

# Post Configurator Enablement Training Advanced Modifications

## Topics

### Topic

#### Tcl Modifications – Advanced

- What is Tcl
- Configuration objects/ Propertyts
- Debugging the Postprocess
- Extend existing MOM-events
- Adding a new Level to the Sourcing
- Changing the Access Level of an Object
- Use the Buffer Output -> ToDo

#### Q&A

## What is Tcl

The name Tcl is derived from "Tool Command Language" and is pronounced "tickle". Tcl is a radically simple open-source interpreted programming language that provides common facilities such as variables, procedures, and control structures as well as many useful features that are not found in any other major language.

You'll find all kind of manuals about TCL on the web. One possible starting point could be :

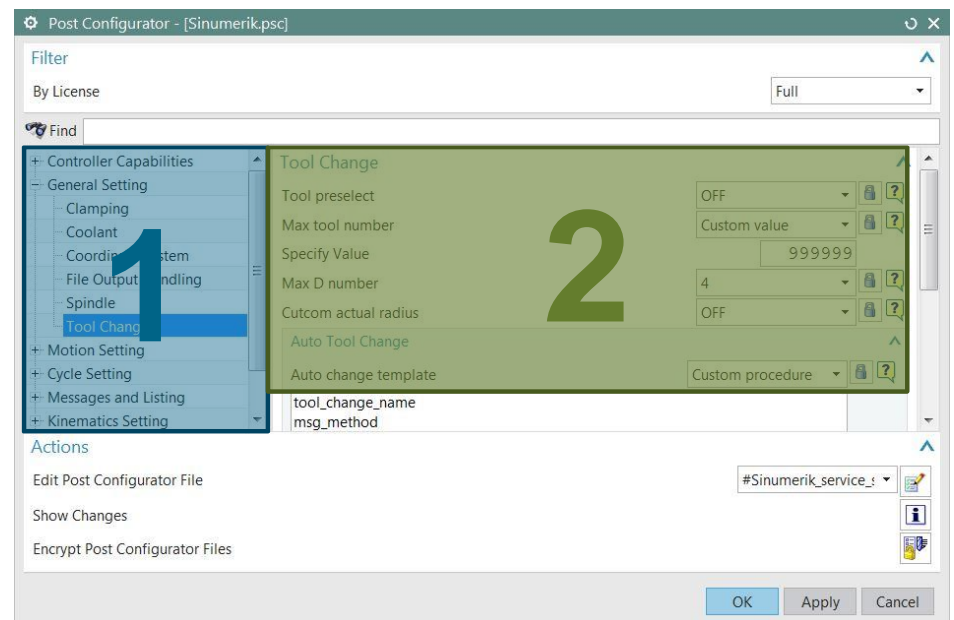
[http://en.wikibooks.org/wiki/Tcl\\_Programming/Introduction](http://en.wikibooks.org/wiki/Tcl_Programming/Introduction)

- NX Postprocessors are TCL based (MOM environemnt is a TCL environnement)
- PostConfigurator post processors are based on the existing MOM architecture

# Configuration Objects / Properties

1. Configuration Objects (Property groups)

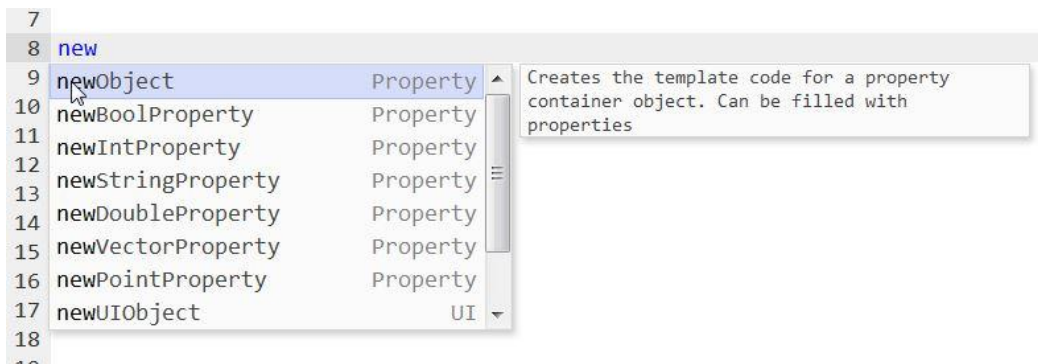
2. Configuration Property



## Define a Configuration Object

➤ use Intellisense from Tcl-Editor for fast creating of new objects

```
7
8 new
9 newObject      Property
10 newBoolProperty Property
11 newIntProperty Property
12 newStringProperty Property
13 newDoubleProperty Property
14 newVectorProperty Property
15 newPointProperty Property
16 newUIObject    UI
17
18
```



➤ rename the object and the UI-name/ tooltip

```
8 LIB_GE_CREATE_obj CONF_CUSTOM_tool_dimension {} {
9     LIB_GE_property_ui_name      "Define Tool Dimension output"
10    LIB_GE_property_ui_tooltip    "Define if and how Tool Dimension should be output"
11
12
13 }
```

## Define a Property within an Object 1/4

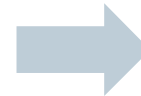
- Create a new INT- property within the Tcl-editor

```

26
27 new
28 newObject      Property
29 newBoolProperty Property
30 newIntProperty  Property
31 newStringProperty Property
32 newDoubleProperty Property
33 newVectorProperty Property
34 newPointProperty Property
35 newUIObject     UI
36
37
38

```

Creates the template code for an integer property.



```

27 set id "IntProperty"
28   set $id 0
29   set options($id)      {*VALUE*}
30   set datatype($id)     "INT"
31   set access($id)       222
32   set dialog($id)       {{Int Property}}
33   set descr($id)        {{A numeric Property}}
34   set ui_parent($id)    "GroupName"
35
36 }

```

- define, rename and set access level
- optional define a DropDown menu for the UI

```

27 set id "output_tool_dimension"
28   set $id 0
29   set options($id)      {NO|YES}
30   set options_ids($id)  {0|1}
31   set datatype($id)     "INT"
32   set access($id)       222
33   set dialog($id)       {{Output Tool Dimension}}
34   set descr($id)        {{Turn the output of the tool dimension on or off}}
35   set ui_parent($id)    "???"
36
37 }

```

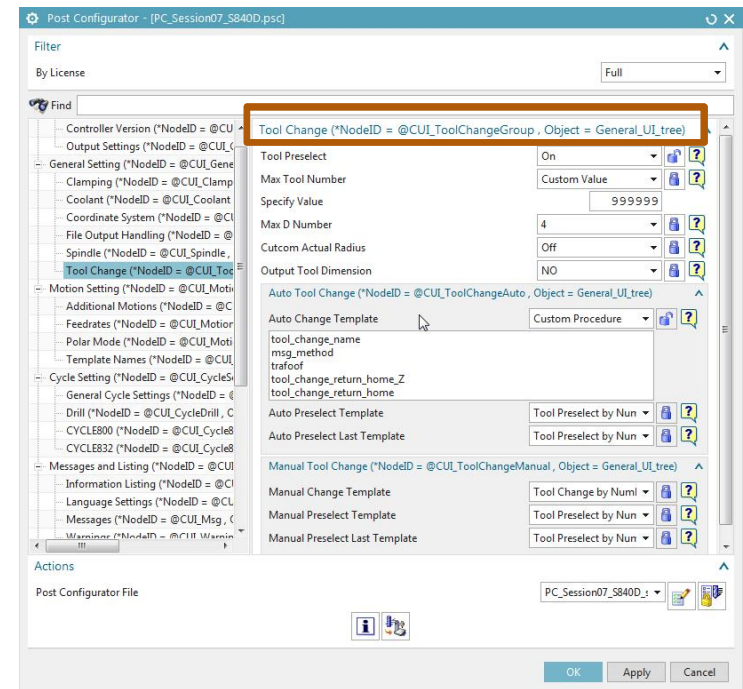
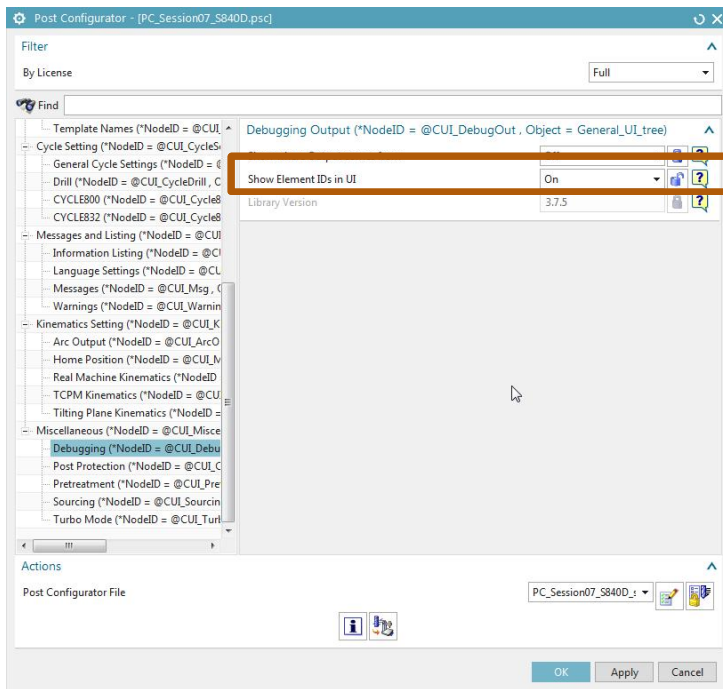
## Define a Property within an Object 2/4

TCL Code	Description
<code>set id "output_tool_dimension"</code>	<b>Unique identifier of property</b>
<code>set \$id 0</code>	<b>Default value</b>
<code>set options(\$id) {NO YES}</code>	<b>Option list to display in DropDown menu</b>
<code>set option_ids(\$id) {0 1}</code>	<b>Return values assigned to menu options</b>
<code>set datatype(\$id) INT</code>	<b>Datatype of the property</b>
<code>set access(\$id) 222</code>	<b>Define access rights for property</b>
<code>set dialog(\$id) {{Output Tool Dimension}}</code>	<b>Property name shown in UI</b>
<code>set descr(\$id) {{Turn the output of the tool dimension on or off}}</code>	<b>ToolTip for UI</b>
<code>set ui_parent(\$id)</code>	<b>Define in which group this property will be shown</b>

**Optional, Only needed for DropDown menu**

## Define a Property within an Object 3/4

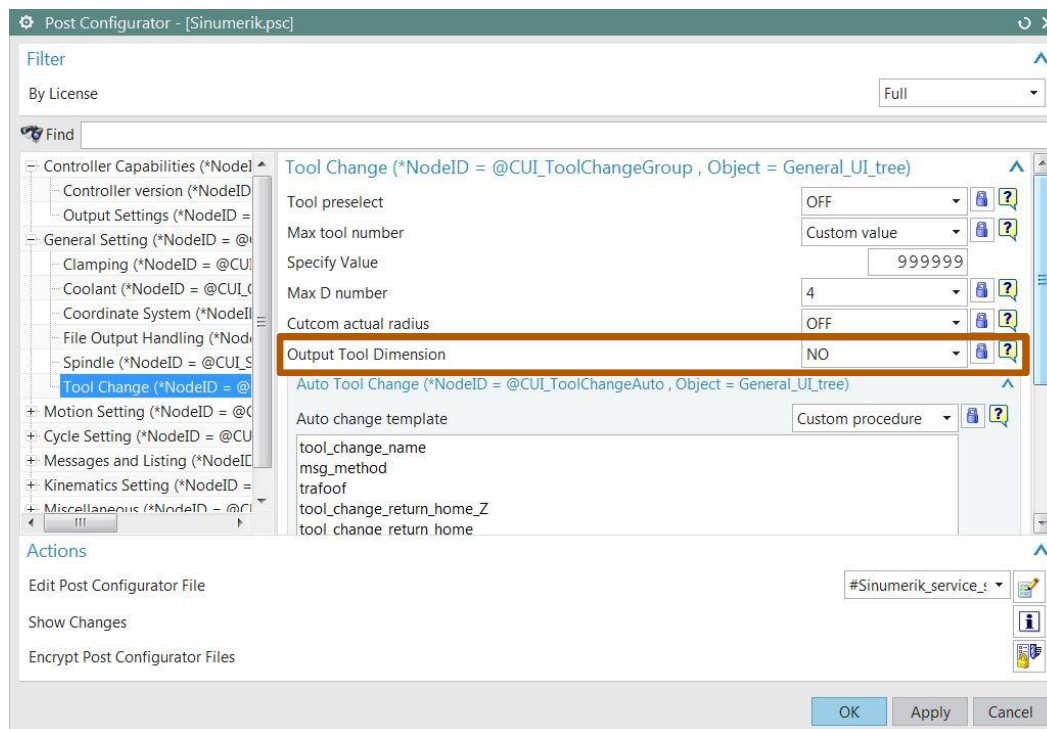
- to put the property in an existing group turn the Show element ID's ON
- all Node-ID's and Group-ID's are shown in the UI





## Define a Property within an Object 4/4

- Set the `ui_parent` for the property to get it in the group



```

27 set id "output_tool_dimension"
28   set $id 0
29   set options($id)      {NO|YES}
30   set options_ids($id)  {0|1}
31   set datatype($id)     "INT"
32   set access($id)       222
33   set dialog($id)       {{Output Tool Dimension}}
34   set descr($id)        {{Turn the output of the tool dimension on or off}}
35   set ui_parent($id)    "@CUI_ToolChangeGroup"
36
37 }
38

```

## Fully defined Configuration Object

```
LIB_GE_CREATE_obj CONF_CUSTOM_tool_dimension {} {
```

### ConfigurationObject

```
LIB_GE_property_ui_name "Define Tool Dimension Output"
```

```
LIB_GE_property_ui_tooltip "Define if and how Tool Dimension should be output"
```

```
set id "output_tool_dimension"
```

```
set $id 0
```

```
set options($id) {NO|YES}
```

```
set options_ids($id) {0|1}
```

```
set datatype($id) INT
```

```
set access($id) 222
```

```
set dialog($id) {{Output Tool Dimension}}
```

```
set descr($id) {{Turns the output of the tool dimension on or off}}
```

```
Set ui_parent($id) "@CUI_ToolChangeGroup"
```

### ConfigurationProperty

## How to deal with properties in TCL

### Getting the current value of a property

```
*object_name* *property_name*
```

e.g.

```
set current_value [CONF_CUSTOM_tool_dimension output_tool_dimension]
```

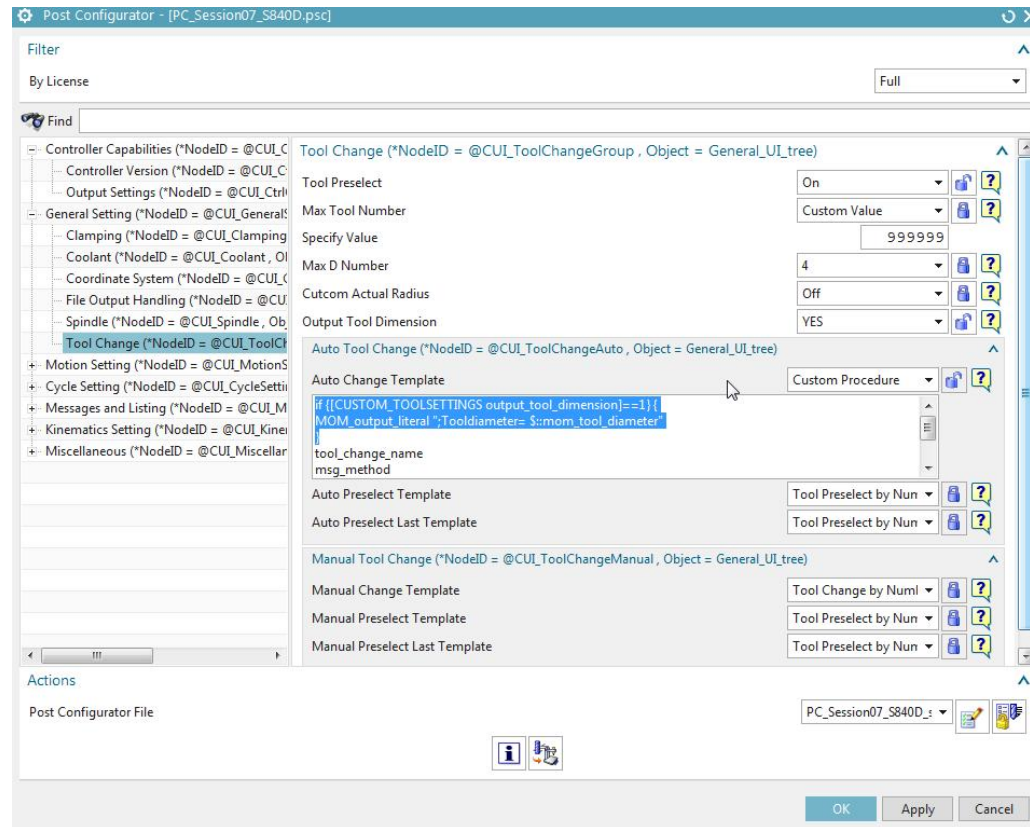
### Changing the value of a property in TCL

```
*object_name* set *property_name* value
```

e.g.

```
CONF_CUSTOM_tool_dimension set output_tool_dimension 1
```

# Output the Tooldiameter depend on Property setting

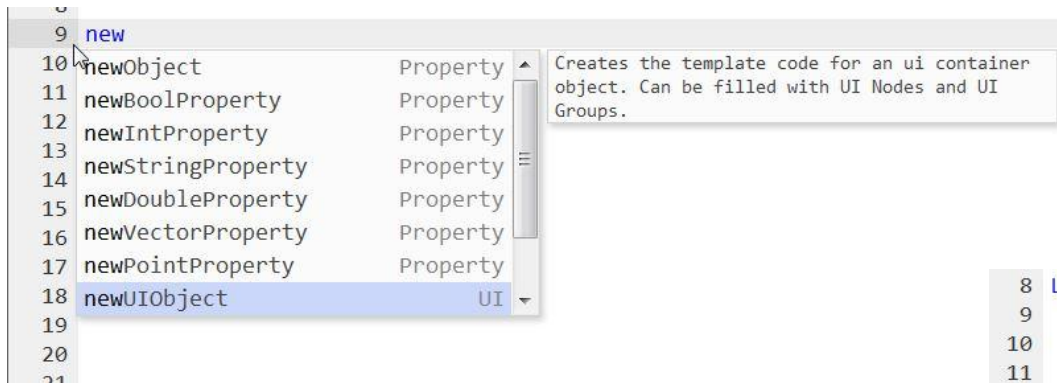


# Customize Grouping 1/3

The screenshot shows the 'Tool Change' configuration window in Siemens software. The left sidebar lists various settings categories, with 'Tool Change' selected. The main area displays parameters for tool change, including 'Tool preselect', 'Max tool number', 'Specify Value', 'Max D number', 'Cutcom actual radius', and 'Output Tool Dimensi'. Below these are 'Auto Tool Change' settings like 'Auto change template' and 'Auto preselect template'. The 'Actions' section at the bottom includes 'Edit Post Configurator File', 'Show Changes', and 'Encrypt Post Configurator Files'. Three green callout boxes with arrows point to specific UI elements: 'LIB\_GE\_property\_ui\_tooltip' points to the window title bar; 'LIB\_GE\_property\_ui\_name (Group)' points to the 'Tool Change' entry in the sidebar; and three boxes labeled '\$id / options(\$id)', 'dialog(\$id)', and 'descr(\$id)' point to dropdown menus and help icons for 'Specify Value', 'Output Tool Dimensi', and 'Tool preselect by nu' respectively.

## Customize Grouping – Create a new group 2/3

- for the UI there are templates, e.g. for Nodes, Groups
- there is a special datatype for groups



- the ui\_parent can set to existing nodes or groups
- for create a new Main group use „root“

```

8 LIB_GE_CREATE_obj MyUser_tree {UI_TREE} {
9   LIB_GE_property_ui_name      "MyCustomTree"
10  LIB_GE_property_ui_tooltip   "Define MyCustomTree"
11
12  set id "Tooldimension_Group"
13    set $id "0"
14    set datatype($id)          "GROUP"
15    set access($id)            222
16    set dialog($id)            {{Tooldimension}}
17    set descr($id)             {{Setting for tool output}}
18    set group_status($id)      1
19    set ui_parent($id)         "@CUI_ToolChange"
20    set ui_sequence($id)       -1
21 }

```

## Customize Grouping – add the property to a new group 3/3

- groups only visible if they contain a property
- property must be created in an own object
- id of the group is the ui\_parent for the property

```
LIB_GE_CREATE_obj MyUser_tree {UI_TREE} {
  LIB_GE_property_ui_name      "MyCustomTree"
  LIB_GE_property_ui_tooltip   "Define MyCustomTree"

  set id "Tooldimension_Group"
  set $id "0"
  set datatype($id)           "GROUP"
  set access($id)             222
  set dialog($id)             {{Tooldimension}}
  set descr($id)              {{Setting for tool output}}
  set group_status($id)       1
  set ui_parent($id)          "@CUI_ToolChange"
  set ui_sequence($id)        -1
}

LIB_GE_CREATE_obj CONF_CUSTOM_tool_group {} {
  LIB_GE_property_ui_name      "Define Tool Dimension output"
  LIB_GE_property_ui_tooltip   "Define if and how Tool Dimension should be output"

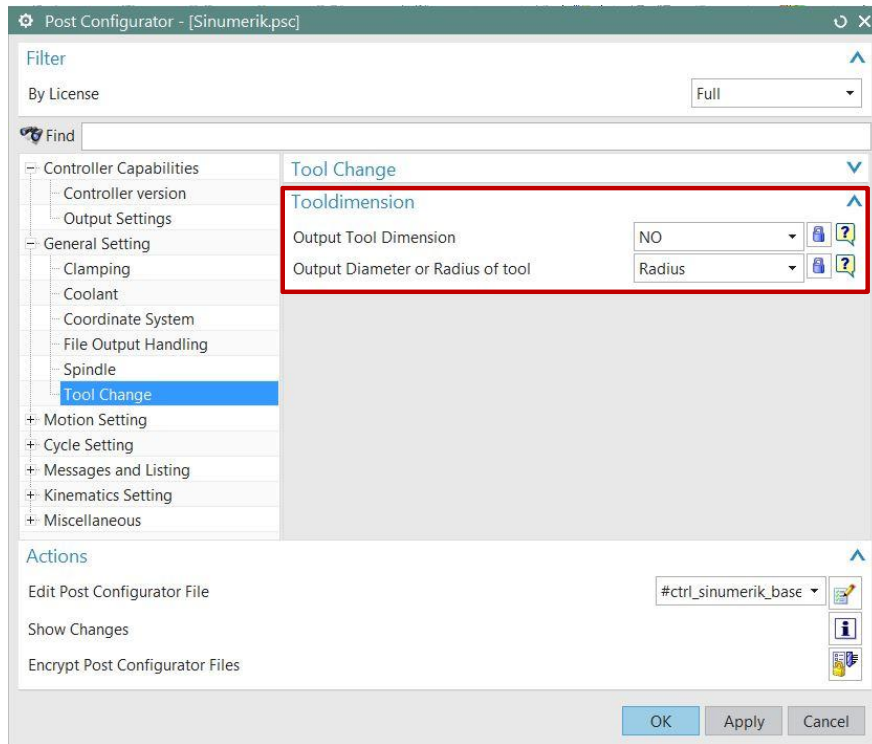
  set id "output_tool_dimension"
  set $id 0
  set options($id)            {NO|YES}
  set options_ids($id)        {0|1}
  set datatype($id)           "INT"
  set access($id)             222
  set dialog($id)             {{Output Tool Dimension}}
  set descr($id)              {{Turn the output of the tool dimension on or off}}
  set ui_parent($id)          "Tooldimension_Group"
}
```

## Property Datatypes

```
set datatype($id) INT
set datatype($id) DOUBLE / REAL
set datatype($id) STRING
set datatype($id) Vector           (Vector of 3 doubles)
set datatype($id) Point
set datatype($id) GROUP           (for propertys)
set datatype($id) NODE           (need for own groups in the tree)
Set datatype($id) COMMANDBLOCK   (for tcl Code or Block templates)
```



## Example – Create 2nd property



```
LIB_GE_CREATE_obj CONF_CUSTOM_tool_dimension{} {
```

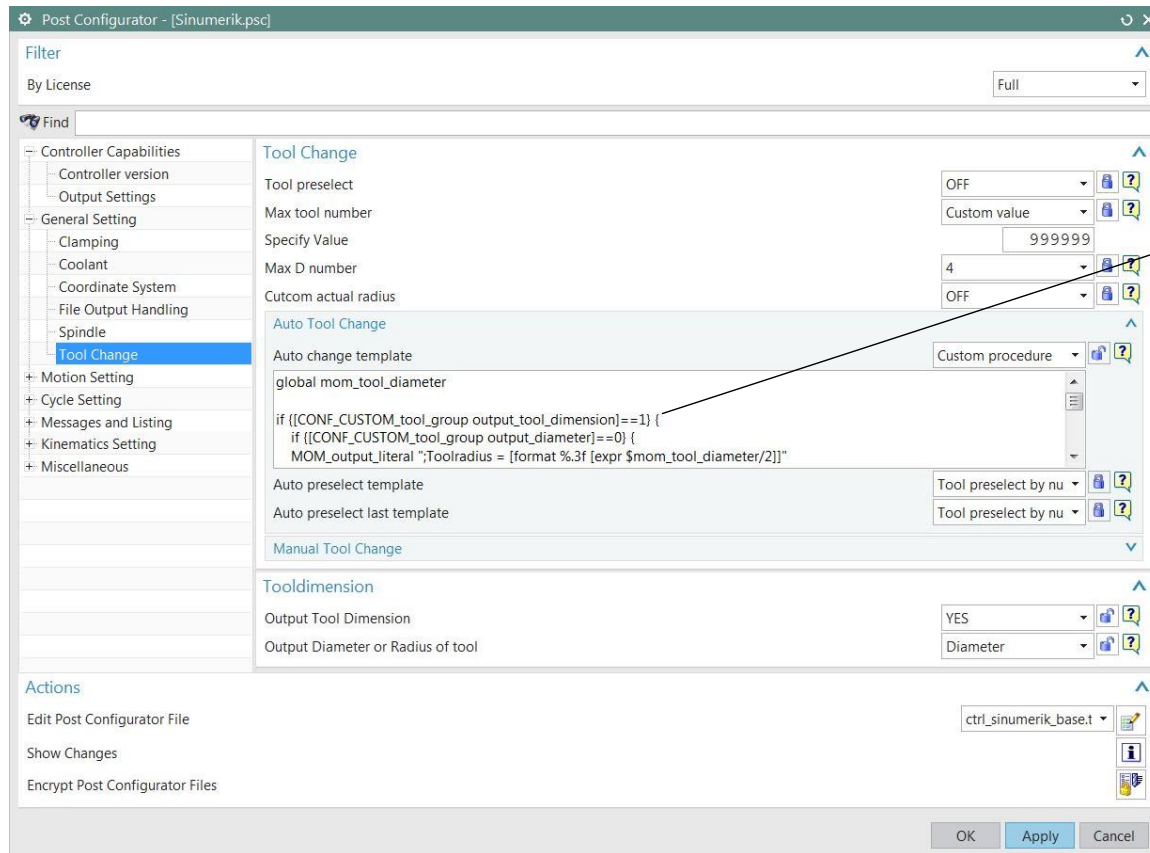
```
LIB_GE_property_ui_name "Define Tool Dimension Output"
LIB_GE_property_ui_tooltip "Define if and how Tool Dimension should be output"
```

```
set id "output_tool_dimension"
set $id 0
set options($id) {NO|YES}
set options_ids($id) {0|1}
set datatype($id) INT
set access($id) 222
set dialog($id) {{Output Tool Dimension}}
set descr($id) {{Turns the output of the tool dimension on or off}}
```

```
set id "output_mode_tool_dimension"
set $id 1
set options($id) {Radius|Diameter}
set options_ids($id) {2|1}
set datatype($id) INT
set access($id) 222
set dialog($id) {{Output as}}
set descr($id) {{Define whether dimension should be output as diameter or radius}}
```

```
}
```

# Option 1: Outputting it with the tool change



```

N14 G153 X0. Y0.
N16 ; ToolDiameter = 40.000
N18 T1 M6
N20 G54

```

## Setting the access level

- Access level is coded by number:



**0**

Hidden



**1**

Read Access



**2**

Read & Write

Access Code	Basic License	Advanced License	Full License
222	Read / Write	Read / Write	Read / Write
122	Read	Read / Write	Read / Write
022	Hidden	Read / Write	Read / Write
012	Hidden	Read	Read / Write
002	Hidden	Hidden	Read / Write
001	Hidden	Hidden	Read
000	Hidden	Hidden	Hidden

`set access($id)`

2

2

2

# Debugging the Post Processor



## Debugging the Post Processor

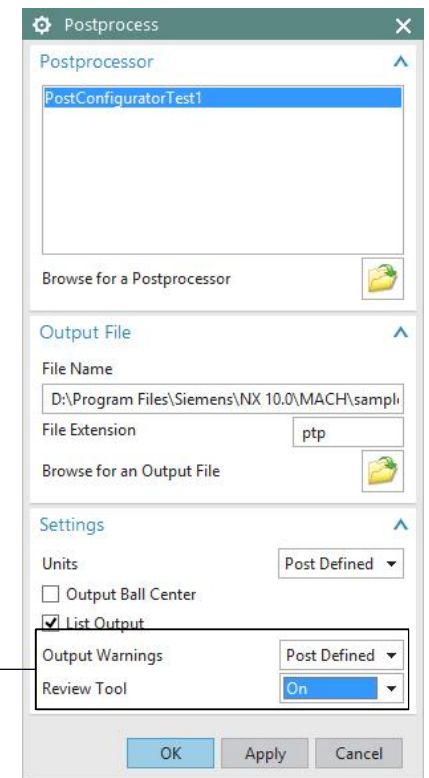


- Standard NX *Review Tool* still works
- Some review output will be suppressed if the Turbo Mode is On
- Overview of MOM Events and MOM Variables
- New debug functionality „*Show where output comes from*“ (UI)
- *Helps to identify what is creating certain output lines*

## NX Review Tool – Turbo Mode ON

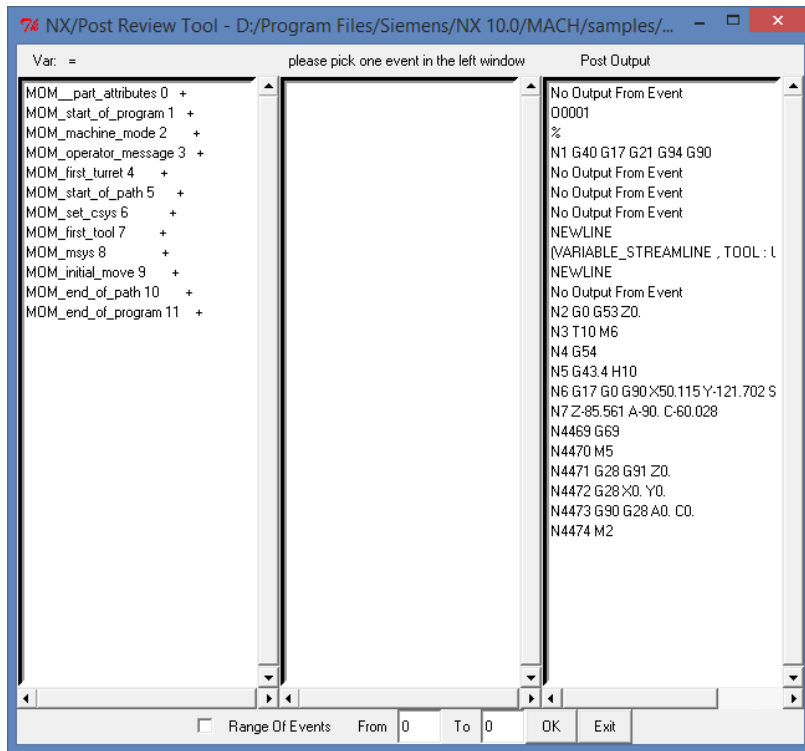
1. Turbo Mode is ON (See Day 02)
2. Post Process the required Operation with Review Tool ON
  
3. Turn Turbo Mode OFF (See Day 02)
4. Post Process the required Operation with Review Tool ON

**Note:** Using the review tool will dramatically increase the post processing time.

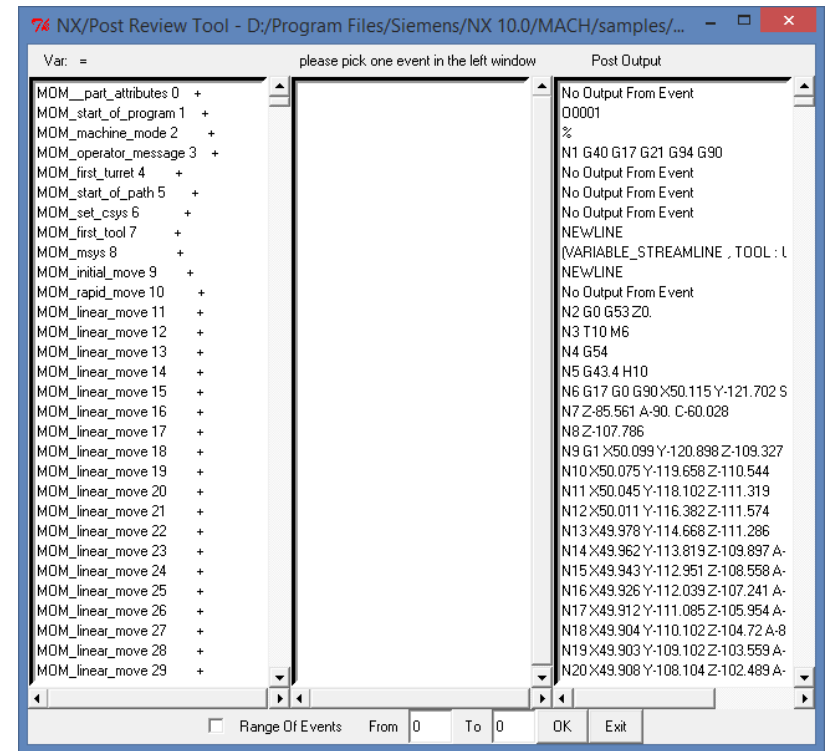


# Using Review Tool with VARIABLE\_STREAMLINE

Turbo Mode ON



Turbo Mode OFF



## Show where output comes from

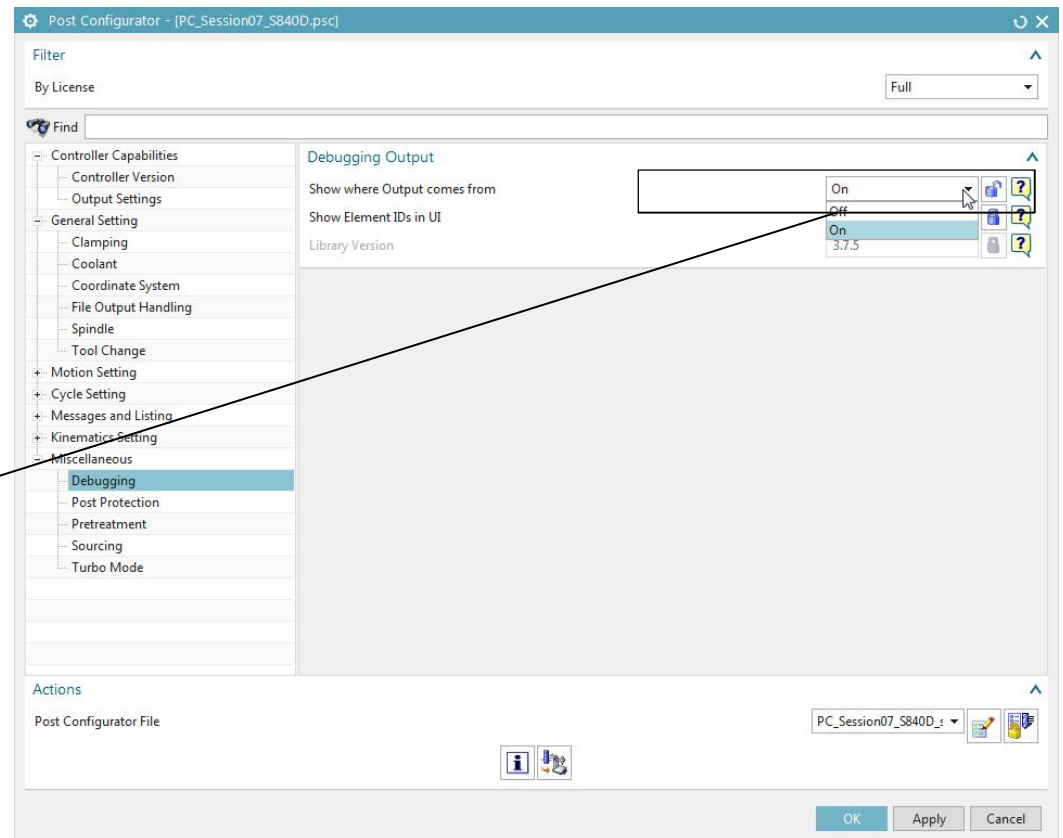


- New functionality for Post Configurator posts to list
- Which MOM Event generated this line
- Which Library function generated that line
- Buffer and Sequence Information of the Line



## Show where output comes from

1. Turn „Show where output comes from“ ON
2. Post Process an Operation



**START BUFFER: Buffer Name**

**TAG\_INFO: Line Tag Name**

```

File Edit Information
O0001
>>===== START BUFFER SEQ : LIB_SFF_program_header_comment HEADER_COMMENT
|
|>===== END BUFFER SEQ :>>LIB_GE_command_buffer_seq LIB_SFF_program_header_comment HEADER_COMMENT (OOTB_program_header)<<
|>===== START BUFFER SEQ : MOM_start_of_program_LIB PROGRAMSTART
N1 G40 G17 G21 G94 G90
|>===== END BUFFER SEQ :>>LIB_GE_command_buffer_seq MOM_start_of_program_LIB PROGRAMSTART (@START_OF_PROGRAM)<<
|>===== START BUFFER SEQ : LIB_SFF_operation_header_comment HEADER_COMMENT
|
(CAVITY_TOP , TOOL : UGT0202_001)
|>===== END BUFFER SEQ :>>LIB_GE_command_buffer_seq LIB_SFF_operation_header_comment HEADER_COMMENT (@NEWLINE1 @INFOLINE @NEWLINE2)<<
N2 GO G53 Z0.
|>===== START BUFFER SEQ : MOM_tool_change_LIB TOOL_CHANGE
N3 T01 M6
|>===== END BUFFER SEQ :>>LIB_GE_command_buffer_seq MOM_tool_change_LIB TOOL_CHANGE (@TOOL_CHANGE)<<
|>===== START BUFFER SEQ : LIB_SFF_first_tool_path_motion MILL,CONSTANT
N4 G55
N5 G68.2 X-6.443 YO. Z-1.726 I-90. J15. K90.
N6 G53.1
|>===== END BUFFER SEQ :>>LIB_GE_command_buffer_seq LIB_SFF_first_tool_path_motion MILL (@MAIN_ORIGIN_CALL @ROTARY_POSITIONNING)<<
N7 G17 G43 GO G90 X35.412 Y31.842 Z27.941 S2228 H1 M3
N8 Z8.866
N9 G1 X1.58 Y10.372 F1203.
N10 G2 X0.274 Y9.558 R97.316
N11 G3 X0.504 Y-9.707 R11.5
N12 G2 X1.568 Y-10.367 R89.939
N13 G1 X35.412 Y-31.659
N14 GO X44.267 Y19.501 Z5.91
N15 G1 X5.554 Y11.5
N16 G2 X4.342 Y11.262 R64.281
N17 G3 X4.406 Y-11.275 R11.5
N18 G2 X5.653 Y-11.51 R48.864
N19 G1 X44.139 Y-19.282
N20 GO X51.606 Y5.439 Z2.955
>>MOM_initial_move , LIB_ROTARY_positionning_first_move_pos , MOM_do_template rapid_move
>>MOM_rapid_move , MOM_rapid_move_LIB , MOM_do_template rapid_move
>>MOM_linear_move , MOM_linear_move_LIB , MOM_do_template linear_move
>>MOM_circular_move , MOM_circular_move_LIB , MOM_do_template circular_move
>>MOM_circular_move , MOM_circular_move_LIB , MOM_do_template circular_move
>>MOM_linear_move , MOM_linear_move_LIB , MOM_do_template linear_move
>>MOM_rapid_move , MOM_rapid_move_LIB , MOM_do_template rapid_move
>>MOM_linear_move , MOM_linear_move_LIB , MOM_do_template linear_move
>>MOM_circular_move , MOM_circular_move_LIB , MOM_do_template circular_move
>>MOM_circular_move , MOM_circular_move_LIB , MOM_do_template circular_move
>>MOM_linear_move , MOM_linear_move_LIB , MOM_do_template linear_move
>>MOM_rapid_move , MOM_rapid_move_LIB , MOM_do_template rapid_move
|
TAG_INFO:OOTB_program_header
TAG_INFO:OOTB_program_header
TAG_INFO:$START_OF_PROGRAM
TAG_INFO:$NEWLINE1
TAG_INFO:$INFOLINE
TAG_INFO:
TAG_INFO:
TAG_INFO:

```

**Complete Buffer Sequence Command**

**MOM Events, MOM Templates**

## Option 2: Using the sequences

### What is the OutputBuffer:

- Within a buffer sequence tags are assigned to NC-lines
- Output can be reordered within the same buffer
- Additional output can be added to the buffer sequence
- Output can be suppressed from buffer sequence
- A template of the command to manipulate the buffer will be displayed if „Show where output comes from“ is activated
- LIB\_GE\_command\_buffer\_seqLIB\_SPF\_operation\_header\_comment HEADER\_COMMENT {@NEWLINE1 @INFOLINE @NEWLINE2}

## Add Tool Information output to operation header buffer

Create a procedure containing the code to output Tool information

```
proc Custom_Tool_Information { } {  
  global mom_tool_diameter  
  
  if {[CONF_CUSTOM_tool_group output_tool_dimension]==1} {  
    if {[CONF_CUSTOM_tool_group output_diameter]==0} {  
      MOM_output_literal ";Toolradius = [format %.3f [expr $mom_tool_diameter/2]]"  
    } else {  
      MOM_output_literal ";Tooldiameter = [format %.3f $mom_tool_diameter]"  
    }  
  }  
}
```

Add this proc to the output buffer

```
LIB_GE_command_buffer_seq MOM_tool_change_LIB TOOL_CHANGE {Custom_Tool_Information @TOOL_CHANGE}
```

# Extending existing MOM Events

## Extending existing MOM Events



- If it possible to react to certain MOM events to modify them, or extend them
- It is possible to execute Tcl code before or after certain MOM events

**Note:** Not every MOM event can be extended (List in the next slides)

## Available MOM Events

- MOM\_auxfun\_ENTRY {args} {}
- MOM\_before\_motion\_ENTRY {args} {}
- MOM\_before\_output\_LIB\_ENTRY {args} {}
- MOM\_bore\_back\_move\_LIB\_ENTRY {args} {}
- MOM\_bore\_drag\_move\_LIB\_ENTRY {args} {}
- MOM\_bore\_dwell\_move\_LIB\_ENTRY {args} {}
- MOM\_bore\_manual\_dwell\_move\_LIB\_ENTRY {args} {}
- MOM\_bore\_manual\_move\_LIB\_ENTRY {args} {}
- MOM\_bore\_move\_LIB\_ENTRY {args} {}
- MOM\_bore\_no\_drag\_move\_LIB\_ENTRY {args} {}
- MOM\_circular\_move\_LIB\_ENTRY {args} {}
- MOM\_coolant\_off\_ENTRY {args} {}
- MOM\_coolant\_on\_ENTRY {args} {}
- MOM\_delay\_ENTRY {args} {}
- MOM\_drill\_break\_chip\_move\_LIB\_ENTRY {args} {}
- MOM\_drill\_deep\_move\_LIB\_ENTRY {args} {}
- MOM\_drill\_dwell\_move\_LIB\_ENTRY {args} {}
- MOM\_drill\_move\_LIB\_ENTRY {args} {}
- MOM\_drill\_text\_move\_LIB\_ENTRY {args} {}
- MOM\_end\_of\_path\_LIB\_ENTRY {args} {}
- MOM\_end\_of\_program\_LIB\_ENTRY {args} {}
- MOM\_end\_of\_subop\_path\_LIB\_ENTRY {args} {}
- MOM\_first\_move\_LIB\_ENTRY {args} {}
- MOM\_first\_tool\_LIB\_ENTRY {args} {}
- MOM\_from\_move\_LIB\_ENTRY {args} {}
- MOM\_gohome\_move\_LIB\_ENTRY {args} {}
- MOM\_helix\_move\_LIB\_ENTRY {args} {}
- MOM\_linear\_move\_LIB\_ENTRY {args} {}
- MOM\_machine\_mode\_LIB\_ENTRY {args} {}
- MOM\_nurbs\_move\_LIB\_ENTRY {args} {}
- MOM\_operator\_message\_ENTRY {args} {}
- MOM\_opstop\_ENTRY {args} {}
- MOM\_prefun\_ENTRY {args} {}
- MOM\_rapid\_move\_LIB\_ENTRY {args} {}
- MOM\_start\_of\_path\_LIB\_ENTRY {args} {}
- MOM\_start\_of\_program\_LIB\_ENTRY {position} {}
- MOM\_start\_of\_subop\_path\_LIB\_ENTRY {args} {}
- MOM\_stop\_ENTRY {args} {}
- MOM\_tap\_move\_LIB\_ENTRY {args} {}
- MOM\_tool\_change\_LIB\_ENTRY {args} {}
- LIB\_RETURN\_move\_LIB\_ENTRY {args} {}
- LIB\_ROTARY\_positioning\_init\_ENTRY {args} {}
- LIB\_ROTARY\_simultaneous\_init\_ENTRY {args} {}
- LIB\_SPF\_KINEMATICS\_set\_plane\_output\_kin\_ENTRY {args} {}
- LIB\_SPF\_KINEMATICS\_set\_simultaneous\_kin\_ENTRY {args} {}
- LIB\_SPF\_KINEMATICS\_set\_turn\_kin\_ENTRY {args} {}
- LIB\_SPF\_default\_initial\_setting\_ENTRY {args} {}
- LIB\_SPF\_polar\_cart\_ENTRY {args} {}
- LIB\_SPF\_spindle\_direction\_ENTRY {args} {}
- LIB\_SPF\_spindle\_max\_min\_ENTRY {args} {}
- LIB\_SPF\_spindle\_set\_ENTRY {args} {}
- LIB\_SPF\_program\_header\_comment\_LIB\_ENTRY {args} {}
- LIB\_SPF\_operation\_header\_comment\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_cut\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_rapid\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_engage\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_retract\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_firstcut\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_approach\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_stepover\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_departure\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_return\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_traversal\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_sidecut\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_from\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_gohome\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_gohome\_default\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_cycle\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_lift\_move\_LIB\_ENTRY {args} {}
- LIB\_CTRL\_undefined\_move\_LIB\_ENTRY {args} {}

## Extending of a MOM Event

### Code Snippet:

```
proc MOM_end_of_program_LIB_ENTRY {position} {
  switch -- $position {
    "start"
    {
      #Your code before the event goes here
    }
    "end"
    {
      #Your code after the event goes here
    }
  }
}
```



# Adding a new Level to the Sourcing

## Adding a new Level to the Sourcing

- The sourcing is handled in the main Post Configurator entry file (file ending .psc)
- .psc file is an XML file
- To add a new layer add a new entry to the sourcing sequence
- Levels will be loaded in the order of the sourcing
- The sourcing will automatically check if there is a Tcl or PCE file (Tcl will be preferred in the case both exist)

```
<Sourcing>
  <Sequence>
    <Filename Name="ctrl_fanuc_base" Processing="true"/>
    <Filename Name="PostConfiguratorTest1_mtb" Processing="true"/>
    <Filename Name="oem_ootb_5ax_fanuc" Processing="true"/>
    <Filename Name="machine_ootb_5ax_fanuc" Processing="true"/>
    <Filename Name=„new_level_file" Processing="true"/>
    <Filename Name="PostConfiguratorTest1_service_fanuc" Processing="true"/>
  </Sequence>
</Sourcing>
```

# Changing the Access Level of an Object



## Changing the Access level for existing Objects

- It is possible to decrease the Access level of a predefined object
- The corresponding and required names can be found in the documentation

**Note:** It is not possible to raise a access level again, it is only possible to lower it.

**Syntax:**

**LIB\_GE\_CONF\_set\_property\_access** *\*Object\_name\** *\*Properties\** *\*\*Access\_Levels\** *\*Option\**

Object_name	The name of the parent object	
Properties	A list of properties the access level need to be lowered. Also possible to change all.	"rotate_before rotate_after" all
Access_Levels	A list of license level for which the access code will be changed.	"BASIC ADVANCED FULL"
Option	The new access option	HIDE READONLY

**Samples:**

**LIB\_GE\_CONF\_set\_property\_access** *CONF\_FANUC\_controller* all "BASIC ADVANCED" HIDE

**LIB\_GE\_CONF\_set\_property\_access** *CONF\_FANUC\_G68* "rotate\_before rotate\_after" BASIC READONLY

