



Documentation

**INNEO**  
That's IT.



GENIUS TOOLS  
**Model Processor**

Tutorial 03  
Item

**GENIUS TOOLS**

INNEO Solutions GmbH  
Rindelbacher Straße 42  
73479 Ellwangen

[www.inneo.com](http://www.inneo.com)  
[inneo@inneo.com](mailto:inneo@inneo.com)  
Free \* Infoline 0800/7263742 (\*within Germany)

Phone +49 (0) 7961 890-0  
Fax +49 (0) 7961 890-177



# Tutorial 03 – Parameters

*You can only use the tutorial after you have installed the Model Processor in addition to a supported Creo / Pro/ENGINEER version. More detailed installation instructions can be found in the document “Tutorial 00 - Model Processor Installation”. Please install the Model Processor before you start this tutorial. When we talk about Creo in the text below, we also mean any previous version of the program that was published by PTC. The individual tutorials are based on each other.*

*This tutorial can be carried out with any user status.*

## Table of Contents

Tutorial 03 – Parameters .....	1
Table of Contents .....	2
Thematic Introduction .....	2
Copyright / Trademarks .....	2
Typical sequence of a model rework .....	3
Model filter .....	3
Standard filters .....	3
Extended filters .....	4
Variables .....	5
Exercise .....	7
Task .....	7
Initial situation .....	7
Step 01 – List analysis .....	8
Step 02 – Action definition .....	11
Step 03 – Tests .....	14
Step 04 – Model rework .....	15

## Copyright / Trademarks

Copyright 2013 by INNEO Solutions GmbH (Rindelbacher Straße 42 D-73479 Ellwangen)

This documentation is an INNEO Solutions product. All rights reserved.

This document may not be copied, photocopied, reproduced, translated, communicated or converted into an electronic or machine-readable form in whole or in part without the prior written consent of an authorized representative of INNEO Solutions.

The unauthorized use of the documentation can lead to a claim for liquidated damages or legal prosecution.

INNEO Solutions does not accept any liability for possible faulty information in this documentation, and the consequences resulting from such.

### Note concerning registered trademarks

Most of the names of software, hardware and brands are registered trademarks of the individual manufacturers.

### Registered trademarks of Parametric Technology Corporation or its subsidiaries

Behavioral Modeling, Expert Machinist, Flexible Engineering, MECHANICA, Parametric Technology, Parametric Technology Corporation, PHOTORENDER, Pro/DESKTOP, Pro/E, Pro/INTRALINK, Pro/MECHANICA, Pro/TOOLKIT, PTC, PDMLink, Creo, Creo Parametric, Creo View Shaping Innovation, and Windchill.



## Thematic Introduction

### Typical sequence of a model rework



Figure 1 - Sequence of a model rework

#### Why are these individual steps necessary?

- Many models are modified inside a short time.
- Frequently, the effect of a modification is not immediately obvious.  
**Example:** Changing the material can lead to invalid relationships.  
(MATERIAL=MATERIAL\_PARAM("CONDITION"))
- Individual actions offer complex definitions.  
**Example:** "NO" shall be inserted at the 7th digit of the "CLASSIFICATION" parameter of all models that contain a "PARTTYPE" parameter with the value "\*NORM".

### Model filter

Filters can be defined for most actions of the Model Processor. Here, we distinguish between the four standard filters (2, 3, 4 and 5) and the extended filters (6).

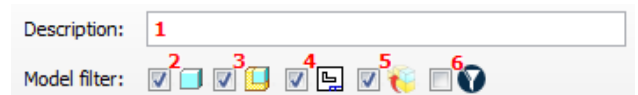


Figure 2 – Filters

#### Standard filters

The filters that are directly related to file types are “standard filters”. These are:

- Parts (2): The selected model is reworked if it is a part.
- Assemblies (3): The selected model is reworked if it is an assembly.
- Drawings (4): The selected model is reworked if it is a drawing.
- Import geometry (5): The selected model is reworked if it is not a Creo model.



## Extended filters

Like for any other action, you can specify a description for each filter. The description does not have any effect on the behavior of the filter. The description is shown in the action list to give a better orientation of the logic contents of the action. It is thus only of use to the user.

If the checkbox of the extended filter (Figure 2 – Filters (6)) is active, the extended filter is used for the decision about the execution of the action. The standard filters are continued to be evaluated. An extended filter can be set globally, or per action that is to be carried out. Clicking on the filter icon opens a menu that permits filters to be added.

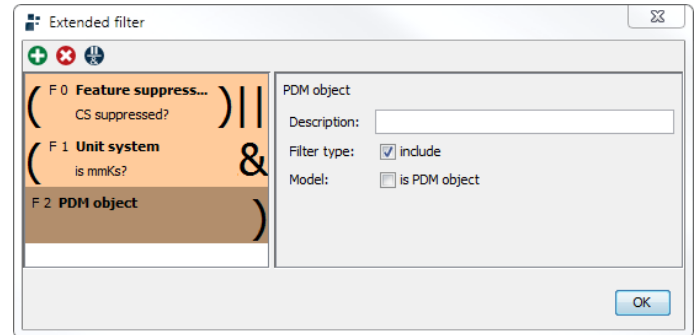


Figure 3 - Extended model filter

The plus **+** icon can be used for adding filters to the list. The minus **-** icon can be used for removing selected actions from the list. All filters in the list can be moved with “Drag and Drop” within the filter list. To add a new filter, use the same procedure as for adding an action in an action list.

There are two possibilities of interconnecting filters.

In an AND operation with two or more filters (shown by an & at the end of the action representation), the filters are combined in the following way:

&	Filter 1 = true	Filter 1 = false
Filter 2 = true	true	false
Filter 2 = false	false	false

In an OR operation with two or more filters (shown by || at the end of the action representation), the filters are combined in the following way:

	Filter 1 = true	Filter 1 = false
Filter 2 = true	true	true
Filter 2 = false	true	false

Use the button “Other operation” to toggle between AND and OR operations.



## Variables

Using variables permits automated value adjustment in many actions of the Model Processor. A value with an entered variable automatically adjusts to the environment.

Not every action supports every variable. The supported variables are listed in each action.

### **@date@**

Outputs the current date in German format.

Example:

12.11.2011

### **@mdl@**

Output of the current model name.

### **@mdlpath@**

Output of the current directory of the current model.

### **@number@**

Output of a consecutively increasing number. This number is incremented every time @number@ is invoked. Here, distinction between actions is not made. Thus, the number is also incremented when @number@ is used in different actions.

### **@para@**

Outputs the value of the current parameter.

### **%PARAM%**

Outputs the value of another parameter. Here, the input between the % characters can vary.

Example:

%DESIGNATION%



### **§env-var§**

Output of an environment variable of the Model Processor. These variables are directly influenced by the environment variables that are specified by the operating system.

Example:

%USERNAME%

%COMPUTERNAME%

%HOMEDRIVE%

%LOGONSERVER%

%USERDOMAIN%

### **\$\$sub\$x\$y\$TEXT\$\$**

Cuts from a text (TEXT) an entry that begins at x and ends at y. x and y are numerical values. You must ensure that the text provides the required number of characters. Count starts at 0 for the first character.

Example:

\$\$sub\$2\$5\$@para@\$

### **\$\$pre\$x\$TEXT\$\$**

Substring from start (character at position 0) to character x. x is a numerical value.

### **\$\$pree\$x\$TEXT\$\$**

Substring from start (character at position 0) up to x characters before the end of the string.

### **\$\$poste\$x\$TEXT\$\$**

Substring of the length x, measured from the end of the string. Corresponds to the last x characters of the string.

### **\$\$posts\$x\$TEXT\$\$**

Substring up to the end, beginning at character x. Corresponds therefore to a variable number of output characters.

### **\$\$repl\$A\$B\$TEXT\$\$**

Replace A with B in TEXT. Using RegExp in input A is possible here.

Example:

\$\$repl\$.\$-\$@para@\$

Other possible variable declarations can be found in the help at Model Processor → Action list → Variables.
---



## Exercise

### Task

All parts and assemblies should have the same parameters. Selected parameters should be identified so that their values can be transferred to Windchill PDMLink. The reason of the standardization is the utilization of the parameter values in bills of material, drawing headers, and for a search in Windchill PDMLink. Typical situations for this task are:

- PDMLink implementation (processing existing models).
- Models from suppliers or customers with different parameter records.
- Library parts or standardized parts from the WEB or from other sources (including STEP...).
- Base models and base parameter records that change in the course of time.

To ensure during the action definition that the required results are obtained, you should perform an analysis before each rework. This example shall find out the parameters that are contained, and whether these parameters have values assigned that can further be used.

### Initial situation

The models that are to be reworked are in a file structure.

The following parameter record shall be in all models after the rework:

Name	Type	Standard value (if not available)	Designated for PDM	Description
EDITOR	String	-	No	
EDIT_DAT	String	-	No	
DESIGNATION	String	-	Yes	Name of the part, e.g. retaining plate
NAME	String	-	Yes	Name in German
SEMIFINISHED PRODUCT	String	-	Yes	
IDNO	String	-	Yes	ERP number
CATEGORY	INTEGER	-1	No	0...assemblies 1...parts (for sorting STK)
MATERIAL	String	-	Yes	
NAMING	String	-	No	Name in English
INSPECTOR	String	-	No	
INSPEC_DAT	String	-	No	
DRAWING NUMBER	String	-	Yes	



To create a suitable parameter record for a rework, performing a list report (see also Step 01 – List analysis) first is useful to include information and suggestions from existing models. Since many models are reworked and, consequently saved inside a short time (version increment), you should really think carefully about each rework.

## Step 01 – List analysis

1. Create a **new project** (as described in Tutorial 01).
2. Create an **action list** of the name "List analysis".
3. In the action list, create an action "**Write list report**" from the "**Report**" action group.
4. At "**File**", enter "@project\_dir@Auflistungsreport.csv".

Figure 4 – List analysis

The list report must be created in the model area. The checkbox for "If existing" should be activated for "Append". If it is set to "Overwrite", only the model edited last will be displayed.





5. Select the data source **“Batch mode”**.

The list analysis is performed for the selected data base. Due to the possible parts and the individual items, we do not want to use the “All in session” mode. With this mode, you must load all parts in the session, and ensure that all drawings are in the session too. Batch processing mode is easier here.

6. Use the **“Configure the batch list”** icon to open batch processing configuration.

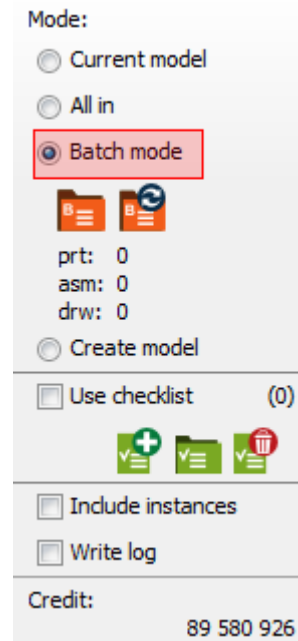




Figure 5 - Data base

7. Using the icon , add a new list with data, that shall be processed, to this user interface. Using the icon , add the folder with the files, that are to be processed, to the list.

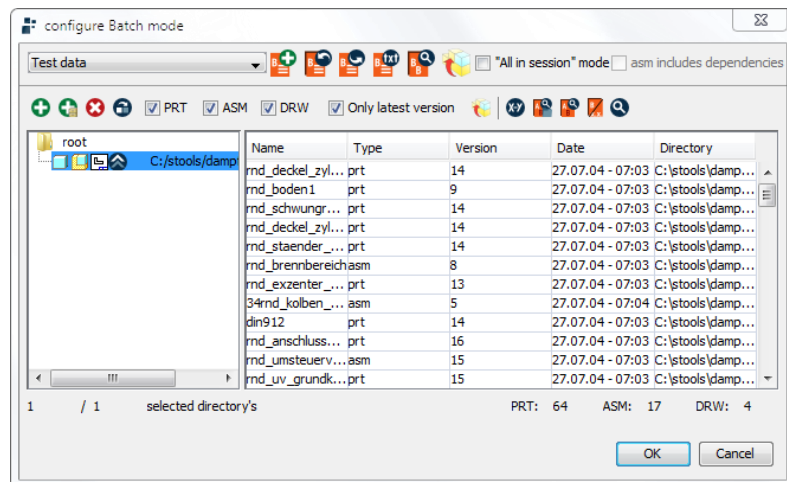



Figure 6 - Batch processing

After a folder has been added in the configuration user interface, you can select here further settings - such as filters - PRT, ASM, DRW and OHV (only highest version). This defines the data that shall be used from the added folder. With the right mouse button you can additionally deactivate individual files explicitly in the folder. These files are displayed in red after deactivation. If you need further information, please check the help that opens when you press F1 in the Model Processor.

8. Click on the **“OK”** button to exit the configuration window. Start the run.
9. Using a suitable program after the run, you can view the result in the Project Explorer  in the “Reports” category.



If you use UTF 8 characters, ensure that the report is set to UTF 8 and that the viewing / editing program opens the file in a UTF-8-compatible way.

*In Microsoft Excel, there is no UTF.8 support when \*.csv files are opened in the default configuration. See also example and explanation in Tutorial 02.*

	A	B	C	D	E	F	G
1	Filename	Parameter	Type	Value	Description	Is designated	Is relation driven
2	rnd_deckel_zyl_u_2000.prt	BENENNUNG	string	unt.Zylinderdeckel		true	false
3	rnd_deckel_zyl_u_2000.prt	BEZEICHNUNG	string	DEC-12-Ms		true	false
4	rnd_deckel_zyl_u_2000.prt	MATERIAL	string	Ms		false	false
5	rnd_deckel_zyl_u_2000.prt	GEWICHT	double	0		true	true
6	rnd_deckel_zyl_u_2000.prt	HALBZEUG	string	-		true	false
7	rnd_deckel_zyl_u_2000.prt	ZEICHNUNGSNUMMER	string	-		true	false
8	rnd_deckel_zyl_u_2000.prt	KATEGORIE	integer	1		false	false
9	rnd_deckel_zyl_u_2000.prt	KOLBENDURCHMESSER	double	12		false	false
10	rnd_deckel_zyl_u_2000.prt	FIRMA	string	-		true	false
11	rnd_deckel_zyl_u_2000.prt	BEARBEITER	string	-		true	false
12	rnd_deckel_zyl_u_2000.prt	PRUEFER	string	-		true	false
13	rnd_deckel_zyl_u_2000.prt	BEARB_DAT	string	-		true	false
14	rnd_deckel_zyl_u_2000.prt	PRUEF_DAT	string	-		true	false
15	rnd_deckel_zyl_u_2000.prt	MASSE	double	0		false	false
16	rnd_boden1.prt	BENENNUNG	string	-		true	false
17	rnd_boden1.prt	BEZEICHNUNG	string	-		true	false
18	rnd_boden1.prt	MATERIAL	string	---		false	false
19	rnd_boden1.prt	GEWICHT	double	0		true	false

Figure 7 – List report

Using the on-board tools of our viewing program, we can find out the following simplified facts:

- Some parts already have a parameter NAME that contains values which correspond to the ones of our future parameter DESIGNATION.
- Some parts already have some of the other parameters. They should neither be overwritten nor deleted.
- Some parameters are identified. This identification must be set exactly to the parameters that are defined in the parameter model.



## Step 02 – Action definition

1. Create a new action list (as described in Tutorial 01).
2. Create an action “**Change all designation flags**” from the “**Model – Parameters**” group. With this action, you don't to change the values. However, you should enter a description to make the action easily recognizable.
3. Next, add an action “**Add parameter**” from the “**Model – Parameters**” action group. Using the green plus **+**, you can add parameters to the list. All settings of the action can be found in “Figure 8 - Adding parameters”. Ensure that the action definition “**overwrite designated at existing**” is activated.

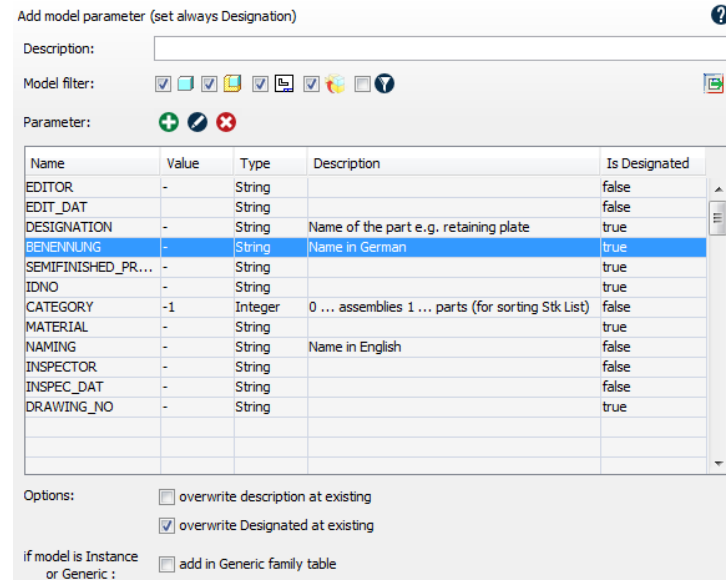


Figure 8 - Adding parameters

Note: Model Processor creates a parameter only if it has not yet been created at the model. A CATEGORY parameter, for example, which exists at the model, retains its value and is not reset to 0.

4. The CATEGORY parameter shall - if it is “-1” - obtain different values for assemblies and parts. For assemblies, the parameter shall be filled with 0. For parts, it shall be filled with 1. The parameter shall not be changed if it contains any other number. Create an action “**Edit parameters**” and establish the simple filters (ASM, PRT, DRW) “**Only for assemblies**”. Also observe “Figure 9 - Editing parameters”.

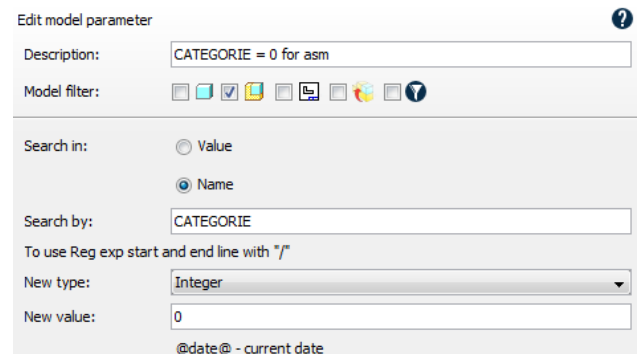


Figure 9 - Editing parameters

5. To carry out the action only when the CATEGORY parameter contains really “-1”, you must add an extended filter for parameters via the **filter icon** . The menu for extended filters works here like any other list menu. The type of the extended parameter filter shall be including.



6. Fill the filter with the following values:

**Note: Filter type:**

Include – the action is performed when the filter is true. In our example: If parameter CATEGORY = -1.

Exclude – the action is performed when the filter is false. In our example: If parameter CATEGORY < -1 or CATEGORY > -1.

7. After the extended filter has been created, add the action to your action list. It should now look like in “Figure 11 - Action list with 3 actions”.
8. You need another action, such as the currently created B2. To copy action B2, select it with the left mouse button. To copy action B2, select it with the left mouse button. Using the right mouse button, select “Copy” from the menu. Next, select “Paste after” from the menu. After you have selected the new action with the left mouse button, you can edit it and modify it for parts.

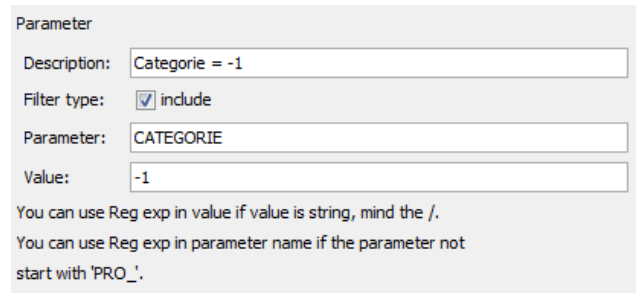


Figure 10 - Extended filter CATEGORY

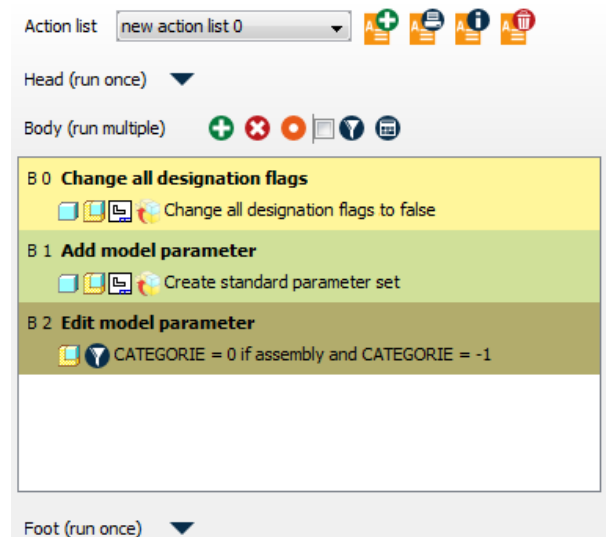


Figure 11 - Action list with 3 actions

9. In the newly created action B3, change the filter from “Assemblies” to “Parts”, and the value from “0” to “1”. Ensure that you also adjust the description if it contains value-specific information.
10. Create another “Edit parameters” action to fill the parameter DESIGNATION with the content of NAME. Here, we use %NAME% as new value. It uses the value from NAME as the new value of DESIGNATION.
11. Create two extended filters for the newly created action. Both filters should be of the parameter type and correspond to the values specified in “Figure 12 - Enhanced filter DESIGNATION”.

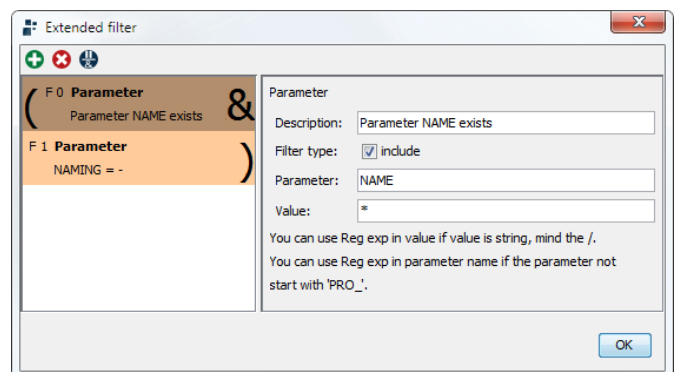


Figure 12 - Enhanced filter DESIGNATION



12. Next, the parameter NAME is no longer required, and is therefore deleted. To do this, create an action “Delete parameters” with the content NAME. Do not use any wildcards to avoid deleting more parameters than necessary.

After this action has been created, the list is complete and you can start with the next step.

The screenshot shows the configuration of an action list. At the top, the 'Action list' is set to 'new action list 0'. Below this, there are icons for adding, deleting, and saving actions. The 'Head (run once)' section is currently empty. The 'Body (run multiple)' section contains five actions:

- B 0 Change all designation flags**: Change all designation flags to false
- B 1 Add model parameter**: Creates standard parameter set
- B 2 Edit model parameter**: CATEGORIE = 0 if asm and CATEGORIE = -1
- B 3 Edit model parameter**: CATEGORIE = 1 if part and CATEGORIE = -1
- B 4 Edit model parameter**: NAMING = NAME
- B 5 Delete model parameter**: NAME

The 'Foot (run once)' section is also empty.

Figure 13 - Entire action list



## Step 03 – Tests

After you have defined the actions that are to be processed, you **must** now test the models that are to be reworked.

These tests should first be performed with the current model and then (if possible) in batch processing mode with all models that are to be processed. Since we do not save the models after rework, the models proper will not be modified.

To check whether our processing worked, you should create a report that lists the interesting parameters. The creation of a CSV report is described in “Tutorial 02 – Simple reports”.

The tests **must** be performed thoroughly to avoid any corruption and thus modification of data during overhaul.

Action list: new action list 0

Head (run once)

- H 0 **Open CSV report**  
open Report

Body (run multiple)

- B 0 **Report parameters**  
Report all Parameters
- B 1 **Change all designation flags**  
Change all designation flags to false
- B 2 **Add model parameter**  
Creates standard parameter set
- B 3 **Edit model parameter**  
CATEGORIE = 0 if asm and CATEGORIE = -1
- B 4 **Edit model parameter**  
CATEGORIE = 1 if part and CATEGORIE = -1
- B 5 **Edit model parameter**  
NAMING = NAME
- B 6 **Delete model parameter**  
NAME
- B 7 **Report parameters**  
Report all Parameters

Foot (run once)

Figure 14 – Entire action list with reports



## Step 04 – Model rework

Once you have successfully completed all tests with positive results, you can start overhaul.

To do this, deactivate or delete the report functions, and add a **“Save model”** action group right at the end of the action list. Use the fundamental philosophy that we do not save faulty data. Since the errors can later be retrieved from the error log, correcting any unexpected errors is not a problem.

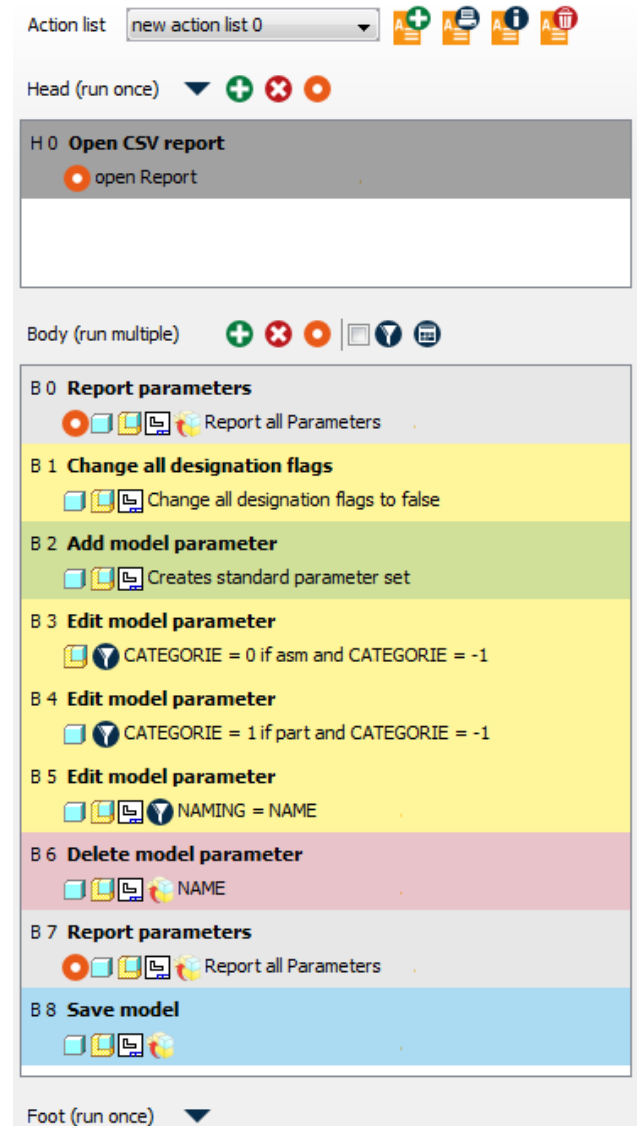


Figure 15 - Entire action list including saving