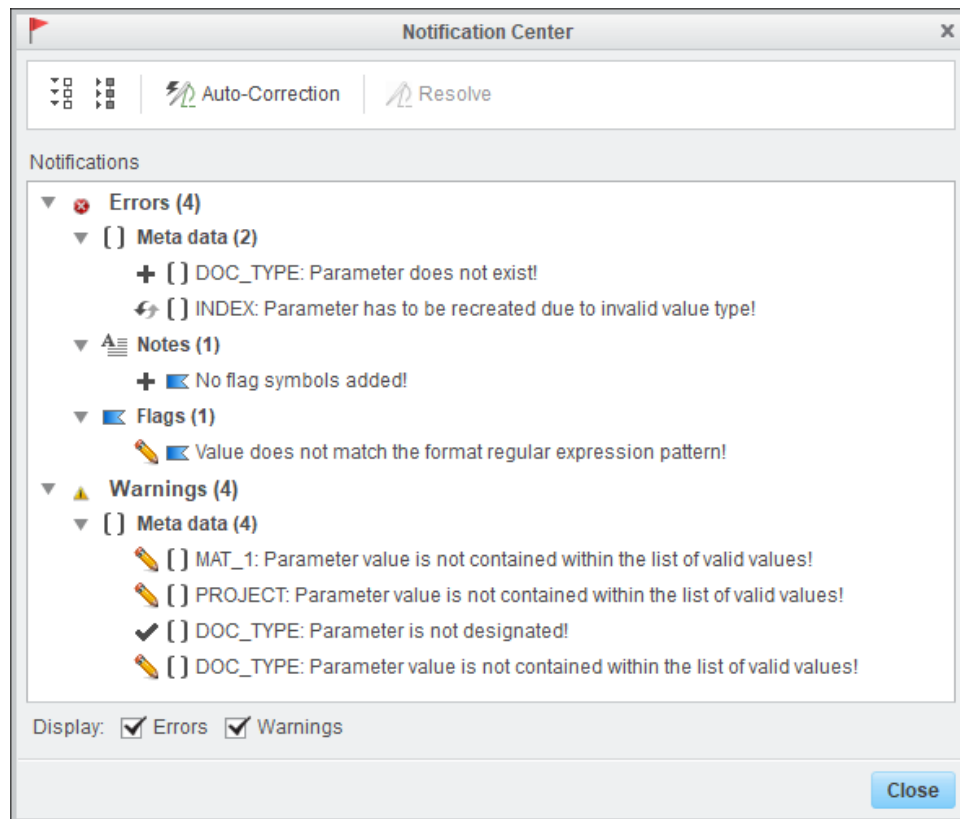
> 📝 **Note**
>
> The dialog is not available if there are no warnings or errors.

Resolving Errors

1. Select a single Warning or an Error from the **Notifications** list.

2. Select the **Resolve** icon.

   - The software will take you directly to the place where you can solve the issue manually (i.e. bring you into the tab to correct a value that does not match, starts placing a flag that was not added) or even resolves the issue automatically (i.e. creating a missing parameter automatically).

3. (Optional) Select the **Auto-Correction** icon.

   - The software will try to resolve the issues automatically. The corrected issues will disappear from the list. Work on the remaining issues manually.

Ease of use

1. Tooltip for every entry in the message list

2. Cross reference highlighting when selecting a note type message in the dialog

3. Listing numbers of stack notes are included in message lines

10

Configuration

The Configuration Dialog

The Configuration dialog within **SMARTAnnotate** allows the administrator to customize and prepare the software for production use.

1. Select **Configuration** ⚒ from the **SMARTAnnotate** tab

    *   The **Configuration** dialog opens
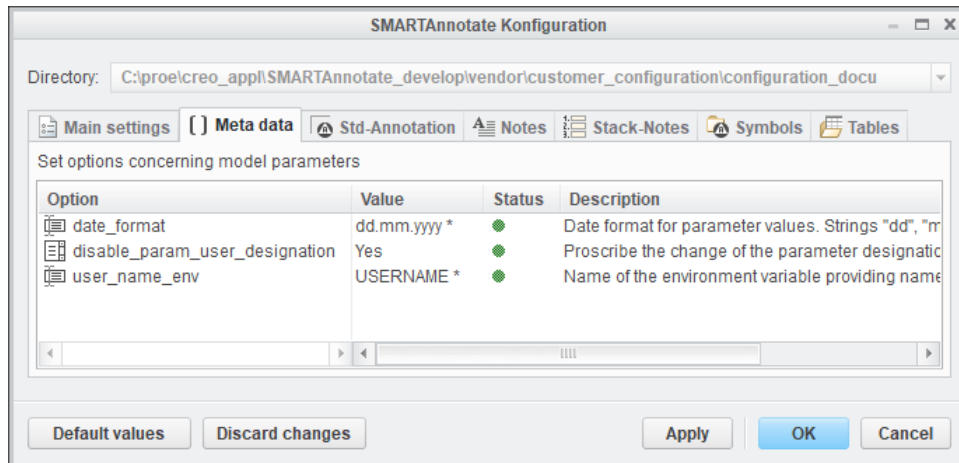
1. The current configuration **Directory** is displayed. Select another directory from the pull-down menu if available

2. Toggle between **Tabs** to manage the corresponding config options.

3. The **Options Table** contains the following columns

    *   **Option name** — Name of the config option

    *   **Value** — Select or enter a Value for the option.

    *   **Status** — Highlights options that are not applied to the current session

    *   **Description** — Explains the option and its usage.

4. Select the **Default Values** button to reset all option to its default values.

5. Select the **Discard Change**s button to undo all unsaved modifications.

General Configuration Options

Initial setup

*   *CUSTOMER_NAME*

Enter the company name. This option is used to distinguish **SMARTAnnotate** behavior for different companies.

- *LOAD_RIBBON_FILE*

  Show the **SMARTAnnotate** tab in the Creo ribbon.

  ○ YES — Enabled (default)
  ○ NO — Disabled

User dialog look & feel

- *INITIAL_DIALOG_HEIGHT*

  The initial dialog height. Manually changed size will be retained during a Creo session.

  ○ Enter a value in Pixel (300 default)

- *INITIAL_DIALOG_WIDTH*

  The initial dialog width. Manually changed size will be retained during a Creo session.

  ○ Enter a value in Pixel (650 default)
  ○ NO — Disabled

- *FILTER_TREE_WIDTH*

  Define the default width of the filter tree area when opening the user dialog. You can define anything between minimum and maximum size of the tree area. This value is corresponding to the INITIAL_DIALOG_ WIDTH setting. If this is too small you will not see any impact.

  ○ Enter an integer value between 0 and 100 (-1 default)

User dialog settings and permissions

- *SHOW_TAB_MAIN*

Show/Hide tab for managing main settings.

  ○ YES — Displayed (default)
  ○ NO — Hidden

- *SHOW_TAB_META_DATA*

  Show/Hide tab for managing parameters.

  ○ YES — Displayed (default)
  ○ NO — Hidden

- *SHOW_TAB_STD_ANNOTATION*

  Show/Hide tab for managing Std-Annotations.

  ○ YES — Displayed (default)
  ○ NO — Hidden

- *SHOW_TAB_NOTES*

  Show/Hide tab for managing notes.

  ○ YES — Displayed (default)
  ○ NO — Hidden

- *SHOW_TAB_STACK_NOTES*

  Show/Hide tab for managing stack notes.

  ○ YES — Displayed (default)
  ○ NO — Hidden

- *SHOW_TAB_GRAPHICAL_SYMBOLS*

  Show/Hide tab for managing symbols.

  ○ YES — Displayed (default)
  ○ NO — Hidden

- *SHOW_TAB_TABLES*

  Show/Hide tab for managing 2D-tables.

- YES — Displayed (default)
- NO — Hidden

- *ENABLE_FORMAT_MANAGEMENT*

  Enable the format management and show corresponding layout within the main tab.
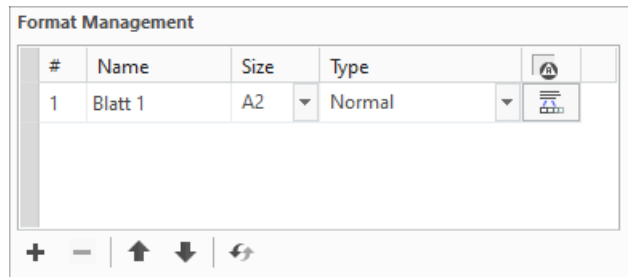
  - YES — Layout displayed (default)
  - NO — Layout not displayed

- *ENABLE_FILTER_FAVORITE*

  Enable user to set the favorite filter and show the corresponding button.
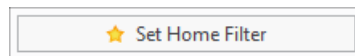
  - YES — Enabled
  - NO — Disabled (default)

- *ENABLE_GLOBAL_SEARCH*

  Allow to search for data by string input independent from filters. Disply the respective layout.
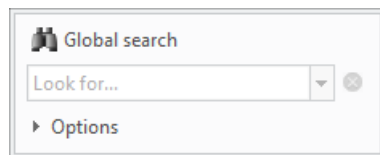
  - YES — Enabled (default)
  - NO — Disabled

- *GLOBAL_SEARCH_AUTO_CASE_SENITIVE*

Defines the behavior of the global search and checks the 'case sensitive' check box by default.
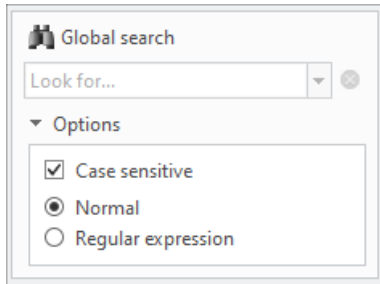
- ○ NO — (default)
- ○ YES

- *ENABLE_NAVIGATION_AREA_VIS_TOGGLE*

  Enable possibility to hide the navigation area.
  - ○ YES — Enabled (default)
  - ○ NO — Disabled

- *ENABLE_USER_PROFILES*

  Enable usage of custom user profiles and display the corresponding layout.
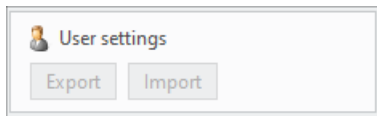
  - ○ YES — Enabled (default)
  - ○ NO — Disabled

Language options

- *ENABLE_MULTI_LANGUAGE*

Enable user to select additional languages and show the corresponding layout.

Additional languages:

German

- ○ YES — Enabled (default)
- ○ NO — Disabled

- *DEFAULT_LANGUAGES*

  Semicolon separated list of languages (e.g.: "usascii;german;french" will result in English as first, German as second and French as third language)

- *FIXED_MAIN_LANGUAGE*

  Folder name of the language which shall be unchangeable set as main language by default. Enter a folder name (i.e. usacii).

- MAX_ADDITIONAL_LANGUAGES

  Maximum number of languages which can be set by the user in addition to the main language. "-1" specifies no limitation and is the default value. For legacy models using more additional languages than allowed by this option, a dialog will prompt the user that the exceeding languages will be automatically removed from the current choice. This dialog might also be displayed, if the option "default_languages" contains more language names than allowed by this option.

- *ALLOW_BATCH_TRANSLATION_PARAM_NAME*

  Name of the designated parameter to be created which provides the information whether batch translation shall be allowed for the corresponding model/drawing. If empty, no parameter is created. The parameter will be created as read-only, designated and owned by PDM (source).

- USED_LANGUAGES_PARAM_NAME

  Name of the designated parameter to be created and filled with the suffixes of the used languages. Underscores ('_') will be removed from the suffixes and the list is separated via semicolons. A possible value might be "EN;DE;CN". If option value is empty, no parameter is created.

The parameter will be created as read-only, designated and owned by PDM (source).

Misc

- *ENABLE_MULTIUSER_ADMINISTRATION*

  Enable multiple administrators to work simultaneously on the configuration.

  ○ YES — Enabled
  ○ NO — Disabled (default)

- *ENABLE_EXCEL_IMPORT*

  Enable administrator to import data from prepared Excel sheets. A button will appear within the admin dialog.

  ○ YES — Enabled
  ○ NO — Disabled (default)

- *ENABLE_REVISIONS_CLEANUP*

  Enables the functionality to clean-up all revisions of objects and texts. Only the lastest revision will be kept, set to 1.0.0 and marked as unpublished.

  ○ YES — Enabled
  ○ NO — Disabled (default)

- *MAILTO_LINK*

  Specify the 'mailto' command to open within your standard email client. An empty value will hide the corresponding layout.
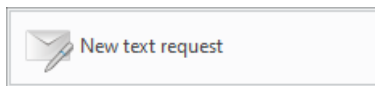
- *PROMPT_NOTIFS_AFTER_OBJECT_RETRIEVAL*

  Ask user whether to show notification center in case a model/drawing was opened in session, for which notifications were registered.

  ○ YES — Enabled
  ○ NO — Disabled (default)

- *RELEASE_STATE_PARAM_NAME*

  Enter the parameter name that identifies the drawing or model as 'released'

- *RELEASE_STATE_PARAM_RULE*

  Enter a regular expression to match the value of the 'Release State Parameter'

- *SOFTWARE_HELP_DISPLAY_MODE*

  Defines how the User or Admin Guide is displayed

  ○ Creo Browser — Internal Creo Browser(default)
  ○ Chm file — External window displaying CHM file.

- *MISSING_REVISIONS_PERFORM_CHECK*

  Activates the functionality to check if there are any missing revisions in the configuration

  ○ YES — Enabled
  ○ NO — Disabled (default)

- *MISSING_REVISIONS_PLACEHOLDER_TEXT*

  Predefines the placeholder text for missing revisions. This text will be entered instead of the missing revisions text.

  It can be empty or formated e.g. <textID> - <revision> - Text missing

  ○ <textID> - <revision> - Text missing (default)
  ○ Empty

- *MISSING_REVISIONS_OUTPUT_PATH*

  Sets the path to the logfile of missing revisions (missing_revisions.log).

  ○ Empty — The logfile will be created in the working directory.(default)
  ○ <path>

Meta Data Configuration Options

User dialog settings and permissions

- *DATE_FORMAT*

  Date format for parameter values that use the function 'Today's date'. Strings "dd", "mm" and "yyyy" have to be contained.

  - `dd.mm.yyyy` (default)

- *DISBALE_PARAM_USER_DESIGNATION*

  Lock the 'Designation' column for user changes. Inconsistencies between configuration and parameter can still be resolved by selecting **'Set designations'** in the user dialog.

  - YES — Enabled
  - NO — Disabled (default)

- *USER_NAME_ENV*

  Value of a environment variable providing the name of the logged-in user. Used by the 'Logged In' name function.

  - `USERNAME` (default)

Std-Annotation Configuration Options

Std-Annotations behavior

- *EXPAND_DIRECTION_2D*

  Definition of the general building direction for the Std-Annotation stack (drawing mode). Building origin is defined within each Std-Annotation set and refers to the drawing origin (bottom left corner)

  - bottom-up — Annotation stack is build from bottom up (default)
  - top-down — Annotation stack is build from top down.

- *EXPAND_DIRECTION_3D*

  Definition of the general building direction for the Std-Annotation stack (solid mode). Building origin is defined within each Std-Annotation set and refers to the default Csys in the model.

  ○ bottom-up — Annotation stack is build from bottom up

  ○ top-down — Annotation stack is build from top down (default)

User dialog look & feel

- *COLOR_MANDATORY*

  Background color for the header line of mandatory entries within the user dialog. Click into the colored area to open the **RGB** dialog to change the color.

  ○

- *COLOR_OPTIONAL*

  Choose color for the header line of optional entries within the UI. Click into the colored area to open the **RGB** dialog to change the color.

  ○

- *COLOR_REQUIRED*

  Choose color for the header line of required entries within the UI. Click into the colored area to open the **RGB** dialog to change the color.

  ○

- *OPTIONMENU_DEFAULT_COLOR*

  Choose background color for the area of the optionmenu for Std-Annotation sets. Click into the colored area to open the **RGB** dialog to change the color.

  ○

- *STD_ANNO_DEFAULT_SASH_POSITION*

  Defines the percentage of the display space for the 'preview' area of the Std-Annotations user dialog. '0' allows for the minimum preview space. '100' provides the maximum preview space.

  ○ Integer value (0 — 100)

User dialog settings and permissions

- *ENABLE_MARGIN*

  Enable user to type in an offset value to the next symbol within the Std-Annotation set. Values correspond to drawing units.

  Margin: 0

  ○ YES — enable (default)
  ○ NO — disable

- *ENABLE SEPARARTOR*

  Enable user to display a separator symbol between the entities of the Std-Annotation set. The symbol used is defined by the config option `SEPARATOR_SYMBOL`.

  ☐ Separator

  ○ YES — enable (default)
  ○ NO — disable

- *SEPARATOR_SYMBOL*

  Define the name of the separator symbol within folder `...\configuration\std_annotation\symbols`.

  ○ `buw_separator` (default) — Enter the name of separator symbol to be used.

- *ENABLE_SORTING*

Enable user to change the order of the entities within a Std-Annotation set.

- ○ YES — enable (default)
- ○ NO — disable

- *SHOW_ADD_BUTTON_PRECHECKED_ITEMS*

Additional button to allow the user to select all 'required' and 'mandatory' items from the Std-Annotations pool for automatic placement.

- ○ YES — enable
- ○ NO — disable (default)

- *UNCHECK_MANDATORY_ITEMS*

Setting this option to YES, will allow the user to deselect mandatory items from the Std-Annotations preview list.

- ○ YES — enable
- ○ NO — disable (default)

- *ENABLE_STD_ANNOTATION_FLAGS*

Enable the user to create flag symbols on the drawing referencing a specific block in the symbol stack. This affects only 2D mode.

- ○ YES — enable (default)
- ○ NO — disable

- *STD_ANNO_MIXED_FLAG_STATE*

Enable user to set a third state for the flags (= active | = inactive | = mixed). Mixed behaves like the active state, except that no notifications are prompted if no flag symbols were added on the drawing.

- ○ YES — enable
- ○ NO — disable (default)

Misc

- *ANNOTATION_FEATURE_NAME* (3D mode only)

  Define the name of the annotation feature which will contain the elements belonging to the standard annotation. If the annotation feature with this name does not exist, the software will automatically create it.

  - `STD_ANNOTATION` (default)

- *MOVE_ANNOTATION_FEAT_TO_FOOTER* (3D mode only)

  Defines whether to move the annotation feature to the model tree footer (solid mode only).

  - YES — move to footer (default)
  - NO — keep in regular model tree

- *STD_ANNO_USER_PROFILES_DIR*

  Path to the folder location for saving and retrieving user profiles. Environment variables can be called out by a leading '$' character e.g. "C:\temp\$MY_ENV_VAR".

- *ENABLE_FLAG_SYMBOL_PROTECTION*

  Flags are protected from user manipulation. If enabled, the flag gets deselected when the user attempts to select it. Changing the flag text is thus prevented. If disabled, the user can apply any actions to the flags, that Creo base functionalities provide. Therefore, all flag texts get updated automatically in the background, before the model/drawing gets saved.

  - YES — enable (default)
  - NO — disable

Notes Configuration Options

Notes behaviour

- *ALLOW_NOTES_USER_SELECTION*

  Via **SMARTAnnotate** created notes won't be selectable by the user. Use the 'Edit' commands to Edit, Move or Delete notes.

  ○ YES — User can select notes

  ○ NO — User cannot select notes (default)

- *NOTE_COLOR*

  The color that the note will display in drawings and models. Click into the colored area to open the **Color** dialog to select a color from the Creo default colors.

  ○

User dialog look & feel

- *UI_COLOR_HEADER_LINE*

  Background color for the header line of note entries within the UI. Click into the colored area to open the **RGB** dialog to change the color.

  ○

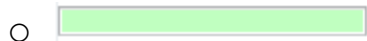- *NOTES_DEFAULT_SASH_POSITION*

  Defines the percentage of the display space for the 'preview' area of the Notes user dialog. '0' allows for the minimum preview space. '100' provides the maximum preview space.

  ○ Integer value (0 — 100)

User dialog settings and permissions

- *ENABLE_NOTE_SORTING*

  Defines if the user may sort the notes within the UI

  ○ YES — (default)
  ○ NO

Stack Notes Configuration Options

Stack Notes behavior (General)

- *STACK_NOTE_COLOR*

  The color that the regular stack note entry will display in drawings and models. Click into the colored area to open the **Color** dialog to select a color from the Creo default colors.

  ○

- *CUSTOM_NOTE_COLOR*

  Choose color for the custom stack notes to appear on the drawing or in the model..

  ○

- *NOTE_LISTING_TYPE*

  Defines the type itemization for selected notes.

  ○ NONE
  ○ NUMBERING —(default) a numbering scheme will be added to the selected notes.

- *NOTE_IDENTION_FORMAT*

  Used when NOTE_LISTING_TYPE is set to NUMBERING: Semicolon separated list of regular expressions to define the numbering indention.

○ [0-9].;[A-Z].;[a-z]. (default)

- *NOTE_INDENTION_FORMAT_ONLY_FOR_LAST*

Ignores the 'dot' between the list entries of the NOTE_INDENTION_ FORMAT list except the last one.

○ NO — (default)
○ YES

- *USE_INDENTION_FORMAT_WHEN_FLAGGED*

Allows the administrator to define a different indention format when the note is indicated as flagged.

○ NO — (default)
○ YES

- *ENABLE_FLAG_NOTE_PROTECTION*

Flags are protected from user manipulation. If enabled, the flag gets deselected when the user attempts to select it. Changing the flag text is thus prevented. If disabled, the user can apply any actions to the flags, that Creo base functionalities provide. Therefore, all flag texts get updated automatically in the background, before the model/drawing gets saved.

○ YES — enable (default)
○ NO — disable

- *ENABLE_MULTI_LANGUAGE_CUSTOM_NOTES*

Enable user to specify a custom note text independently for each chosen language.

○ YES — enable
○ NO — disable (default)

Stack Notes behavior (2D drawing)

- *GAP_LINES_2D*

Defines the number of empty break lines between the notes in the selection.

- 1.0 (default) — Enter an integer value for empty break lines.

- *INDENTION_SPACE_PER_LEVEL_2D*

  Defines the number of blanks that are added in front of each line of indented entries

  - 5 (default)

- *MIN_NUMBERING_TO_TEXT_SPACE_2D*
  The minimum number of spaces setting the margin between the longest listing string in the stack (e.g. <12A>) and the corresponding 2D note text.

  - 3 (default) — Enter an integer value

- *LISTING_FLAG_FORMAT_2D*

  Defines the display of the flag in Creo text syntax

  - @[%s@] — (default)

- *MAX_CHARS_PER_LINE_2D*

  Maximum allowed number of characters per line. If exceeded, line is broken at the last whitespace before the maximum is reached.

  - 45 (default) — Enter an integer value for number of characters

- *MAX_CHARS_PER_LINE_TABLE_FILE_2D*

  Name of a xml file within the path [configuration_dir]\stack_notes \column_width_tables, which allows to define the maximum allowed number of characters per line depending on the particular drawing sheet size. Option 'max_chars_per_line_2d' sets the default.

  - None (default)

- *FLAG_NOTE_PATTERN_FILE_PATH_2D*

  Name of a pattern file within folder <config>\stack_notes\patterns, that can be used to define the text attributes (font, height, bold...) for 2D flag notes.

○ None (default)

- *CUSTOM_NOTE_PATTERN_FILE_PATH_2D*

  Name of pattern file within folder <config>\stack_notes\patterns, that can be used to define the text attributes (font, height, bold...) for 2D custom notes.

  ○ None (default)

Stack Notes behavior (3D model)

- *DESTINATION_COMB_STATE_MODE*

  Either distribute the stack notes over multiple auto-created combined states or create all notes within a single pre-existing combined state. The combined state(s) is/are named as defined per option COMBINED_ STATE_NAME_FORMAT. If option is set to **Single**, then the placeholder "%s" gets removed at runtime from the option value and the result defines the name of the combined state that will hold the notes. The same behavior applies for the option ANNOTATION_FEATURE_ NAME_FORMAT.

  ○ Multiple (default)
  ○ Single

- *ANNOTATION_FEATURE_NAME_FORMAT*

  Define the name of the annotation feature which will contain the elements belonging to the standard annotation. If the annotation feature with this name does not exist, the software will automatically create it. %s will be substituted by the corresponding index if option DESTINATION_COMB_STATE_MODE is set to "Multiple".

  ○ `MBD1_%s_Notes` (default)

- *FLAG_ANNOTATION_FEATURE_NAME*

  Define the name of the annotation feature which will contain the 3D flag notes. If the annotation feature with this name does not exist, the software will automatically create it.

  ○ `MBD1_FLAGS` (default)

- *COMB_STATE_LAYER_STATE*

  Defines the layer state that is used in the specified combined state. In case the combined state needs to be created, this layer state is used.

  ○ None — (default)

- *COMBINED_STATE_NAME_FORMAT*

  Define the name of the combined state which will display the stack notes. %s will be substituted by the corresponding index if option DESTINATION_COMB_STATE_MODE is set to "Multiple".

  O `Mbd1_%s_Notes` (default)

- *COMB_STATE_VIEW_NAME*

  Defines the orientation that is used in the specified combined state. In case the combined state needs to be created, this orientation is used.

  ○ None (default)

- *INDENTION_SPACE_PER_LEVEL_3D*

  Defines the number of blanks that are added in front of each line of indented entries

  ○ 5 (default)

- *MIN_NUMBERING_TO_TEXT_SPACE_3D*

  The minimum number of spaces setting the margin between the longest listing string in the stack (e.g. <12A>) and the corresponding 3D note text.

  ○ 3 (default) — Enter an integer value

- *GAP_LINES_3D*

  Defines the number of empty break lines between the notes in the selection.

  ○ 2.0 (default) — Enter an integer value for empty break lines.

- *LISTING_FLAG_FORMAT_3D*

Defines the display of the flag in Creo text syntax

   ○ [%s] (default)

- *MAX_CHARS_PER_LINE_3D*

  Maximum allowed number of characters per line. If exceeded, line is broken at the last whitespace before the maximum is reached.

   ○ 72 (default) — Enter an integer value for number of characters

- *3D_PLACEMENT_HORIZONTAL_DEFAULT*

  Horizontal screen coordinates to start placing the flat to screen notes automatically

   ○ 30 (default) — Enter an integer value (0–1400)

- *3D_PLACEMENT_HORIZONTAL_HEADER*

  Horizontal screen coordinates to place the header of the stack automatically

   ○ 15(default) — Enter an integer value (0–1400)

- *3D_PLACEMENT_VERTICAL_END*

  Vertical screen coordinates to end placing the flat to screen notes and break into a next column.

   ○ 150 (default) — Enter an integer value (0–800)

- *3D_PLACEMENT_VERTICAL_START*

  Vertical screen coordinates to start placing the flat to screen notes automatically
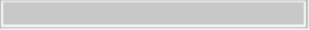
   ○ 790 (default) — Enter an integer value (0–800)

- *FLAG_NOTE_PATERN_FILE_PATH_3D*

  Name of a pattern file within folder <config>\stack_notes\patterns, that can be used to define the text attributes (font, height, bold...) for 3D flag notes.

   ○ None (default)

- *CUSTOM_NOTE_PATTERN_FILE_PATH_3D*

  Name of pattern file within folder <config>\stack_notes\patterns, that can be used to define the text attributes (font, height, bold...) for 3D custom notes.

  - ○ None (default)

User dialog look & feel

- *UI_COLOR_STANDARD_NOTE*

  Background color for the header line of note entries within the UI. Click into the colored area to open the **RGB** dialog to change the color.

  - ○

- *UI_COLOR_CUSTOM_NOTE*

  Choose color for the header line of custom note entries within the UI. Click into the colored area to open the **RGB** dialog to change the color.

  - ○

- *UI_COLOR_SELECTED_NOTE*

  Background color for the selected note in within the UI. Click into the colored area to open the **RGB** dialog to change the color.

  - ○

- *STACK_NOTES_DEFAULT_SASH_POSITION*

  Defines the percentage of the display space for the 'preview' area of the Stack Notes user dialog. '0' allows for the minimum preview space. '100' provides the maximum preview space.

  - ○ Integer value (0 — 100)

- *SHOW_HYPERLINK_COLUMN*

Displays an additional column in the user interface to display the hyperlink entry from the configuration. This hyperlink then may point into the Internet or local Intranet.

- ○ NO — (default)
- ○ YES

- *SHOW_ID_COLUMN*

   Show the UI table column exposing the ID of the corresponding note.

- ○ NO — (default)
- ○ YES

User dialog settings and permissions

- *LISTING_FLAG_ENABLE_MIXED_STATE*

   Allows the user to toggle between Flag, No Flag and a flag indication with no entry in the Flag tab.

- ○ NO — (default)
- ○ YES

- *ENABLE_CUSTOM_NOTES_BY_COPY*

   Enable user to create custom note (not translatable) by copying a note definition from configuration.

- ○ YES — enable
- ○ NO — disable (default)

- *ENABLE_CUSTOM_NOTES_BY_CSV*

   Enable user to create custom note (not translatable) by importing a note definition from a CSV file.

- ○ YES — enable
- ○ NO — disable (default)

- *ENABLE_CUSTOM_NOTES_BY_SELECTION*

   Enable user to create custom note (not translatable) by selecting an existing note within the model or drawing.

- ○ YES — enable
- ○ NO — disable (default)

- *ENABLE_CUSTOM_NOTES_BY_USER_INPUT*

  Enable user to create custom note (not translatable) by entering own text.

  - ○ YES — enable
  - ○ NO — disable (default)

- *ENABLE_STACK_NOTE_SORTING*

  Show/Hide the buttons, which allow to move the selected line within the table up or down.

  - ○ YES — enable (default)
  - ○ NO — disable

- *ENABLE_STACK_NOTE_CONVERSION*

  Allow the conversion from a standard note into a custom note. The conversion can be started by clicking into the corresponding table cell within the 'Value' column.

  - ○ YES — enable (default)
  - ○ NO — disable

- *ENABLE_USER_DEFINED_WIDTH*

  Show a button which allows the user to overwrite the pre-configured width (expressed as number of characters) for the stack note. Use the options "max_chars_per_line_2d", "max_chars_per_line_table_file_2d" and "max_chars_per_line_3d" to pre-define the width of the stack note.

  - ○ YES — enable
  - ○ NO — disable (default)

Misc

- *STACK_NOTES_USER_PROFILES_DIR*

Path to the folder location for saving and retrieving user profiles. Environment variables can be called out by a leading '$' character e.g. "C:\temp\$MY_ENV_VAR".

- *CUSTOM_NOTES_PRESERVE_EMPTY_LINES*

   Keep lines within user defined custom notes that are empty or just contain spaces.

   - YES — enable
   - NO — disable (default)

- *CUSTOM_NOTES_CSV_PATH*

   Enable user to create custom note (not translatable) by import from a CSV file.

   - YES — enable
   - NO — disable (default)

- *CUSTOM_NOTES_CHOP_RULE*

   Regular expression for separating the text of a selected note into multiple custom notes.

   - ^[\d]+\.[\s]+|^[@?\[]?[\d]+[@?\]]?[\s]+ (default)

- *CUSTOM_NOTES_CHOP_RULE_CSV*

   Regular expression for separating the text of a note into multiple custom notes during import from CSV file

   - ^[\d]+[\,]{2}|^[\,][A-Z][\.][\,] (default)

- *HEADER_DEFINITION_NAME*

   Name of the stack note definition to be used as header note. Working in 2D and 3D mode.
   You will find the name in the Stack Note definition dialog (i.e. def_ stack_note_6). If no other Stack Note is selected by the user, the header will not be displayed.

- *SHOW_HYPERLINK_COLUMN*

Show the UI table column allowing the user to view additional information in the browser.

Symbols Configuration Options

Symbol behavior

- *ALLOW_SYMBOLS_USER_SELECTION*

  Via **SMARTAnnotate** created symbols won't be selectable by the user. Use the 'Edit' commands to Edit, Move or Delete symbols.

  - ○ YES — User can select symbols
  - ○ NO — User cannot select symbols (default)

User dialog look & feel

- *SHOW_SYMBOL_PROCEED_DIALOG*

  Open small dialog for proceeding options during item placement.

  - ○ YES — Dialog available (default)
  - ○ NO — Disable dialog. Placement via mouse control only.

- *SYMBOLS_DEFAULT_SASH_POSITION*

  Defines the percentage of the display space for the 'preview' area of the Symbols user dialog. '0' allows for the minimum preview space. '100' provides the maximum preview space.

  - ○ Integer value (0 — 100)

Misc

- *SYMBOLS_USER_PROFILES_DIR*

  Default directory for retrieving and saving user configured symbols.

Environment variables can be called out by a leading '$' character e.g. "C:\temp\$MY_ENV_VAR".

Tables Configuration Options

Tables behavior

- *ARRANGE_SPACE*

  Distance between the tables after the arrangement.

  - 0 (default) — Enter value for spacing in drawing units.

- *ALLOW_TABLES_USER_SELECTION*

  Via **SMARTAnnotate** placed tables won't be selectable by the user. Use the 'Edit' commands to Edit, Move or Delete tables.

  - YES — User can select tables
  - NO — User cannot select tables (default)

- *DEFAULT_TABLE_SNAPPING*

  Activate table snapping by default for dynamic placement.

  - YES — Snapping active (default)
  - NO — Snapping disabled

User dialog look & feel

- *COLOR_UI_TABLE_ALTERNATE*

  Alternate row color for the UI table.

  -

11

API

Application Programming Interface (API)

The API of *SMARTAnnotate* enables to perform pre-defined tasks for certain Creo models in an automated and user independent manner. For instance, a Windchill worker can call the API as a batch process in order to translate a model or drawing and export it afterwards into a certain file format like PDF.

Invoking the batch run

Within the binary file folder of the *SMARTAnnotate* installation directory (x86e_win64), there is the file **san_api.exe**, which can be used to invoke the batch run via a command line call. This can be done e.g. using a batch file (*.bat). The call takes one input parameter, which is the file path to a so called "Jobs-File" (for more details see section).

This call of the API binary file will start the batch run:

<SmartAnnotate Installation directory>\x86e_win64**san_api.exe** <Any path>**san_jobs.xml**

Mandatory environment variables which have to be set before the call:

- **PRO_COMM_MSG_EXE**: Its value has to be set to the path of the file **pro_comm_msg.exe**, which is stored within the Creo installation folder under "<Creo Install Dir>\Common Files\x86e_win64\obj".

- **PTC_WF_ROOT**: The directory containing the workspace cache has to be empty!

- **PATH**: Make sure that the folder "<Creo Install Dir>\Common Files\x86e_win64\lib" is contained with the PATH variable, in order *ucore46.dll* and *udata64.dll* can be retrieved by the Creo process.

- **SAN_USER_CONFIG_PATH**: The directory containing the (customized) configuration of *SMARTAnnotate*, which shall be used to perform the tasks on the models.

Optional environment variables which can to be set before the call:

- **BUW_BATCH_LOGGING_DIR_PATH**: The log file 'smartannotate_api.log', giving details on the execution of the batch process will be

created in this directory. If not set, the path defaults to the directory containing 'san_api.exe'.

Jobs-File

The "Jobs-File" is in the XML format and defines which model(s) shall be processed and which tasks are to be performed for each of them. This file has to be created at runtime by the calling application, for instance a Windchill worker. The following nodes and attributes are supported:

- **<?xml>**: The declaration node of the XML file has two attributes:
  - ○ **version** (attribute): Use the value "1.0" for this attribute.
  - ○ **encoding** (attribute): Use the value "UTF-8" for this attribute. In general, the file should be encoded in UTF-8 to avoid errors, which might be caused by non-ascii characters.
- **<batch_process>**: Root node of the XML file. This node may contain one <session> node, one <server> node, one <application> node and an arbitrary number of <job> nodes.
- **<session>**: Information required for the batch process to start an asynchronous Creo Parametric session.
  - ○ **<creo_exec_path>**: Path to the executable (parametric.exe) or batch file (parametric.bat), which shall be used to start the Creo process. These files are located within the Creo installation folder under "<Creo Install Dir>\Parametric\bin".
  - ○ **<creo_start_path>**: The start directory path for the Creo Parametric process. This folder might contain configuration files like the file config.pro. If the batch process is not intended to connect to a Windchill server, the models to be processed should be contained within this folder. Alternatively, the config.pro defines other folders via the option *search_path*, where Creo will look for models.
  - ○ **<no_graphics_mode>**: Possible values "true" or "false". Using "true", the batch process will append the arguments "-g:no_graphics -i:rpc_input" to the file path defined under <creo_exec_path>. This will start a Creo Parametric session in non graphics mode. The value "false" will use the <creo_exec_path> as it is.
- **<server>**: Information required to register, activate and connect the session to an existing Windchill server.
  - ○ **<name>**: Server name (alias).
  - ○ **<location>**: Server URL (address).
  - ○ **<workspace>**: Name of the pre-existing workspace to use.
  - ○ **<user_name>**: User login for the authentication.

- ○ **<password>**: Password for the authentication.
- **<application>**: Paths required to load the SMARTAnnotate application within the Creo Parametric session.

  - ○ **<appl_exec_path>**: File path to *SmartAnnotate.dll*, which is stored within the folder "x86e_win64" of the SMARTAnnotate installation.

  - ○ **<appl_text_path>**: Directory path to the SMARTAnnotate installation folder, which contains the folder "text".

- **<job>**: A job-node relates to one model. If, for instance, five different models shall be processed, five job-nodes are required within the XML file. This node type has two mandatory attributes and contains at least one "task"-node.

  - ○ **model_name** (attribute): Name of the model to be processed without the file extension. E.g. for the drawing "demo_drawing.drw" the value "demo_drawing" has to be specified for this attribute.

  - ○ **model_type** (attribute): This attribute defines the type of the model to be processed. Specify "DRW" for drawings, "PRT" for parts and "ASM" for assemblies.

- **<task>**: A task-node calls one designated method to be performed for the job model. This node type has one attribute ("function_name") and may contain a variable number of sub-nodes. The amount and types of these sub-nodes are dependent of the called method. For instance, such a sub-node may define the output path for a PDF file to be created.

  - ○ **function_name** (attribute): Name of the task. See section for a complete list of supported tasks.

Tasks

The following tasks can be performed for a job model via batch process:

- **SANTranslateModel**: Translate all textual objects (symbols, notes, tables) within the job model, which were created using *SMARTAnnotate*, into specified language(s). This method requires one sub-node:

  - ○ **<languages>**: The value of the node has to contain a semicolon-separated list of language names (e.g. "usascii;german"). The names of the languages have to match with the names of the folders, which are contained in the SMARTAnnotate configuration directory *translations*.

- **SANCreatePDF**: A PDF file will be created for the job model and saved to disc. Four sub-nodes are supported:

- ○ **<output_path>**: File path including the file name and extension (e.g. "C:\temp\my_created_file.pdf").

- ○ **<orientation>** (optional): Specify the sheet orientation within the pdf file. Possible values are "landscape" and "portrait". Default is "landscape".

- ○ **<paper_size>** (optional): Sheet size for the pdf pages. Possible values are "A0", "A1", "A2", "A3", "A4", "A", "B", "C", "D", "E" and "F". For drawings, if not specified, the sheet size will be taken from the drawing sheet as it is. For solid models, the default is "A4".

- ○ **<resolution>** (optional): The DPI resolution used for the rendering of the solid geometry views. Possible values are "100", "200", "300", "400", "500" and "600". The default resolution is "300".

- ○ **<font_stroke>** (optional): Define whether to stroke all fonts (default) or use true type fonts for the PDF output. Possible values are "stroke_all_fonts" and "true_type_fonts". Default is "stroke_all_fonts".

- ○ **<color_depth>** (optional): Set the color depth either to "color" (default), "grayscale", or "monochrome".

- **SANCreatePVZ**: Export a Creo View file (*.pvz). Valid for solid models and drawings.

  - ○ **<output_path>**: File path including the file name and extension (e.g. "C:\temp\my_created_file.pvz").

- **SANCreateTIF**: Export a TIF raster file (*.tif).

  - ○ **<output_path>**: File path including the file name and extension (e.g. "C:\temp\my_created_file.tif").

  - ○ **<gs_exec_path>**: File path to the GhostScript executable *gswin64c. exe* (e.g. "C:\Program Files\gs9.50\bin\gswin64c.exe").

  - ○ **<printer_type>**: Define the printer type for postscript output. This may either be "COLORPOSTSC", which is the default and provides colored results or "POSTSCRIPT" providing monochrome results.

Example

The following shows an example for a Jobs-file which translates a drawing using English (usascii) as primary language and German as secondary language. After translation the drawing is exported to the formats PDF, Creo View and TIF giving the output file paths.

```
<?xml version="1.0" encoding="UTF-8"?>
<batch_process>
<session>
```

```xml
<creo_exec_path>C:\Creo 4.0\M100\Parametric\bin\parametric.bat</creo_exec_path>
<creo_start_path>C:\temp\batch_start_dir</creo_start_path>
<no_graphics_mode>true</no_graphics_mode>
</session>
<server>
<name>Windchill</name>
<location>https://wf.company.com/Windchill</location>
<workspace>my_workspace</workspace>
<user_name>root</user_name>
<password>1234</password>
</server>
<application>
<appl_exec_path>C:\SMARTAnnotate\x86e_win64\SmartAnnotate.dll</appl_exec_path>
<appl_text_path>C:\SMARTAnnotate</appl_text_path>
</application>
<job model_name="demo_drawing" model_type="DRW">
<task function_name="SANTranslateModel">
<languages>usascii;german</languages>
</task>
<task function_name="SANCreatePDF">
<output_path>C:\temp\my_created_file.pdf</outout_path>
<orientation>landscape</orientation>
<paper_size>A3</paper_size>
<resolution>300</resolution>
</task>
<task function_name="SANCreatePVZ">
<output_path>C:\temp\my_created_file.pvz</output_path>
</task>
<task function_name="SANCreateTIF">
<output_path>C:\temp\my_created_file.tif</output_path>
</task>
</job>
</batch_process>
```

SMART Annotate

12

Appendix

File structure of the configuration

The configuration of **SMARTAnnotate** is a collection of files and folders which are bundled within the configuration folder.

The configuration folder can reside anywhere in your local file system or in a server environment. Use the environment variable `SAN_USER_CONFIG_PATH` to define the location.

Note

Do not modify the files within this structure manually. You may add files like formats (.frm), symbols (.sym), images (.png, .jpg) etc. An exception are the pattern files within the `notes/patterns` folder.

The Folders and Files Structure

- 📁 <configuration_dir>
  - 📁 filters
    - ◆ 🖹 san_filters.xml
  - 📁 flags
    - ◆ 📁 images (contains images to be used by flags)
    - ◆ 📁 symbols (contains symbols to be used by flags)
    - ◆ 🖹 san_flags.xml
  - 📁 formats
    - ◆ 📁 frm (contains format files .frm))
    - ◆ 📁 sym (contains symbols to be used within drawing formats)
    - ◆ 📁 tbl (contains tables that are used as title blocks)
    - ◆ 🖹 san_formats.xml
  - 📁 notes
    - ◆ 📁 patterns (contains text files .txt to define the layout of notes)
    - ◆ 🖹 san_notes.xml
  - 📁 parameters
    - ◆ 📁 lists (contains .txt files with values to be used in drop-down choices)
    - ◆ 🖹 san_paramters
    - ◆ 🖹 san_parameter_sets.xml

- 📁 stack_notes
  - 📁 patterns (contains text files .txt to define the layout of stack notes)
  - 📄 san_stack_notes.xml
- 📁 std_annotations
  - 📁 symbols (contains symbols to be used by Std-Annotations)
  - 📄 san_std_annotations.xml
  - 📄 san_std_annotation_sets.xml
- 📁 symbols
  - 📁 images (contains images to be used within the symbols tab in the UI)
  - 📁 sym (contains symbol files .sym)
  - 📄 san_symbols.xml
- 📁 tables
  - 📁 images (contains images to be used within the tables tab in the UI)
  - 📁 tbl (contains the table files .tbl)
  - 📄 san_tables.xml
- 📁 translations
  - 📁 <languages> (one folder per language that contain the texts to be used in the specified language)
  - 📄 san_main_language.xml
- 📁 user_interface
  - 📁 company_logo (contain the logo used in the UI)

The Pattern file

Creo symbols can contain variable texts to define a location within the symbol definition that can be filled with custom text. Creo notes do not provide a similar skeleton, that allows to define the text arrangement.

For that reason the **Pattern file** was introduced. It is a regular text file that drives the layout of a Creo note.

It may look like the following example, that creates a note with two line, first line gives the text in the main language and the second line in the second language:

```
\[L1][A]\
\[L2][A]\
```

- If there is only one language chosen by the user, the note will just contain one line.

- \...\ — The two backslashes mark a callout for the text to be translated.

- [L1] — indicates the language to be used in this layout text area. While "L1" equals the main language, "L2" matches the second, "L3" the third and so on.

- [L1(1)], [L1(2)], [L1(n)] will callout a specific line of the translation.

- [A] — Specifies a text placeholder that will be replaced with the corresponding text during note placement. The link between the placeholder and the text is established through the **Note** object definition in the administrator dialog.

- The positions for values that can be entered or chosen by the user will be defined directly within the translations. There applies the same syntax like for symbols: Here an example text: "Please enter your name: \[V]\". The prompt "V" is again referenced within the **Note** object definition in the administrator dialog.

- In 2D mode, the note texts can be formatted using the following attributes:

  o Slant angle: <slant_angle="35">...</slant_angle>

  o Italic: <i>...</i> (shortform for <slant_angle="15">...<slant_angle>)

  o Thickness: <thickness="0.3">...</thickness>

  o Bold: ... (shortform for <thickness="0.24">...</thickness>)

  o Underlined: <u>...</u>

  o Line height: <height="5.0">...</height>

  o Width factor: <width_factor="1.0">...</width_factor>

  o Font name: <font="arialuni">...

  o Line break:

  Text formatting example:

```
<height="5.0">\[L1(1)][A]\</height>
<u>\[L1(2)][A]\</u>
<i>\[L1(3)][A]\</i>
<b>\[L1(4)][A]\</b>
<height="3.0"><br>
<font="consola">\[L2(1)][A]\</font>
<slant_angle="35">\[L2(2)][A]\</slant_angle>
<thickness="1.0">\[L2(3)][A]\</thickness>
<width_factor="1.5">\[L2(4)][A]\</width_factor>
```
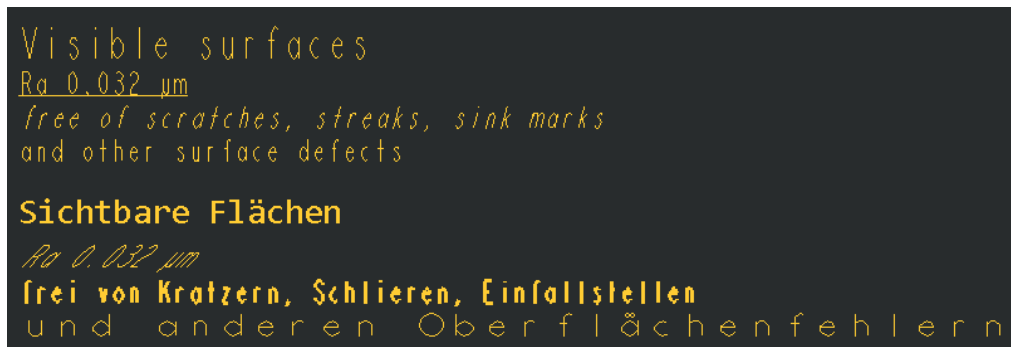
If the font "font.ndx" is used as default font, the 2D note will look like this:

Note:

- ○ For true type fonts bold style, thickness and width factor won't take effect.
- ○ The closing tags like </u> or </height> are not mandatory, as an opening tag (<u> or <height>) will apply to the whole pattern file line.
- ○ Text formatting cannot be provided for 3D notes due to lacking API functionalities.

How to link Format Definitions to .tbl Entries

When setting up your **Format** definition, you may want to know how to finally link this definition to your real .tbl entries. In general you have to distinguish between

- Fixed text entries that are not translated (i.e. simple text)
- Fixed text entries that will need to be translated (i.e. table cell titles with translation)

- Parameter Values without a translation (i.e. creation date)
- Parameter values that will have a translation with it (i.e. article name)

Creating a .tbl file to be used in the Format definition

1. Create a new Creo Format (.frm) or open an existing one
2. Make sure the correct .dtl file is used to get the correct font and text sizes.
3. Create a new table or retrieve any table from file.
4. Design your table layout as desired.

   - Allow the underlying table grid to provide space for table cell titles and call out parameters
   - Blank all unnecessary lines
   - Different line styles can be achieved by overlaying tables
   - Use different colors for texts and titles
5. Save the .tbl file to you configuration folder `configuration/formats/tbl`

Adding a fixed text entry that has no translation

1. Simply type this text into your table cell.
2. Make sure that the font is set to default. Otherwise is will not adopt to your drawing later.

Adding a call out parameter value without a translation

1. Type `&parametername` into your table cell.

   - Use `&parametername` if the value is called out from the active 3D-model.
   - Use `&parametername:D` if the value is called out from the drawing itself.
2. Make sure that the parameter value will have enough space in the cell.

Note

You can use the Meta data functionality from **SMARTAnnotate** to manage your parameters, but you also might use your own or default functionality.

## Adding a fixed text that has a translation with it

1. Open the **SMARTAnnotate** Administration tab and open a **Format** [icon] definition from the object list.

    • The **Object Edit** dialog appears.
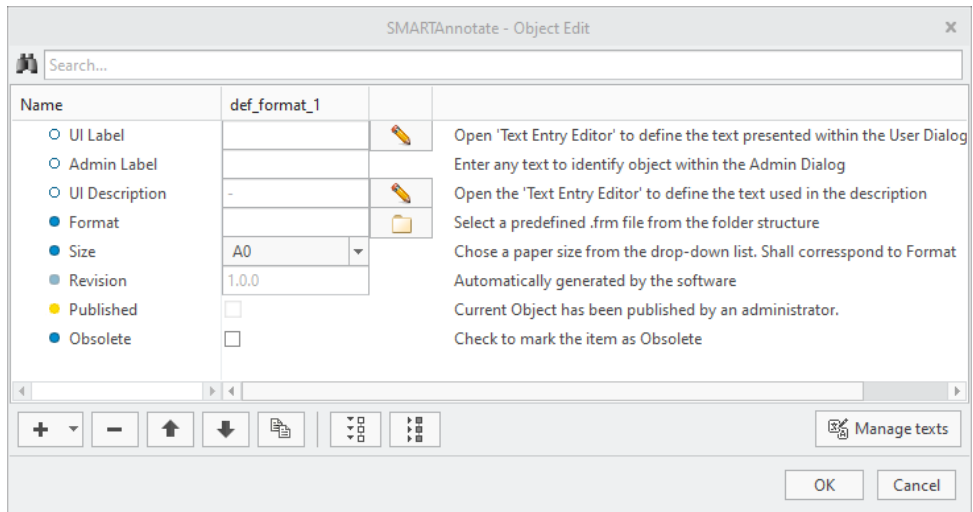
2. Select [icon] to open the **Edit Text** dialog

3. Select **New text entry** to create a new translation text set, fill in the texts and **Save** the entry.

📝 Note

Note the text ID from the window title bar because you will need that in your table definition in Creo.

---
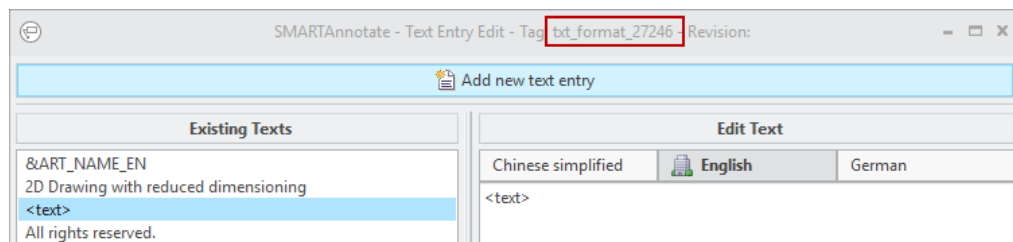
4. Change into Creo and your table definition

5. Enter the text into the table cell as follows

    • Only main language is used

```
\[L1][txt_format_27246]\
```

- dual language is used

```
\[L1][txt_format_27246]\\ / [L2][txt_format_27246]\
```
Note that the separator symbol '/' is included in the L2 definition brackets. If L2 is not used the entry including the separator symbol will be blanked. You may use other separator symbols but backslash is prohibited.

Adding a Parameter Value that has a translation with it

1. Follow the steps from the previous section until you enter the text entry
2. Enter the text as shown
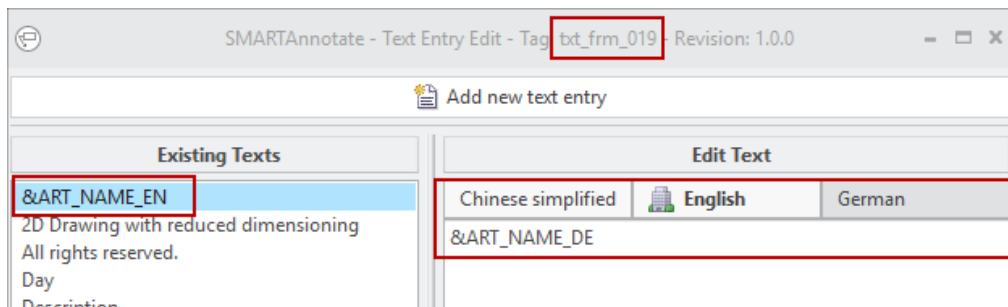
- The parameter that is used in this example is ART_NAME. The language appendix is created by the software automatically, but you have to use it within the text to call out the different translations within the Creo table.

3. Change into Creo and your table definition
4. Enter the text into the table cell as follows

   - To call out the parameter value in main language

     ```
     \[L1][txt_frm_019]\
     ```

   - To call out the parameter value in second language

     ```
     \[L2][txt_frm_019]\
     ```