

GENIUS TOOLS Model Processor User

12.0.3.0

Installation

© 2026 INNEO Solutions GmbH



Contents

1	Installation	2
1.1	Local installation	2
1.2	Installation mit GENIUS TOOLS Starter	4
1.3	Update	6
1.4	Multiple Installations	7
2	Configuration	8
2.1	MPU configuration file	8
2.2	Configuration Options	9
2.3	Configuration of triggers	12
2.4	Configure ribbon menu	14

1 Installation

With GENIUS TOOLS Model Processor User, CAD users can execute task lists for data processing in Creo Parametric, e.g., to adapt catalog parts to company standards. With this license, you have access to the defined task lists that are created with GENIUS TOOLS Model Processor Rework or Report.

There are two options available for installing the GENIUS TOOLS Model Processor User:

1. [Local installation](#) 
2. [Using GENIUS TOOLS Starter in network mode](#) 

1.1 Local installation

Follow these steps to install GENIUS TOOLS Model Processor User as a standalone application.

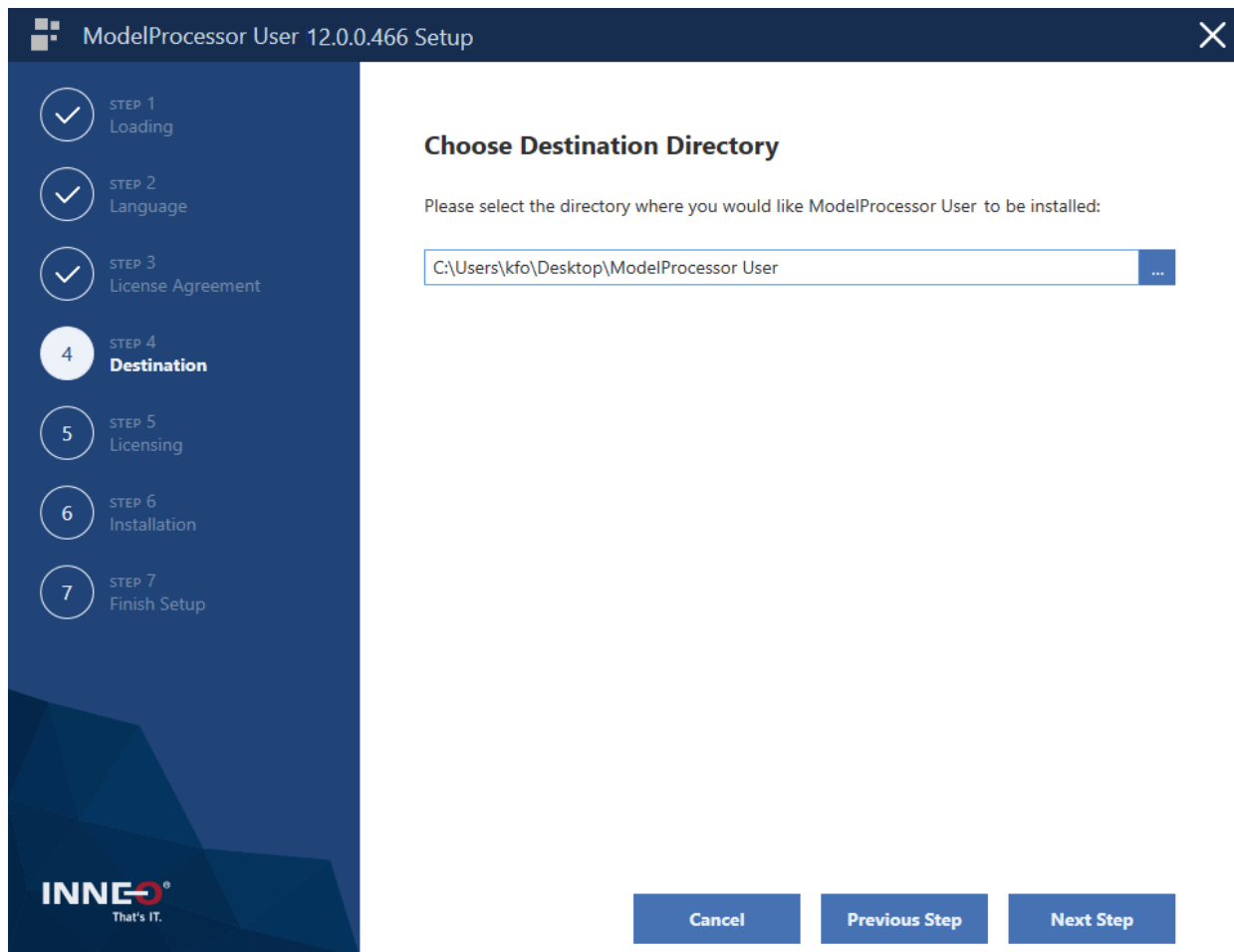
Prerequisites

GENIUS TOOLS License Manager must be accessible either locally or on the network.
The license for Model Processor User must be available.
Creo Parametric must be closed before installation.

Procedure:

1. Download the GENIUS TOOLS Model Processor software package. A ZIP file containing the software will be saved on your computer. The ZIP file contains the installation file *setup_GT-ModelProcessorUser_<Version>.exe*.
2. Start the installation file,
3. Select the language.
4. Read and accept the license agreement.
5. Specify the target directory for installation. The default directory is *C:\Users\<User>\Desktop\ModelProcessor User*.

Please note: Users require read access for the installation directory.



5. Install GENIUS TOOLS Model Processor

6. Complete installation


Please note: If you move the installation directory to another location, run the file *rewriteCreoTK.cmd*. This will automatically correct the folder paths in the relevant configuration files.

Create directory for task lists

7. Create a directory where you will store the MPX files which contain the action lists, e.g., *<installation directory>/MPXs*.

Customize the configuration file

8. In the conf directory, open the [MPU configuration file](#)  *mpu_main.cfg*.

9. Enter the path to the MPX directory and the license server in the corresponding configuration options; see [List of Configuration Options](#) .

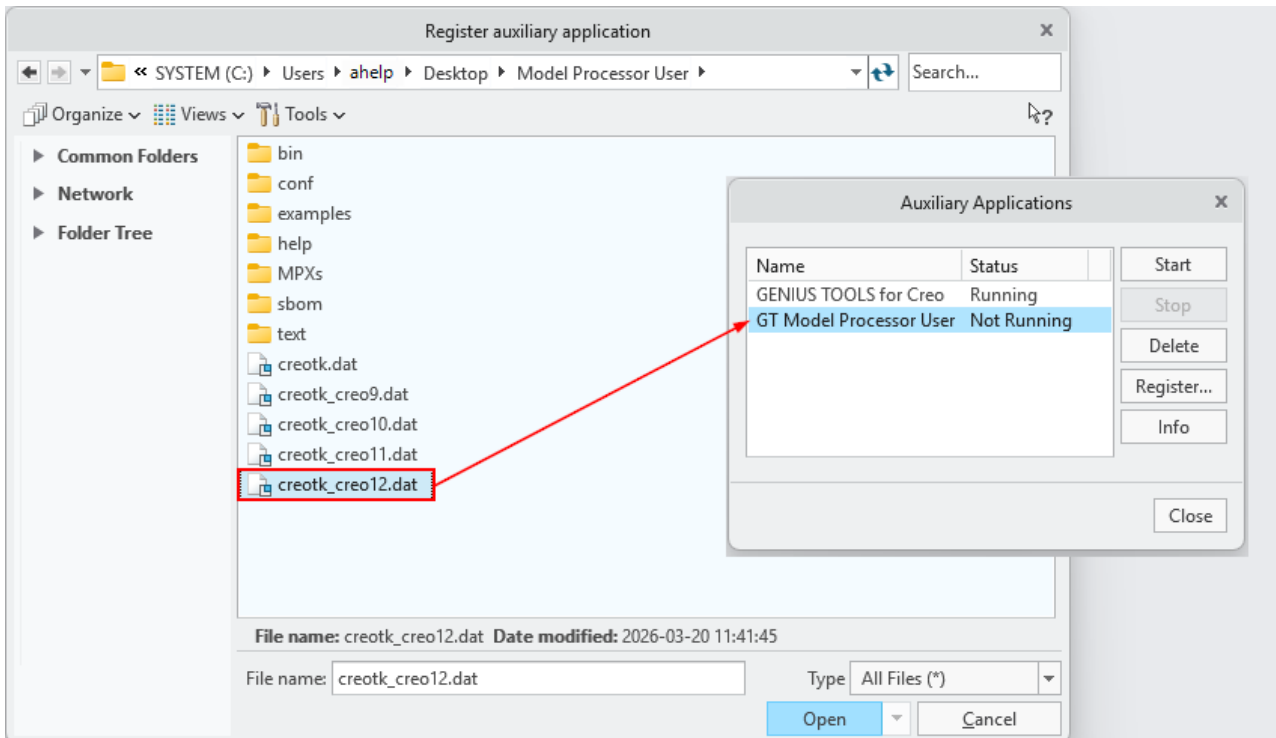
Manually add GENIUS TOOLS Model Processor User in Creo

10. Click in Creo on the Home ribbon tab *Utilities > Auxiliary Applications*.

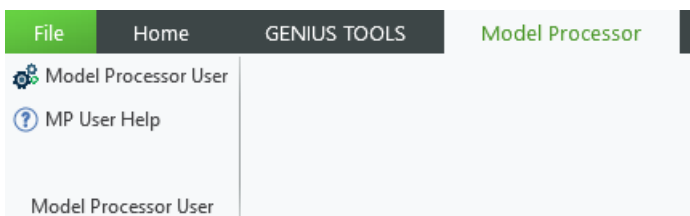
11. Click on *Register...*

12. From the Model Processor User installation folder, select the relevant DAT file, e.g., *creotk_creo12.dat* for Creo 12.

13. Click the row *GT Model Processor User* and then the Start button.



Result: The ribbon *Model Processor* appears. The user interface can be modified to include buttons for task lists, see [Configure ribbon menu](#).¹⁴



1.2 Installation mit GENIUS TOOLS Starter

GENIUS TOOLS Starter is an application for creating and managing operating environments. It is a component of Startup TOOLS. To integrate Model Processor User into a Starter environment, follow these steps.


Prerequisites

GENIUS TOOLS License Manager must be accessible either locally or on the network.
The license for Model Processor User must be available.
Creo Parametric must be closed before installation.

Create installation directory

1. In the Caddepot of the operating environment, create a folder named *mpuser* under *<OperatingEnvironment>\parametric\apps*.

Run setup

2. Install Model Processor User in this folder, as described in steps 1-6 in the chapter [Local installation](#) .


Please note: Installation via the network path is not supported.

Create directory for MPX files

3. Create a folder named *mpx*. This is the folder where the MPX files (task lists) will be stored, e. g.
<OperatingEnvironment>\parametric\configuration\mpx

Tip: You can copy the file *mpuser_example_int.mpx* from the examples directory to this location.


Set the path to the license server

4. In the conf directory, open the [MPU configuration file](#)  *mpu_main.cfg*.
5. Modify the following configuration option:
mpu_mpx_path – Path to the directory storing the MPX files.
To do this, you can comment out the entry on line 20 by adding a semicolon and use the entry below it:

```
mpu_mpx_path=%GTS_CONFIGURATION_DIR%\mpuser
```

6. Modify the following configuration option:
mpu_lic_path – address of GENIUS TOOLS License Manager, must be specified with *PORT@IP* or *PORT@NAME*.
To do this, you can comment out the entry on line 22 by adding a semicolon and use the entry below it:

```
mpu_lic_path=%GT_LIC_SERVER%
```

7. Specify further configuration options, if needed, see [Configuration Options](#) .

Create project option for Model Processor User in GENIUS TOOLS Starter App

8. Copy the file *config_sut_de_start_mpu.pro* in the installation directory.
9. Place the file in the desired configuration directory (standard, units, projects, user) under
<OperatingEnvironment>/parametric/configuration

For information on the configuration directories, see the *Starter Projects* chapter in the GENIUS TOOLS Starter Manual.

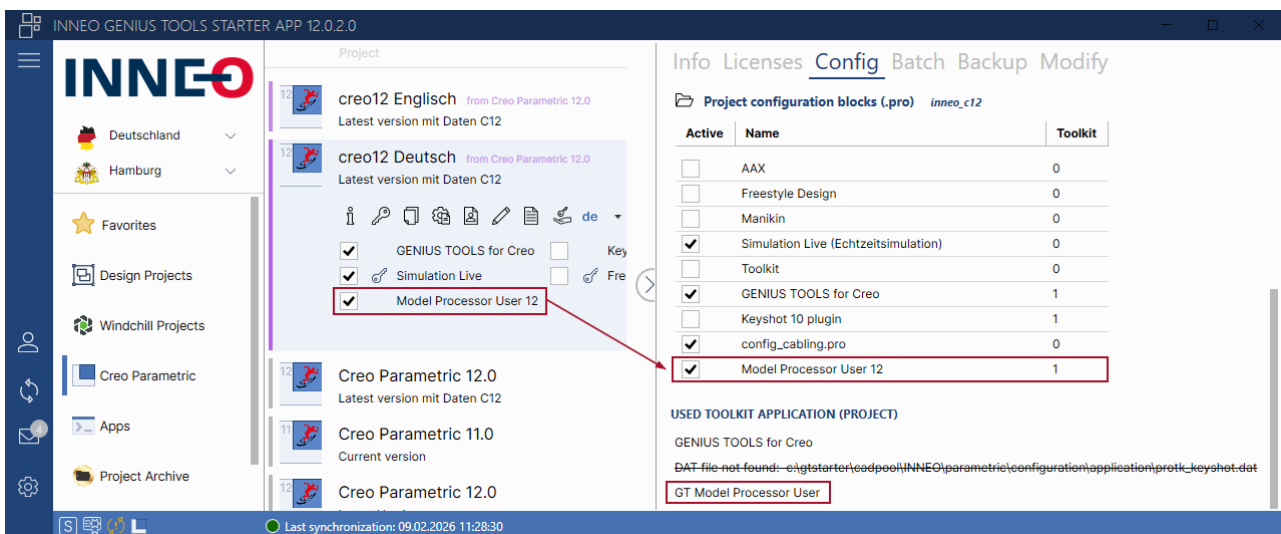
10. Edit the file if you want to customize the project option display. For doing this, change the GTS variable in the first five lines or add additional variables.

Information on creating project options are found the chapter *GENIUS TOOLS Project Configurator > Making use of project options* in the GENIUS TOOLS Starter Manual.

11. You can open and edit this file (configuration module) later in the GENIUS TOOLS Starter app by double-clicking it in the *Config* tab.

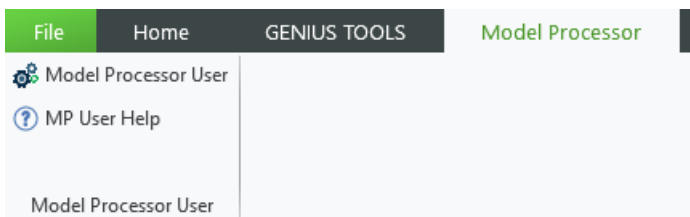
Result:

– The add-on application Model Processor User is added as a project option (checkbox).



Starter project with option to activate Model Processor User

Result: The ribbon *Model Processor* appears in Creo Parametric. The user interface can be modified to include buttons for task lists, see [Configure ribbon menu](#).¹⁴



1.3 Update

If you have already installed GENIUS TOOLS Model Processor User and would like to update the existing installation, follow the steps for initial installation.

During an update, the configuration file `mpu_main.cfg` in the folder `<installation directory>\conf` is not updated so that no company-specific configuration settings are overwritten. When updating, you can find the current configuration options in the file `mpu_main_default.cfg` or in the list in the chapter [Configuration Options](#).⁹

1.4 Multiple Installations

It is possible to install two GENIUS TOOLS Model Processor Users. The first option is to rename one Model Processor User, for example to MPUserBeta or MPUser<version number>. The second option is to install the Model Processor User in a different folder.

2 Configuration

2.1 MPU configuration file

The file containing the configuration settings for Model Processor User is called *mpu_main.cfg*. The file

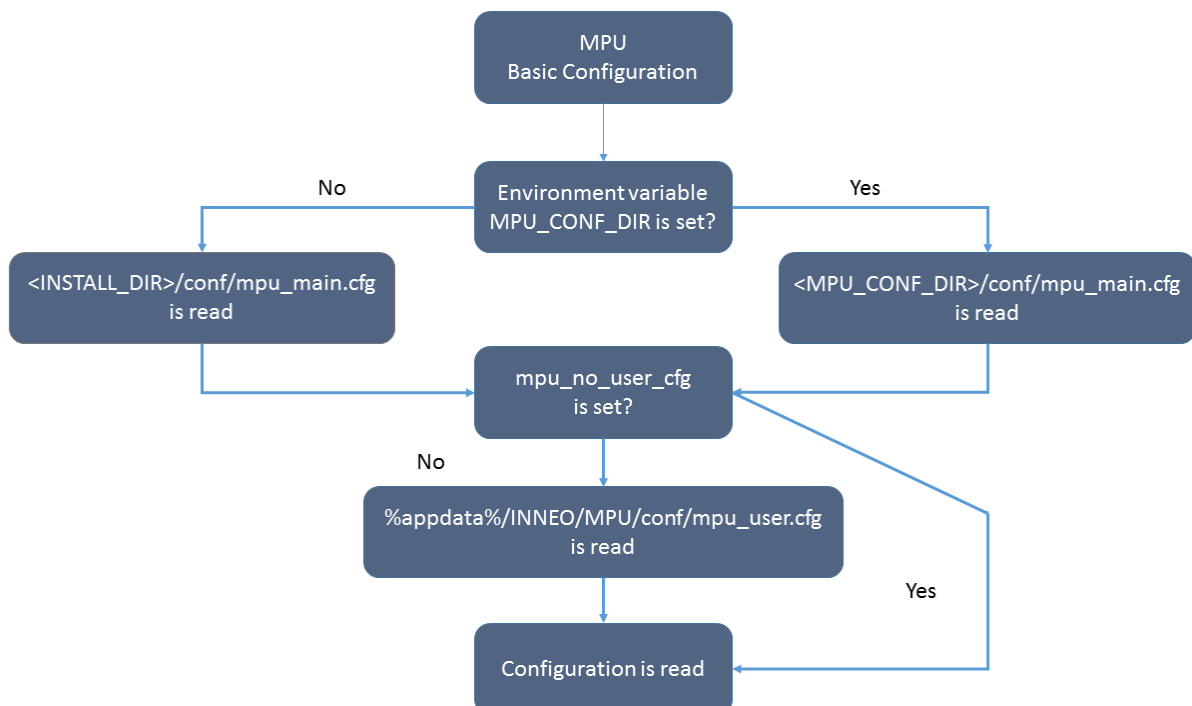
- contains all configuration settings in German and English with a description,
- must be saved with ANSI character encoding,
- is located in the conf directory of the MPU installation. It can also be stored in another file folder. To do this, this folder must be specified via the `MPU_CONF_DIR` environment variable.


Please note: Make sure that no extra spaces are used in the lines and that there is a blank line at the end of the file.

You can also modify configuration options on a user-specific basis. To do this, create an *mpu_user.cfg* file in the user directory at

`%appdata%/INNEO/MPU/conf`

You must create the MPU directory and the conf subdirectory yourself. The same rules apply as in the *mpu_main.cfg* file.



When updating the Model Processor installation, you can find the new configuration settings in the *mpu_main_default.cfg* file. The *mpu_main.cfg* file remains unchanged, see [Update](#) .

2.2 Configuration Options

There are four configuration options in the MPU configuration file that, if incorrect, can cause the system to malfunction. Other configuration settings are optional.

To ensure that log files and runtime behavior can be reviewed later, the temporary directory should be located in a directory that is accessible and readable by regular users. For testing purposes, the deletion of log files can be disabled in the configuration.

Necessary configuration options with default value	Explanation
mpu_mpx_path=C:\temp\MPXs\	Path to MPX directory, where the MPX files are located. - read-only access
mpu_lic_path=7766@localhost	Address of the license server - read-only access
mpu_temp_path=%TEMP%	A temporary path that is as flat as possible, used to write the log files. - read and write access
mpu_ext_filter_file=mp_check.xml	If an automatically selected filter exists, this can be entered here.
Further configuration options with default value	Explanation
mpu_auto_unset_already_reworked_models=0	Specifies whether a preselection (1) of the displayed data should also be performed after the external data, or not (0).
mpu_cancel_if_not_checked_out=0	Only revise models that have been checked out in Windchill.

Necessary configuration options with default value	Explanation
mpu_check_modifyable=0	Checks whether MPU has write access to the models.
mpu_check_parameter_value=1	Defines a corresponding value that should be within the defined parameter for mpu_check_parameter.
mpu_check_parameter	Defines a parameter to be checked.
mpu_column_parameter_width=10	Specification the width in the selection window.
mpu_column_parameter=BENENNUNG	Specification of the designation in the selection window
mpu_disable_generating_text_folder=0	This configuration defines whether the text folder is written to the temporary system path (0) or not (1). If the value is set to 1, the protk.dat file must be set to the installed DLL folder or a suitable text folder.
mpu_filter_asm=1	Specifies whether the Model Processor User is started (1) or not (0) when opening an assembly.
mpu_filter_curmod=1	Specifies whether the Model Processor User is started (1) or not (0) when opening the current model.
mpu_filter_drw=1	Specifies whether the Model Processor User is started (1) or not (0) when opening drawing.
mpu_filter_inst=1	Specifies whether the Model Processor User is started (1) or not (0) when opening Instance.
mpu_filter_prt=1	Specifies whether the Model Processor User is started (1) or not (0) when opening part.

Necessary configuration options with default value	Explanation
mpu_lang=de	Specifies whether the language setting of the log (not in the user interface) is displayed in German (de) or English (en).
mpu_log_as_tree=1	Specifies whether the log output is displayed as a tree (1) or as a list (0).
mpu_log_show_filter_warning_on_parent_node=0	Specifies whether the warning symbols are inherited (1) or not (0) for filter warnings.
mpu_no_debug_in_creo_log=0	No log is output directly in Creo Parametric, but rather in the MPU log file.
mpu_no_pro_notifications=1	MPU reacts (0) to the Creo model changes or it is deactivated (1).
mpu_pdm_mdl_change_dialog=0	Suppresses (0) the Windchill dialog that pops up when modifying read-only models.
mpu_show_log=0	The log area is folded down and is not filled (0) or is unfolded and is filled (1).
mpu_show_no_debug_in_log=0	Specifies whether a complete output of all messages defined in mpu_no_debug_in_creo_log is written to the Creo log file (1) or not (0).
mpu_show_no_pdm_server_information=1	Does not display the model information from the PDM server in the model list.
mpu_start_mpx_	Automatically revises the selected data when saving with the selected MPX and task list (without file extensions).
mpu_start_trigger_without_mpu=1	Specifies whether the use of the MPU license is disabled (1) or not (0).

Necessary configuration options with default value	Explanation
mpu_use_global_filter=0	Specifies whether the global filter of the task list is always evaluated (1), not evaluated (0), or not evaluated when it is activated (-1) in the list.
mpu_write_history_in_models=0	Write a parameter to all revised models that indicates whether the model has been edited.
mpu_write_id=1	Writes the application ID of the loaded DLL to the specified file.
mpu_write_reports_in_time=1	Reports are written to the file (1) each time the model is changed or when the MPU run is terminated (0).
mpu_write_welcome_message=1	Specifies whether the welcome message should be displayed (1) or not (0) when the MPU is started.
mpx_button_filter=^mpuser_example_.*\mpx\$	This configuration defines a filter for MPX files that are displayed as buttons in the Creo ribbon. The filter must be a regular expression.

2.3 Configuration of triggers

The Model Processor User offers the option of automatically revising files when defined events occur. Each trigger only supports one task list. Duplicating configuration options does not result in multiple task lists being executed.

The desired triggers must be defined in the [mpu_main.cfg](#) file. The following triggers are supported:

- Trigger before saving a model: `before_save`
- Trigger after opening a model: `after_open`

For the desired trigger, enter the trigger name *before_save* or *after_open* in the following configuration options. These settings must be configured.

Configuration options for triggers	Explanation
mpu_start_mpx_<Trigger name>=MPX file TaskList	<p>Defines the MPX file to be executed and the task list.</p> <ul style="list-style-type: none"> – MPX file: Name of the MPX file without the file extension – TaskList: Name of the task list to be used <p>ATTENTION: Die Creo onfiguration option <i>retrieve_instance_dependencies</i> must be set to <i>instance_req_generic</i> * for <i>after_open</i>.</p>
mpu_start_mpx_<Trigger name>_conf	Specifies the filter selection for revision with the respective trigger (component, assembly, drawing, instances, and current model) as on (1) or off (0), see section Filter selection ¹³ below.
mpu_start_mpx_<Trigger name>_showUI	Specifies whether the user interface should be displayed after being revised by the respective trigger (1) or not (0).
mpu_start_mpx_<Trigger name>_write_check_rework_history	Specifies whether the history of a model should be evaluated (1) or not (0) and whether the history should be written to the models (1) or not (0).

Example:

```
mpu_start_mpx_before_save=autompx second_task_list
```

```
mpu_start_mpx_before_save_write_check_rework_history=01
```

```
mpu_start_mpx_before_save_conf=11101
```

```
mpu_start_mpx_before_save_showUI=0
```

An MPX file is available as an example of how to define tasks under
 <installationdirectory>\examples\mpuser_example_int.mpx.

Filter selection

The *mpu_start_mpx_after_open_conf* option defines which model types are considered when the trigger is started. The default value is *mpu_start_mpx_after_open_conf=11111*. The sequence of digits indicates the following model types (1 = task is executed, 0 = task is not executed):

```
mpu_filter_prt=1 (part)
```

```
mpu_filter_asm=1 (assembly)
```

`mpu_filter_drw=1` (drawing)

`mpu_filter_inst=1` (Instance)

`mpu_filter_curmod=1` (current model)

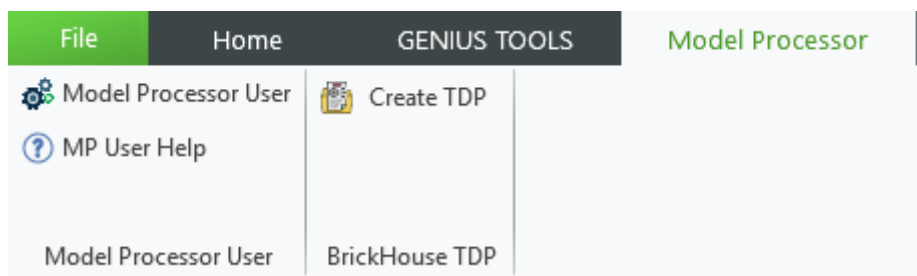
Example: `mpu_start_mpx_after_open_conf=11001` means that the tasks in the task list are only executed for components, assemblies and current models.

The order of the model types can also be found in the comments in the configuration file.

2.4 Configure ribbon menu

Task lists can be added as buttons to the ribbon menu in Creo Parametric in two steps:

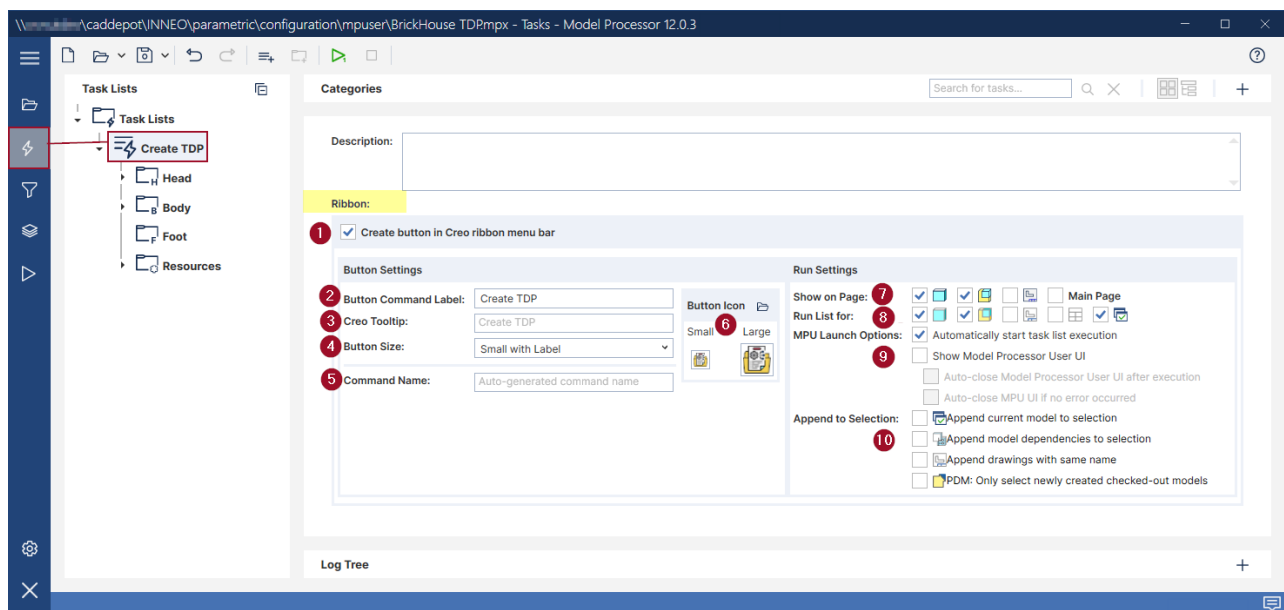
- Settings in Model Processor
- Settings in MPU configuration file



Creo ribbon menu with the Create TDP task list

Settings in Model Processor

In Model Processor, under the Projects menu item, check the box next to *Create button in Creo ribbon menu bar*. Fill out the dialog page as described in the program help.



Settings in MPU configuration file

In the MPU configuration file *mpu_main.cfg*, specify which MPX files should be searched for activated action lists. If no filter is defined, all MPX files are automatically added as buttons in the Creo ribbon menu.

Adjust this configuration for filtering MPX files:

mpx_button_filter=^mpuser_example_.*\.mpx\$

The filter must be a valid regular expression that matches the file name of each MPX file without the path, e. g. *^(file_1.mpx|file_2.mpx)\.mpx\$*

Please note: The configuration option *mpu_disable_generating_text_folder*=0 must be set to 0.
