SIEMENS
Ingenuity for life

NX Post Configurator
Tasks and Solutions

Task 1

SIEMENS
Ingenuity for life

Create an additional UDE with Post Configurator for a horizontal milling machine tool to setup W and V axis for Sinumerik 840D:

Conditions:

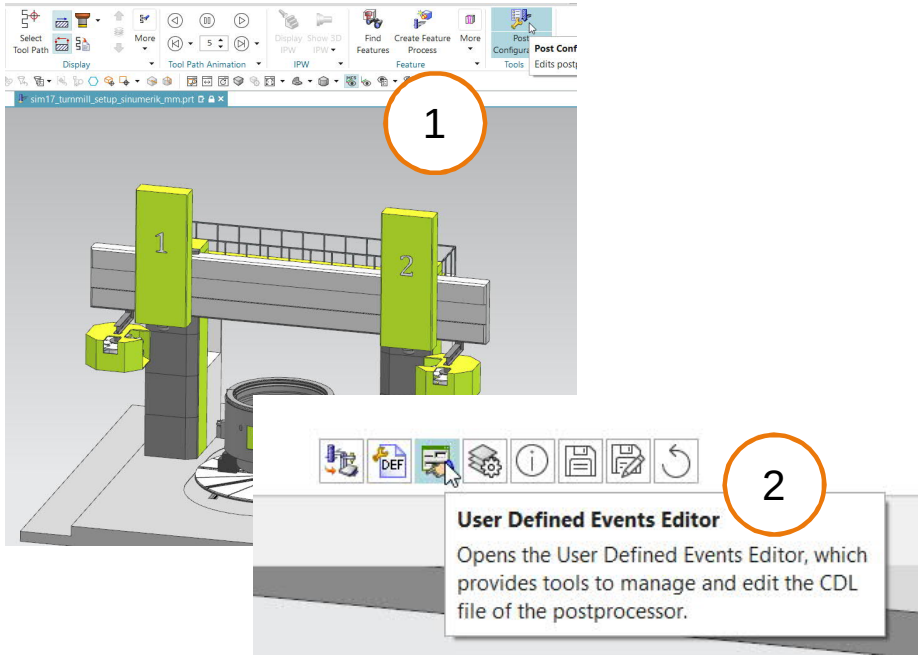
- Possibility to switch between ATRANS and specific OEM Cycle (e.g. L9958, Waldrich Coburg)
- The switch must be done by a property
- The output appears on the right location in NC code
- Contains logic that the values are entered correctly in the UDE

Sample Code Sinumerik standard
ATRANS Z=200 W=-150 V=-50
G0 V0 W0 Z0

Sample Code L9958
ZV=200 WV=-150 VV=-50
L9958()

Task 1
Solution 1/15

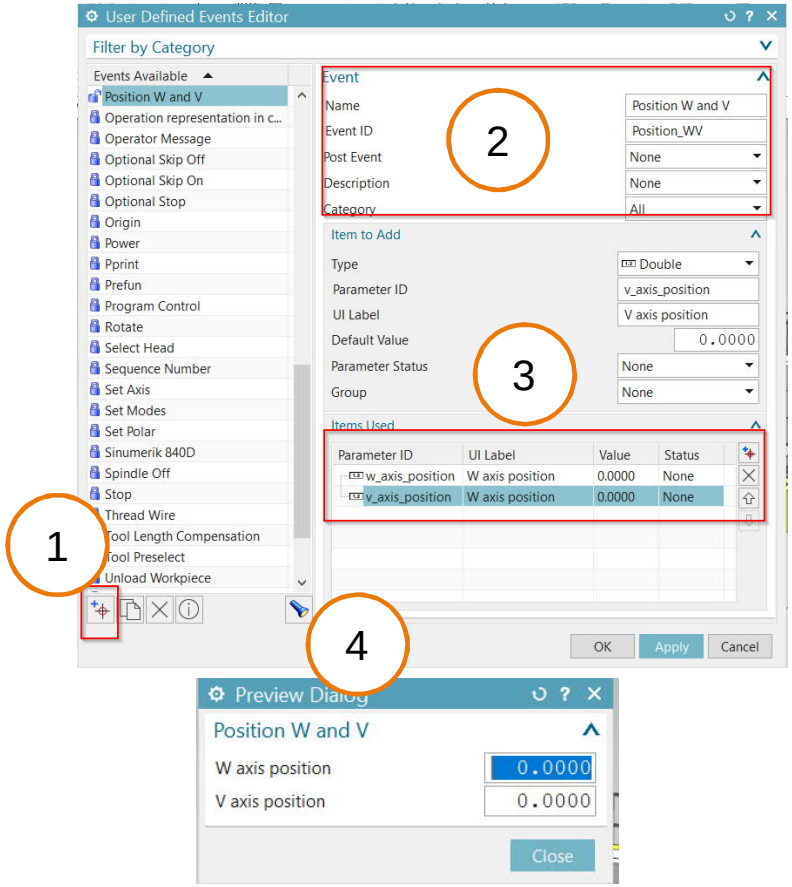
SIEMENS
Ingenuity for life



1. Open sim17 OOTB sample and Post Configurator application
2. Start the UDE editor to define a new UDE for W and V axis values

Task 1
Solution 2/15 – Create UDE for W and V axis

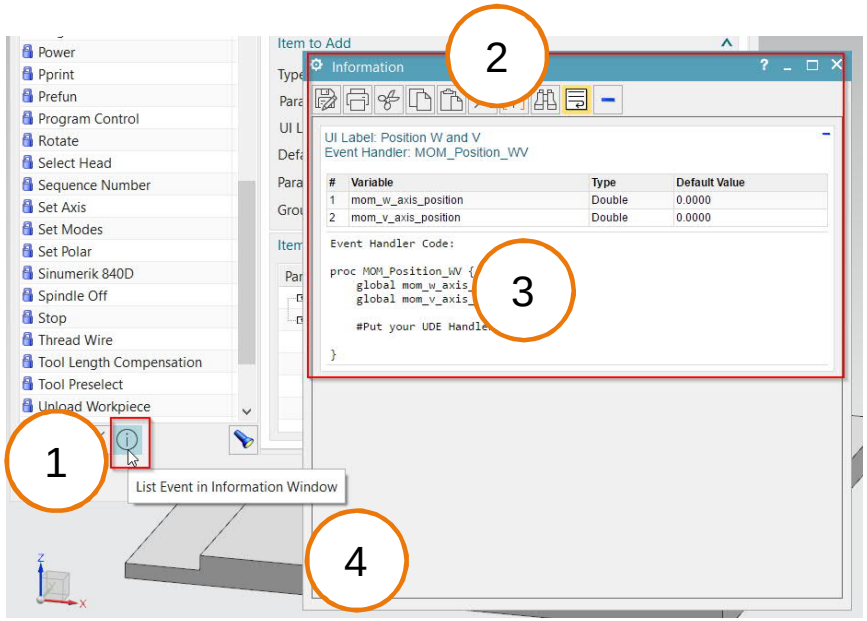
SIEMENS
Ingenuity for life



1. Add a new event for positioning W and V axis
2. Add UI name and MOM event name to execute later Tcl code
3. Add two new items as double to set up the values for W and V axis
4. Check the UDE with the Preview Button and that this contains the created items

Task 1
Solution 3/15 – Get created Tcl information

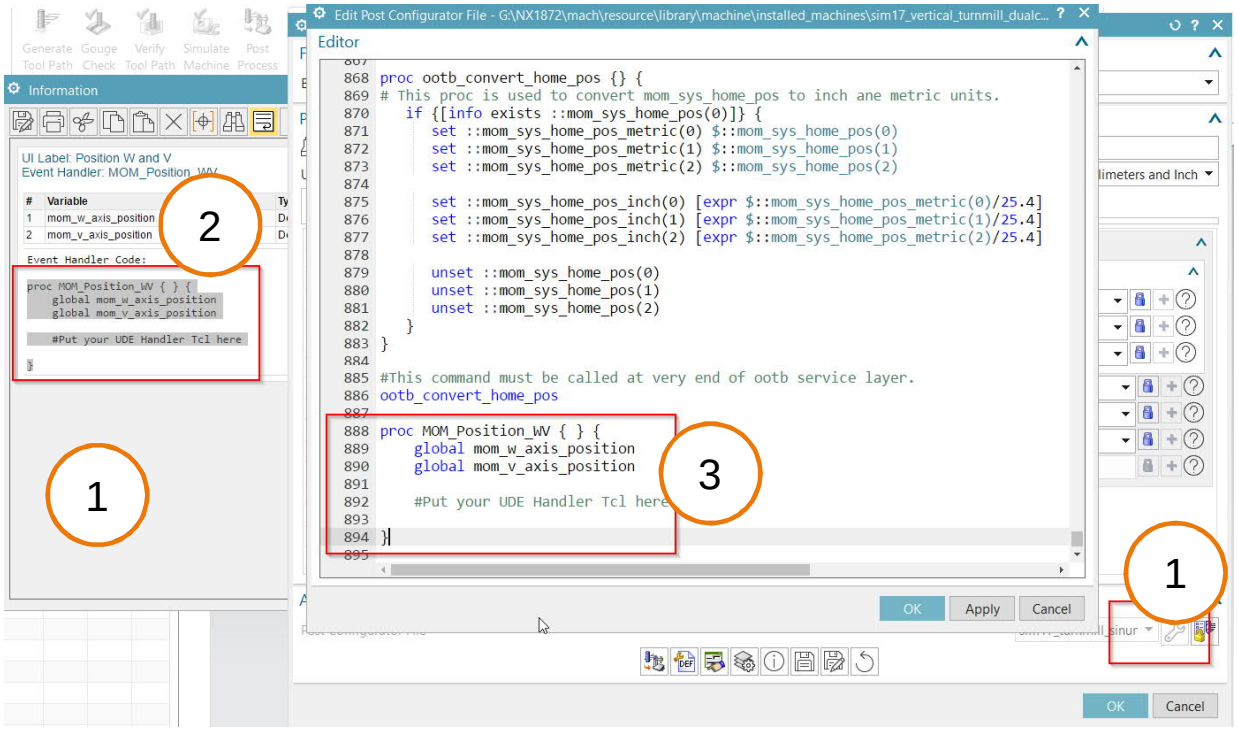
SIEMENS
Ingenuity for life



1. Add a new event for positioning W and V axis
2. Add UI name and MOM event name to execute later Tcl code
3. Add two new items as double to set up the values for W and V axis
4. Check the UDE with the Preview Button and that this contains the created items

Task 1
Solution 4/15 – Add UDE event to service file

SIEMENS
Ingenuity for life



1. Close UDE editor and keep information listing open and open the service file/ layer
2. Copy the eventhandler information from listing window
3. Add eventhandler to service file

Task 1
Solution 5/15 – Create new group and property

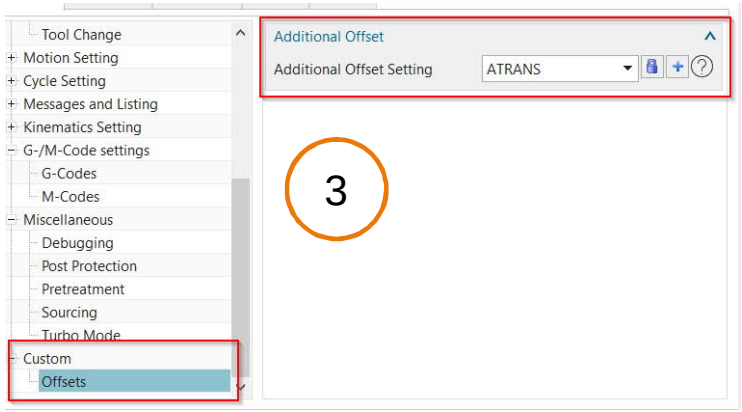
SIEMENS
Ingenuity for life

```
LIB_GE_CREATE_obj Custom_tree {UI_TREE} {
   LIB_GE_property_ui_name       "UIName"
   LIB_GE_property_ui_tooltip    "UITooltip"

set id "cst_tree_root"
   set $id "0"
   set datatype($id)       "NODE"
   set access($id)         222
   set dialog($id)         {{Custom}}
   set descr($id)          {{Custom Elements}}
   set group_status($id)   0
   set ui_parent($id)      "root"
   set ui_sequence($id)    -1

set id "cst_tree_node_offset"
   set $id "0"
   set datatype($id)       "NODE"
   set access($id)         222
   set dialog($id)         {{Offsets}}
   set descr($id)          {{Node description}}
   set group_status($id)   0
   set ui_parent($id)      "cst_tree_root"
   set ui_sequence($id)    -1

set id "cst_tree_group_additional_offset"
   set $id "0"
   set datatype($id)       "GROUP"
   set access($id)         222
   set dialog($id)         {{Additional Offset}}
   set descr($id)          {{Group description}}
   set group_status($id)   1
   set ui_parent($id)      "cst_tree_node_offset"
   set ui_sequence($id)    -1
}
```

1

```
LIB_GE_CREATE_obj CST_CTRL {} {
   LIB_GE_property_ui_name       "Name"
   LIB_GE_property_ui_tooltip    "Tooltip"

set id "additional_offset"
   set $id 0
   set options($id)        {ATRANS|WaCo L9958}
   set options_ids($id)    {0|1}
   set access($id)         222
   set dialog($id)         {{Additional Offset Setting}}
   set descr($id)          {{A False or True Property}}
   set ui_parent($id)      "cst_tree_group_additional_offset"
   set ui_sequence($id)    -1

}
```

2

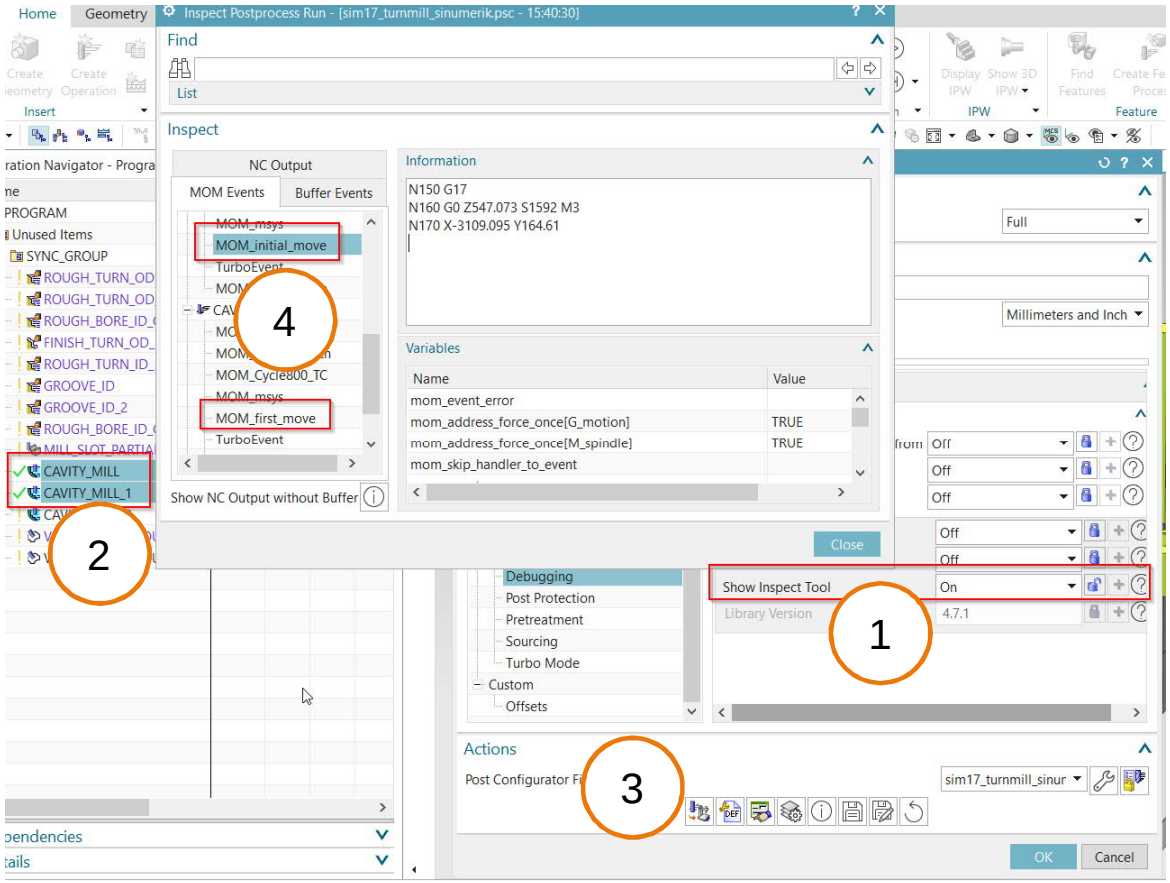


3

1. Create and define a new custom node and group for property settings
2. Create the property which contains different kind of output for additional offsets
3. Check UI that the property is visible and options ATRANS or WaCo L9958 can be choosen by the user

Task 1
Solution 6/15 – Get Buffers to modify

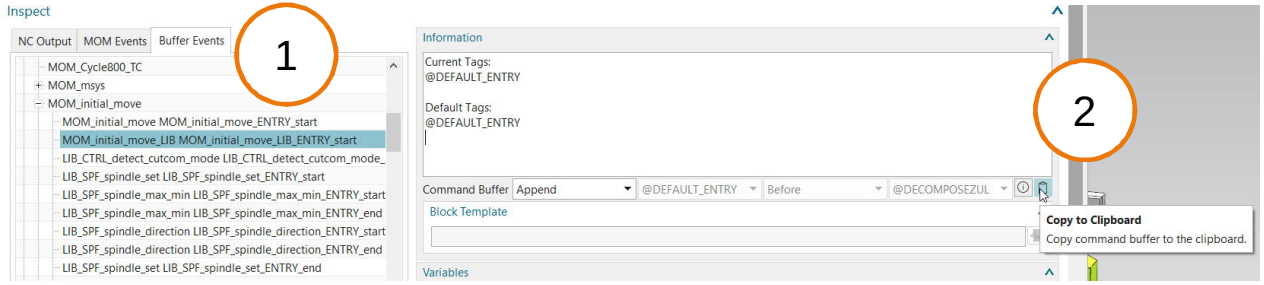
SIEMENS
Ingenuity for life

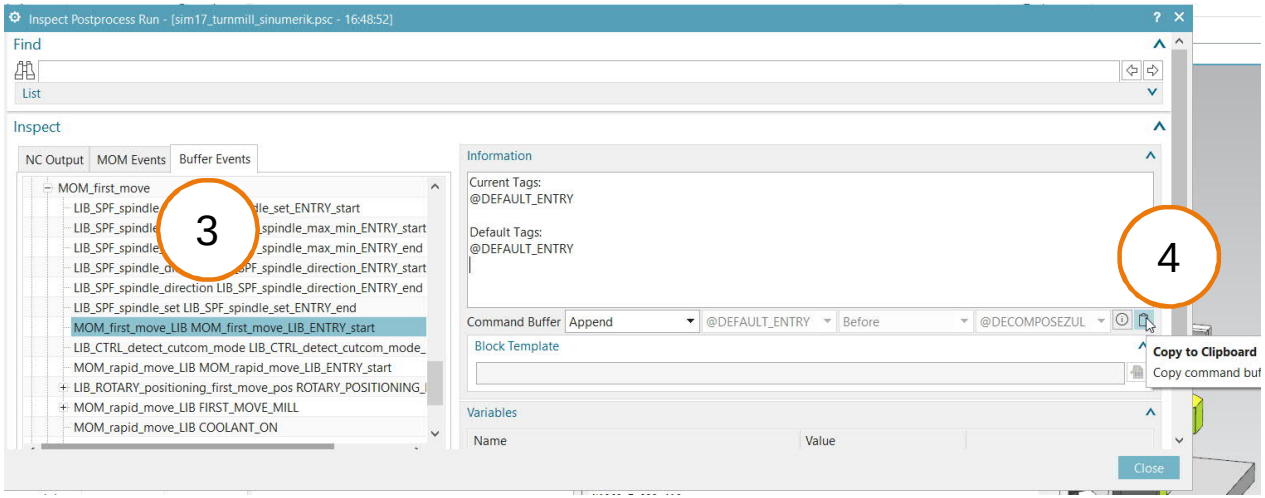


1. Switch On the Inspect Tool to find the customization point for implementing the Offset output, in this case we should output the additional offsets before the first motion in each operation
2. Select 2 operations, ideally without tool change to get the MOM_initial_move and MOM_first_move event
3. Use integrated post process option
4. Search mentioned MOM events in the MOM Events tab

Task 1
Solution 7/15 – Copy Buffers to modify

SIEMENS
Ingenuity for life



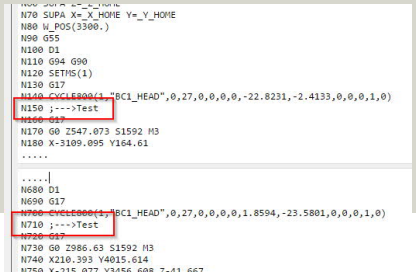


1. Because we need both MOM evevnts for the customization use the MOM_initial_move_LIB_entry_start with append or prepend command
2. Copy to clipboard and insert into service file (do not close the Inspect window)
3. Search in second operation for the MOM_first_move_LIB_entry_start buffer and select the buffer
4. Use copy to clipboard and insert into service file

Task 1
Solution 8/15 – Add new procedure

SIEMENS
Ingenuity for life

1. Create a new procedure body with Autocompletion functionality
2. Rename the procedure, this one is called later in the Buffer
3. Replace <code> and <tag> by the procedure name and the tag which should be have a spoken name
4. For testing work temporary with a MOM_output_literal command to validate with post process that output is on correct place



Task 1
Solution 9/15 – Tcl procedure property logic

SIEMENS
Ingenuity for life

```
#-------------------------------------------------------------
proc Cst_AdditionalOffsets {} {
#-------------------------------------------------------------
#if UDE is not attached to program or operation, set values to 0 otherwise to avoid Tcl error
if {![info exists ::mom_w_axis_position] || ![info exists ::mom_v_axis_position]} {
   set ::mom_w_axis_position 0
   set ::mom_v_axis_position 0
}

#ask property value setting to decide which output
switch [CST_CTRL additional_offset] {
   0 {
      #do something with ATRANS
      MOM_output_literal ";--->ATRANS"
   }
   1 {
      #do something with L9958
      MOM_output_literal ";--->L9958"
   }
   default {
      #do something
   }
}
}
```

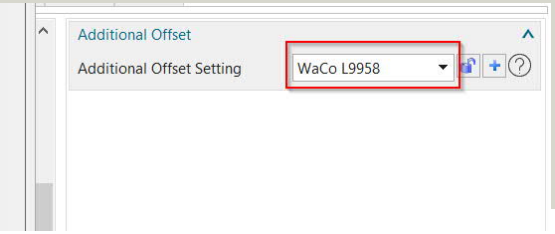
1

2

```
LIB_GE_CREATE_obj CST_CTRL {} {
    LIB_GE_property_ui_name        "Name"
    LIB_GE_property_ui_tooltip     "Tooltip"

set id "additional_offset"
    set $id 0
    set options($id)          {ATRANS|WaCo L9958}
    set options_ids($id)      {0|1}
    set access($id)           222
    set dialog($id)           {{Additional Offset
    set descr($id)            {{A False or True Pr
    set ui_parent($id)        "cst_tree_group_addi
    set ui_sequence($id)      -1

}
```

3

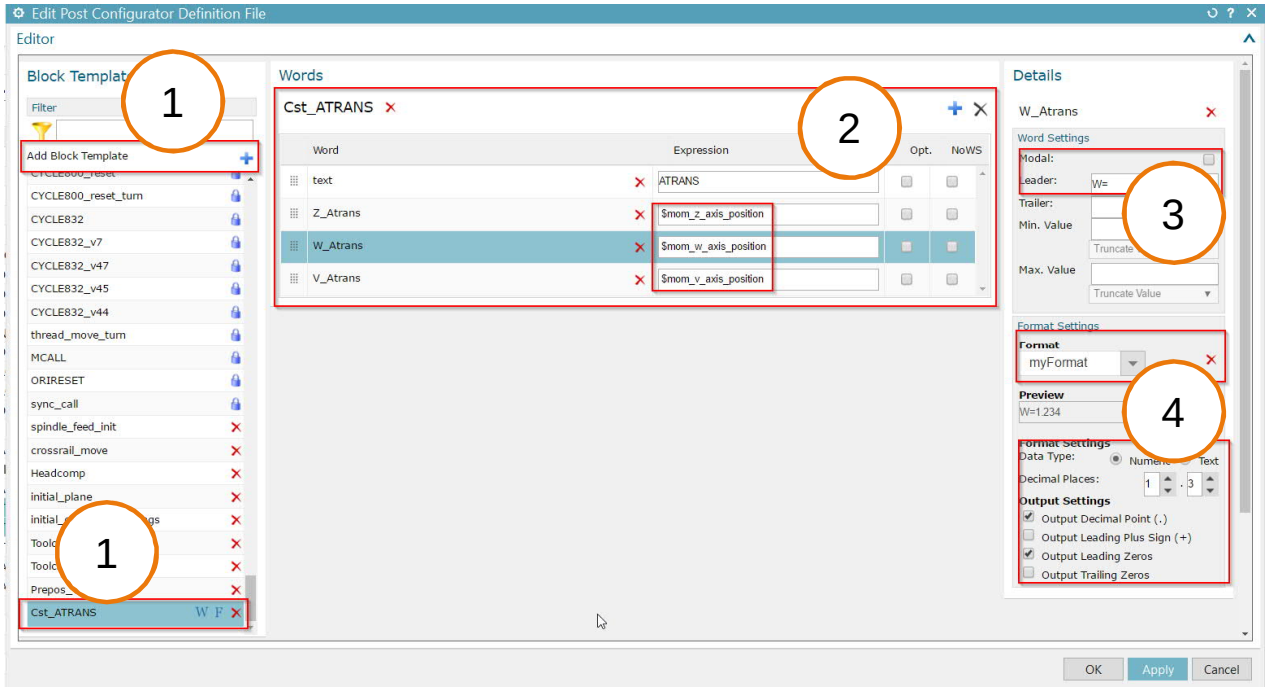
```
N620 TRAFOOF
N630 B_POS(0)
N640 SUPA Z=_Z_HOME D0
N650 SUPA X=_X_HOME Y=_Y_HOME
N660 SUPA Z=_Z_HOME
N670 SUPA X=_X_HOME Y=_Y_HOME
N680 D1
N690 G17
N700 CYCLE800(1,"BC1_HEAD",0,27,0,0,0,0,1.8594,-23.5801,0,0,0,1,0)
N710 ;--->L9958
N720 G17
N730 G0 Z986.63 S1592 M3
N740 X210.393 Y4015.614
N750 X-215.077 Y3456.608 Z-41.667
```



1. First check with a condition if Ude variables are exists, otherwise set those variables to 0 (global variables!)
2. Ask the value of the defined property in a switch block, will return in this case 0 or 1
3. For testing simply output the MOM_output_literal and post process. Change the property to the other value and it should run into second case

Task 1
Solution 10/15 – Add Block Template ATRANS

SIEMENS
Ingenuity for life



1. Add a new Block Template (take care that when creating a new Block Template it will use the last selected)
2. Add text address for ATRANS word and create new address for translation for Z,W,V. The expressions are linked to the mom variables of the UDE and the additional mom_z_axis_position
3. Set the leader and remove modal to get the output everytime even if values are the same
4. Create a new format as double with output leading zeros and assign to all created addresses

Task 1
Solution 11/15 – Add Block Template ATRANS movement

SIEMENS
Ingenuity for life

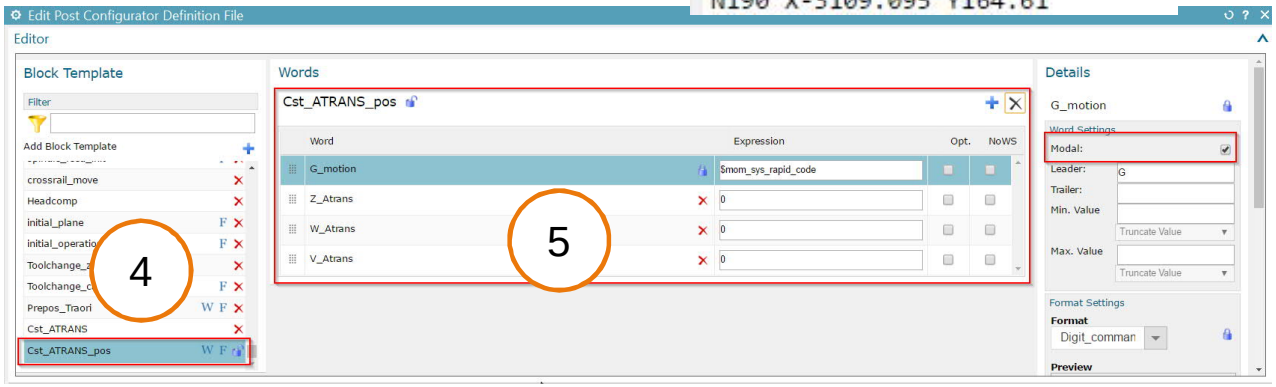
```
#-------------------------------------------------
proc Cst_AdditionalOffsets {} {
#-------------------------------------------------
#if UDE is not attached to program or operation,
if {![info exists ::mom_w_axis_position] || ![in
    set ::mom_w_axis_position 0
    set ::mom_v_axis_position 0
    set ::mom_z_axis_position 0
}

#ask property value setting to decide which outpu
switch [CST_CTRL additional_offset]
    0 {
        #do something with ATRANS
        MOM_output_literal ";--->ATRANS"
        MOM_do_template Cst_ATRANS
    }
    1 {
```

1

2

```
N150 ;--->ATRANS
N160 ATRANS Z=0. W=0. V=0.
N170 G17
N180 G0 Z547.073 S1592 M3
N190 X-3109.095 Y164.61
```

3



4

5

1. Add additional mom_z_axis_position variable in the code (will be calculated automatically later on)
2. Calling the created Block template in Tcl if property is set to ATRANS
3. Run post process and check first output
4. Create second Block Template for the positioning movement after ATRANS (create copy from existing ATRANS)
5. Add existing G_motion address with standard rapid code (e.g. copy from rapid_move template) and set expressions to zero

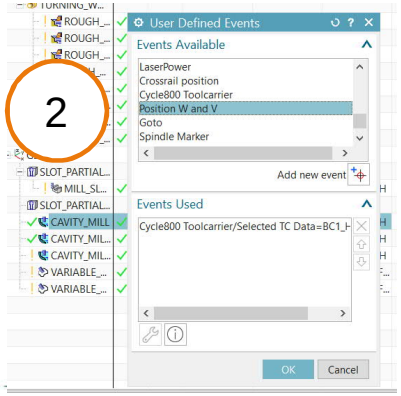
Task 1
Solution 12/15 – Add Z compensation logic

SIEMENS
Ingenuity for life

```
#ask property value setting to decide which output
switch [CST_CTRL additional_offset] {
    0 {
        #do something with ATRANS
        MOM_output_literal ";--->ATRANS"
        MOM_do_template Cst_ATRANS
        MOM_do_template Cst_ATRANS_pos
    }
```

1

```
N730 ;--->ATRANS
N740 ATRANS Z=0. W=0. V=0.
N750 G0 Z=0. W=0. V=0.
N760 G17
```



2

```
proc Cst_AdditionalOffsets {} {
#--------------------------------
#if UDE is not attached to program or operation, set values to 0 otherwise to avoid Tcl error
if {![info exists ::mom_w_axis_position] || ![info exists ::mom_v_axis_position]} {
    set ::mom_w_axis_position 0
    set ::mom_v_axis_position 0
    set ::mom_z_axis_position 0
} else {
#calculate Z-axis compensation
set ::mom_z_axis_position [expr ($::mom_w_axis_position + $::mom_v_axis_position) * -1]
}
```

3

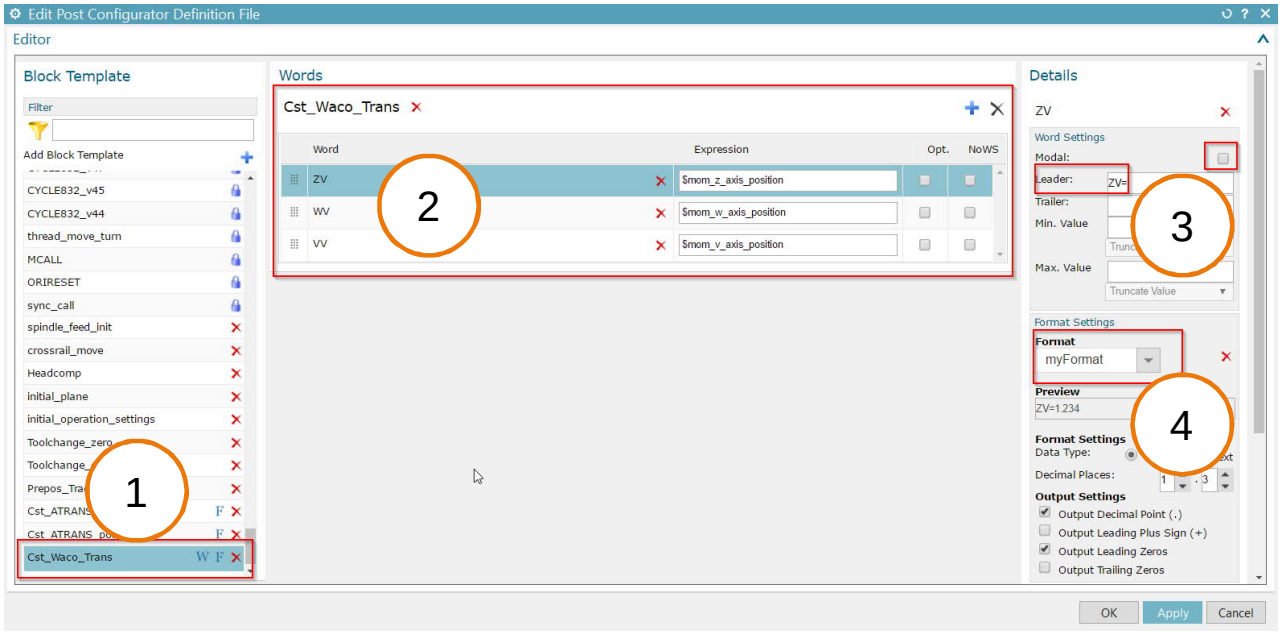
1. Calling additional Block Template in Tcl and run the post process to check result
2. Add the created UDE to the operation and add values for W(200) and V(100)
3. Add additional Tcl code to calculate the Z compensation if the Ude is attached to an operation
4. Check the result. All values of Z, W and V together should be equal 0 in the ATRANS line

```
N150 ;--->ATRANS
N160 ATRANS Z=-300. W=200. V=100.
N170 G0 Z=0. W=0. V=0.
N180 G17
N190 G0 Z547.073 S1592 M3
```

4

Task 1
Solution 13/15 – Add Block Template L9958

SIEMENS
Ingenuity for life



1. Add new Block Template for L9958 output
2. Create new Addresses for ZV, WV and VV with using same mom-variables
3. Setup the leader and remove modal flag
4. Reuse created format from ATRANS Block template

Task 1
Solution 14/15 – Add L9958 output

SIEMENS
Ingenuity for life

```
963
964 #ask property value setting to decide wh          tput
965 switch [CST_CTRL additional_offset] {
966     0 {
967         #do something with ATRANS
968         MOM_output_literal ";--->ATRANS"
969         MOM_do_template Cst_ATRANS
970         MOM_do_template Cst_ATRANS_pos
971     }
972     1 {
973         #do something with L9958
974         MOM_output_literal ";--->L9958"
975         MOM_do_template Cst_Waco_Trans
976         MOM_output_literal "L9958\(\)"
977     }
978     default {
979         #do something
980     }
981 }
```

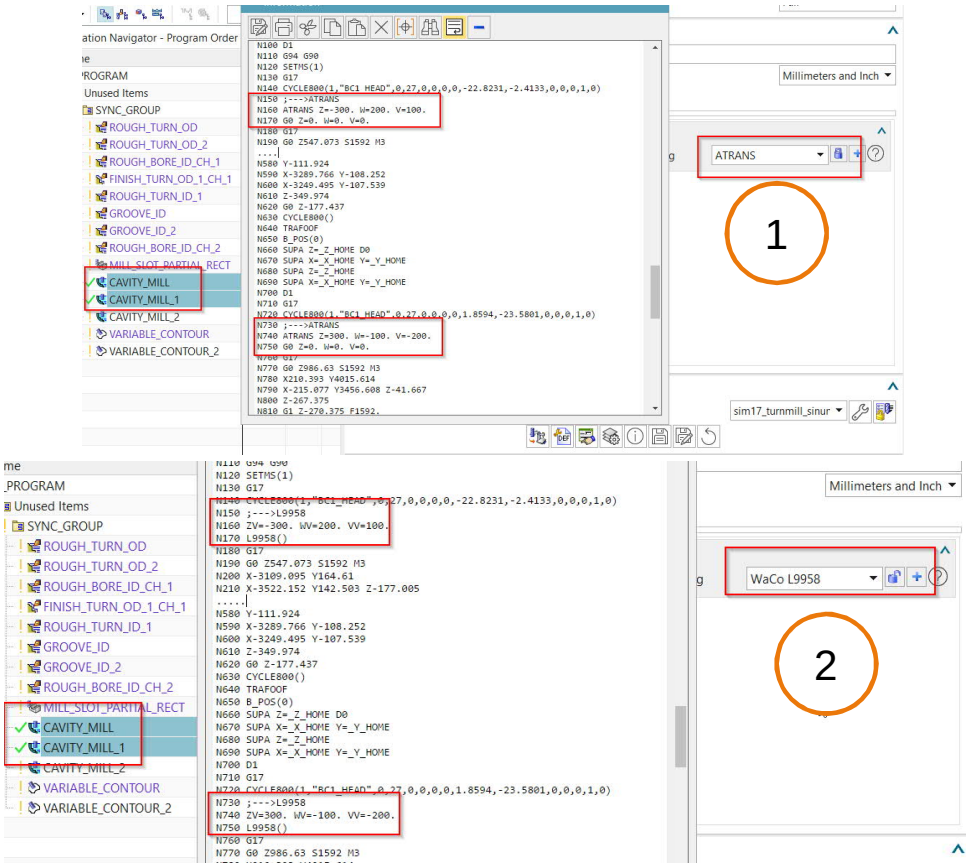
1

2

1. Add Tcl code in case of L9958 of property definition
2. Execute the created Block template and output the subprogram line

Task 1
Solution 15/15 – Validate settings

SIEMENS
Ingenuity for life



1. Add the UDE to 2 operations with different values and run post process with ATRANS setting and validate result
2. Run post process with L9958 option and validate result

Task 1
Conclusion Solution and Benefits

SIEMENS
Ingenuity for life

1. Should be reused in a seperate layer to minimize implementation effort for next post processor

2. One time effort higher, benefit of extremely time decrease for next projects

3. Take care that Cycle800 must be turned off when working with ATRANS (not included in this task)

4. Workflow is similiar to every other UDE

5. Learned in this task :
   - Using UDE editor
   - Using DEF editor
   - Using Tcl editor and Inspect Tool
   - Create and connect created properties
   - Start thinking about generic solutions

Task 2

SIEMENS
Ingenuity for life

Collect the information from the created UDE with pretreatment:

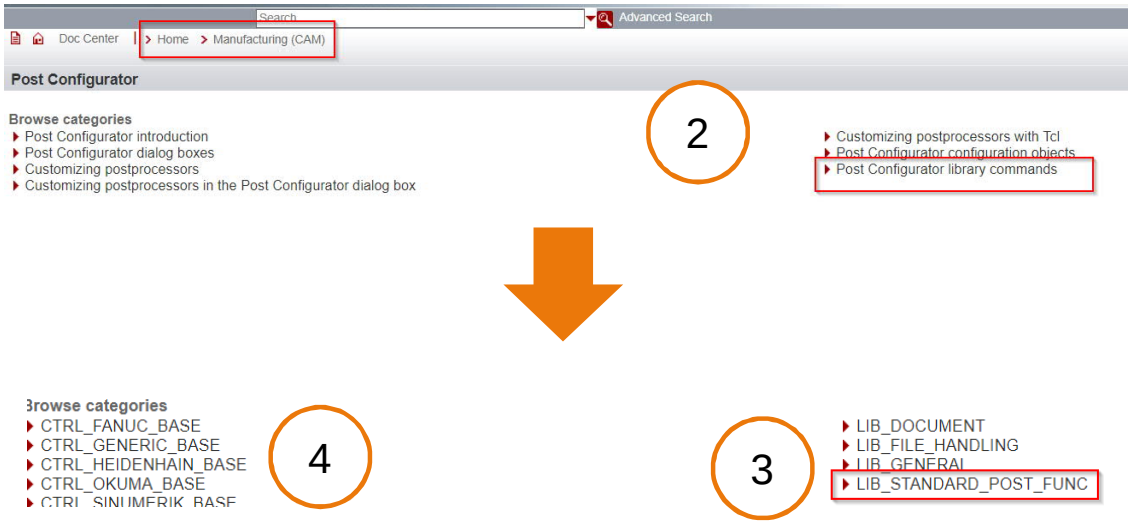
Conditions:

- If the UDE is attached to an operation the values of the transformation are output in the header with operation information (name and used tool)
- The user can control the output with a property to switch on/ off

Task 2
Solution 1/7 – Use PC online documentation

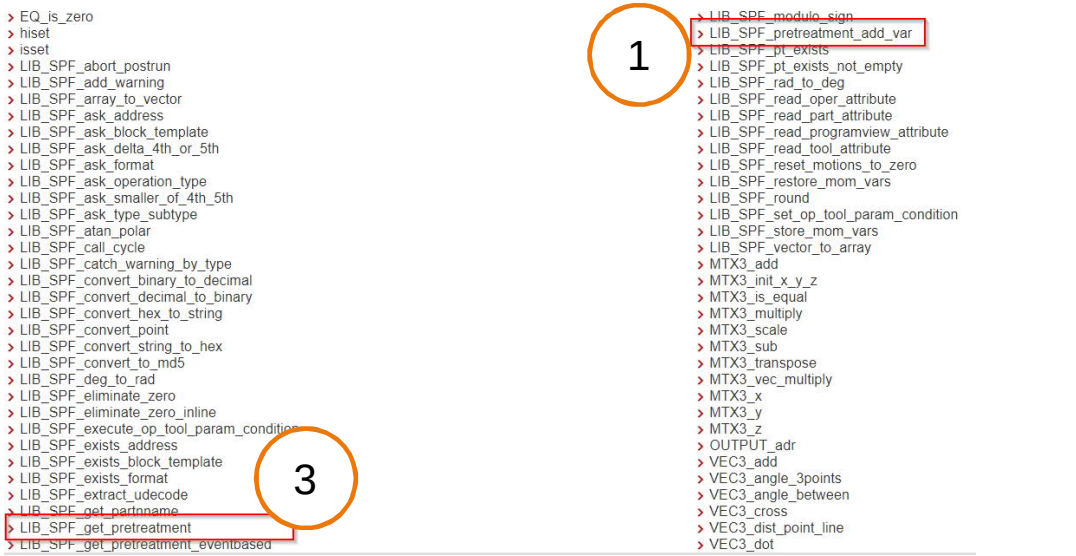
SIEMENS
Ingenuity for life

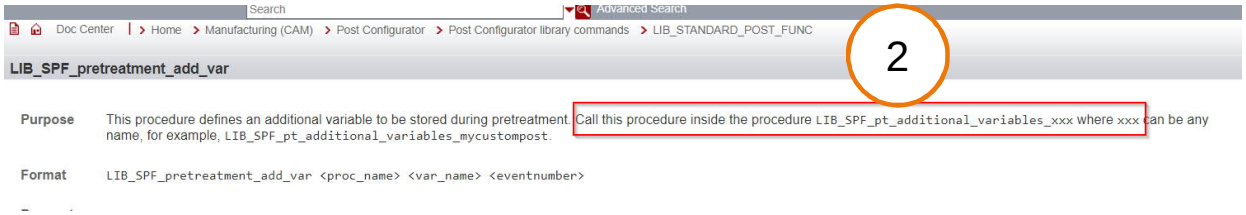
1 https://docs.plm.automation.siemens.com/tdoc/nx/1872/nx_help/#uid:xid1128418:index_mfgpostconfig



1. Use official online documentation for Post Configurator to search for standard procedures for reuse
2. On the Post Configurator documentation page use library commands section
3. Most of standard functionalities are defined in the LIB_STANDARD_POST_FUNC section which are controller independent. It‘s recommend to take 30min to read thru all of this to get a feeling what is available
4. CTRL specific definitions and general File handling are defined in different areas

Task 2
Solution 2/7 – Use PC online documentation

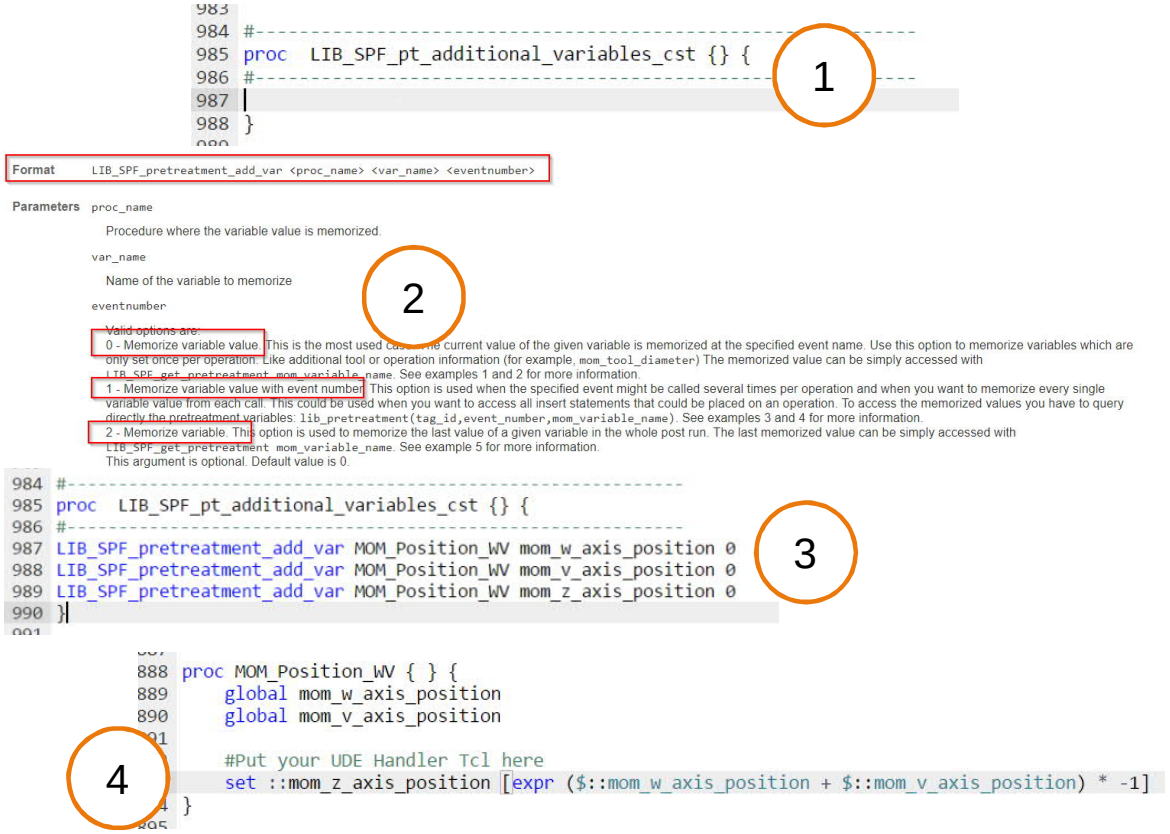




1. First it‘s needed to add the new UDE variables to the pretreatment to output them in the header. Open the LIB_SPF_pretreatment_add_var to read documention and how it should used
2. To add new variables it‘s needed to create a new procedure with a fixed name
3. LIB_SPF_get_pretreatment is to get values. Keep it in mind for later from documentation

Task 2
Solution 3/7 – Add variables to pretreatment

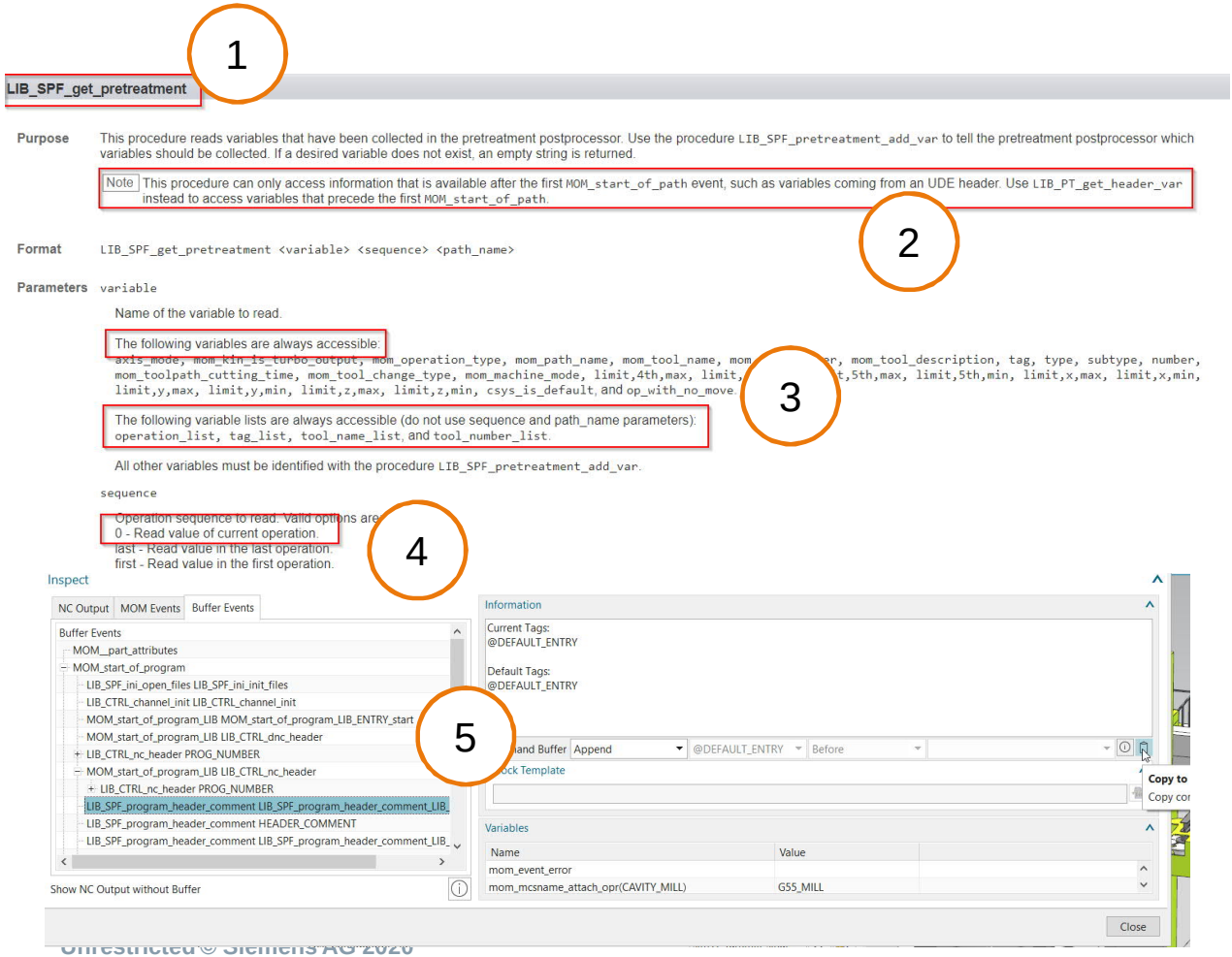
SIEMENS
Ingenuity for life



1. Create a new procedure with the name from documentation
2. Use documentation to find out how to collect and the arguments needed
3. Variables can collected from called MOM events. The UDE is an own event and the variables are accesible first time in this event. Add the MOM event name of the UDE and all variables to collect
4. Because the Z-axis compensation was calculated later in the procedure it‘s needed to add the calculation in the MOM event handler of UDE to have access in pretreatment

Task 2
Solution 4/7 – Output collected values in header

SIEMENS
Ingenuity for life



1. Open documention for LIB_SPF_get_pretreatment
2. Take care of the Note how to collect variables outside of operations (e.g. if the UDE is attached to the program group)
3. A set of variables is always collected and multiple lists are available to cycle thru the collected values. The operation list will be used in next step
4. Argument which values should be collected
5. Run Inspect Tool to get Buffer for the output, e.g. the program header comment buffer

Task 2
Solution 5/7 – Add logic to cycle thru collected values

SIEMENS
Ingenuity for life

```
991
992 LIB_GE_command_buffer_edit_append LIB_SPF_program_header_comment LIB_SPF_program_header_comment_LIB_ENTRY_start OutputPretreatmentVars _AdditionaPTVars
993
994 #-------------------------------------------------------------
995 proc OutputPretreatmentVars {} {
996 #-------------------------------------------------------------
997
998 }
999
```

1

1. Call the new procedure in the program header comment buffer and add a Tagname
2. With foreach from automatically collected operation list cycle thru all operations
3. In each operation output the mom_path_name and mom_tool_name which are also collected automatically
4. Output the variables added to the pretreatment from the UDE definition

```
#-------------------------------------------------------------
proc OutputPretreatmentVars {} {
#-------------------------------------------------------------
foreach e [LIB_SPF_get_pretreatment operation_list] {
#Format and output listing
   MOM_output_text ";Operation Name:[LIB_SPF_get_pretreatment mom_path_name 0 $e] Toolname:[LIB_SPF_get_pretreatment mom_tool_name 0 $e]"
   MOM_output_text ";Additive offset Z=[LIB_SPF_get_pretreatment mom_z_axis_position 0 $e]"
   MOM_output_text ";Additive offset V=[LIB_SPF_get_pretreatment mom_v_axis_position 0 $e]"
   MOM_output_text ";Additive offset W=[LIB_SPF_get_pretreatment mom_w_axis_position 0 $e]"
   MOM_output_text ";"
}

}
```

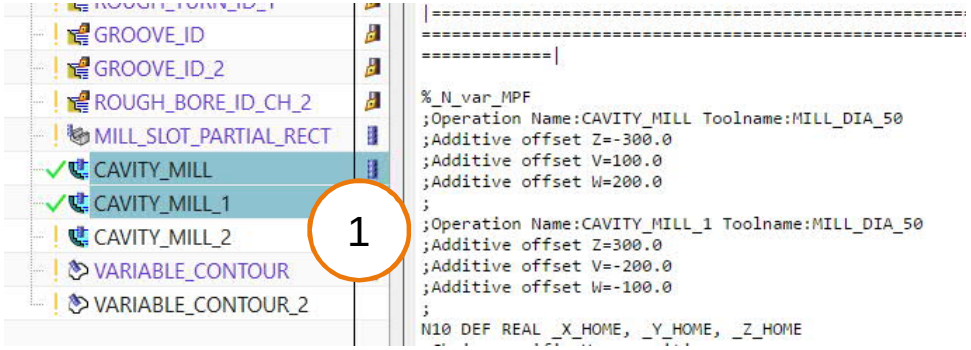
2

3

4

Task 2
Solution 6/7 – Validate Output

SIEMENS
Ingenuity for life



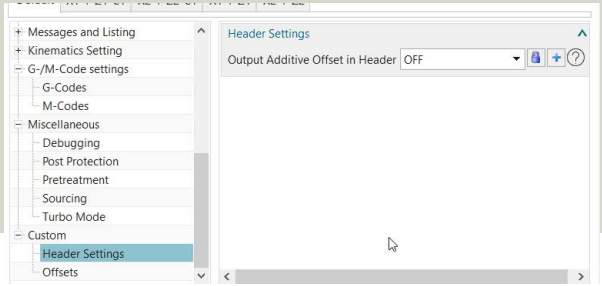

```
set id "cst_tree_group_additional_offset"
    set $id "0"
    set datatype($id)       "GROUP"
    set access($id)         222
    set dialog($id)         {{Additional Offset}}
    set descr($id)          {{Group description}}
    set group_status($id)   1
    set ui_parent($id)      "cst_tree_node_offset"
    set ui_sequence($id)    -1

set id "cst_tree_node_header"
    set $id "0"
    set datatype($id)       "NODE"
    set access($id)         222
    set dialog($id)         {{Header Settings}}
    set descr($id)          {{Node description}}
    set group_status($id)   0
    set ui_parent($id)      "cst_tree_root"
    set ui_sequence($id)    -1

set id "cst_tree_group_header"
    set $id "0"
    set datatype($id)       "GROUP"
    set access($id)         222
    set dialog($id)         {{Header Settings}}
    set descr($id)          {{Group description}}
    set group_status($id)   1
    set ui_parent($id)      "cst_tree_node_header"
    set ui_sequence($id)    -1
}
```

2

```

LIB_GE_CREATE_obj CST_CTRL {} {
    LIB_GE_property_ui_name         "Name"
    LIB_GE_property_ui_tooltip      "Tooltip"

set id "additional_offset"
    set $id 0
    set options($id)        {ATRANS|WaCo L9958}
    set options_ids($id)    {0|1}
    set access($id)         222
    set dialog($id)         {{Additional Offset Setting}}
    set descr($id)          {{A False or True Property}}
    set ui_parent($id)      "cst_tree_group_additional_offset"
    set ui_sequence($id)    -1

set id "header_offset_output"
    set $id 0
    set options($id)        {OFF|ON}
    set options_ids($id)    {0|1}
    set access($id)         222
    set dialog($id)         {{Output Additive Offset in Header}}
    set descr($id)          {{Output additive offset values from UDE in header with Tool name and Operation name }}
    set ui_parent($id)      "cst_tree_group_header"
    set ui_sequence($id)    -1
}
```

3

1. Run the post process with 2 operations where the UDE is attached. Values with operation name and tool name are output in the header
2. Add new group for Header settings in the Custom_tree object definition
3. Add a new property to control the output

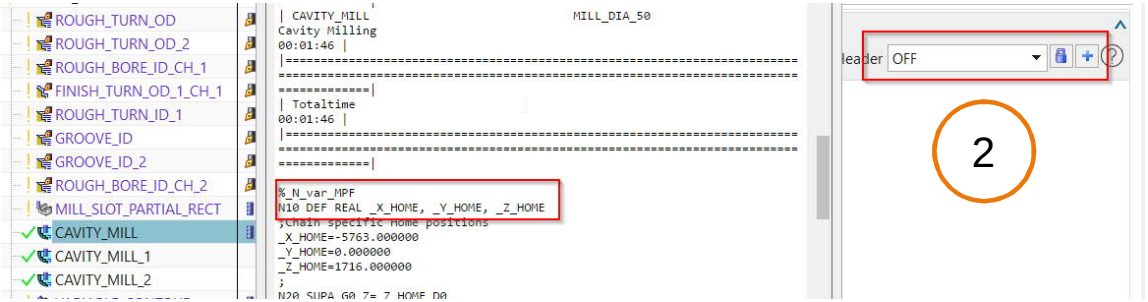


Task 2
Solution 7/7 – Connect Property to command and Validate

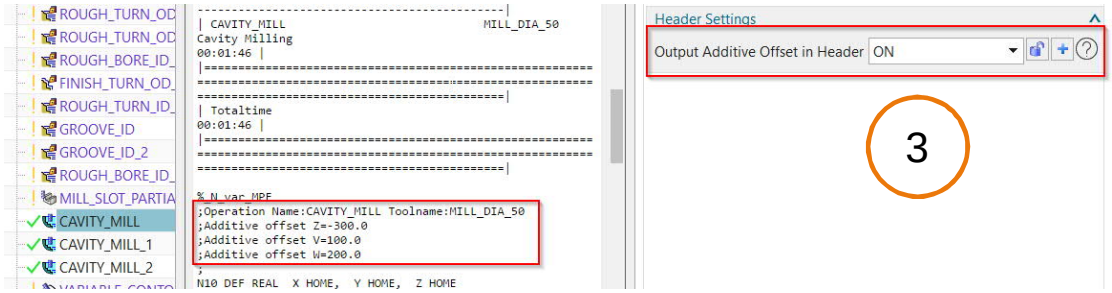
SIEMENS
Ingenuity for life

```
1024 #-------------------------------------------------------------
1025 proc OutputPretreatmentVars {} {
1026 #-------------------------------------------------------------
1027
1028 if {[CST_CTRL header_offset_output] == 0} {return}
1029
```

1



2



3

1. Add logic to return if the created property is set to OFF in the OutputPretreatmentVars procedure which contains the output of values
2. Validate with option OFF, no output should appears
3. Validate with option ON, output should appear if UDE is attached to operation

Task 2
Conclusion Solution and Benefits

SIEMENS
Ingenuity for life

1. Should be reused in a seperate layer to minimize implementation effort for next post processor

2. One time effort higher, benefit of extremely time decrease for next projects

3. Documentation contains a lot of predefined functions for reuse, take 30min or 1hour to go thru documenation to get a feeling what is available

4. Workflow is similiar for each variable which is later available but needed in beginning of program

5. Learned in this task:
   - Using Documentation
   - Using Pretreatment
   - Using Tcl editor and Inspect Tool
   - Create and connect created properties

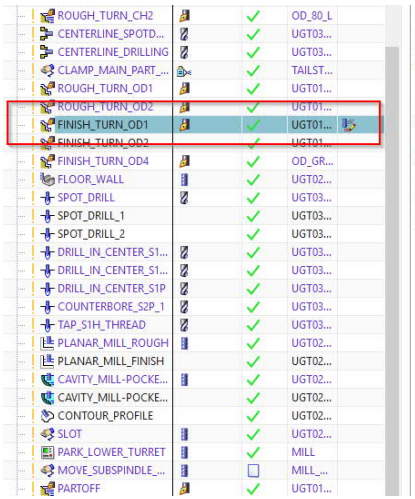
Task 3

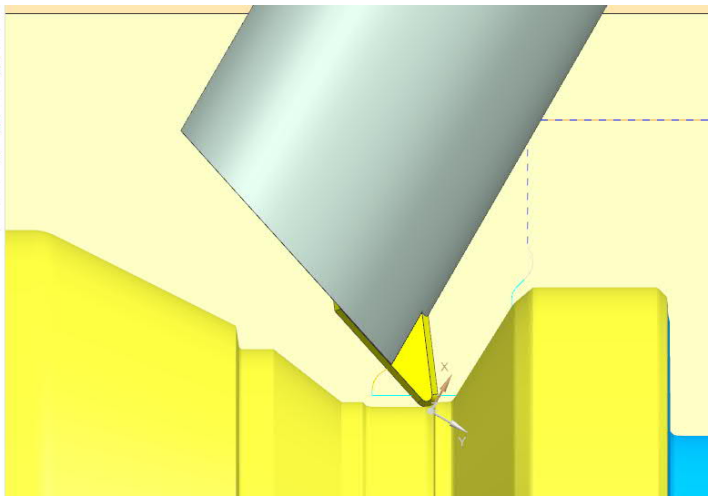
SIEMENS
Ingenuity for life

The customer request a B-axis turning capability in the postprocessor (sim15 Sinumerik):

Conditions:

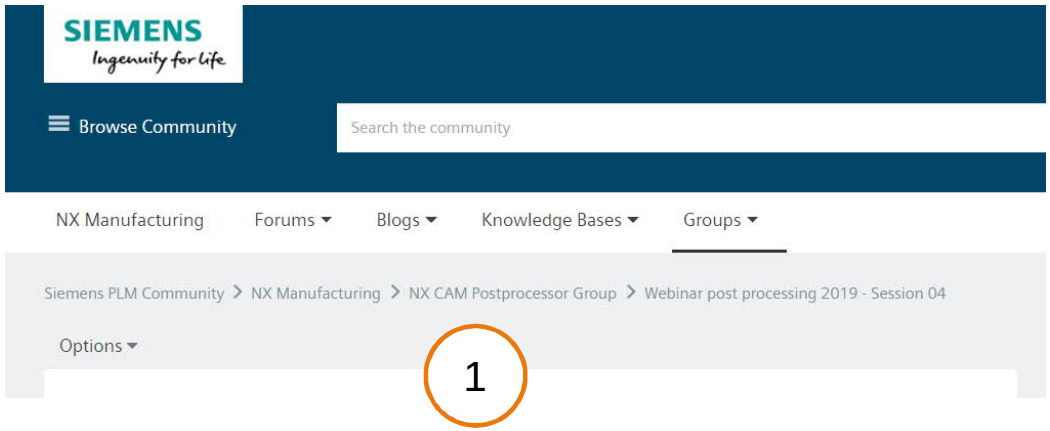
- Use the **…\Handsout\CAM_parts\B_axis_Turning\sim15_millturn_setup_sinumerik_mm_dual_sync.prt** from training material
- Option is needed to switch on and off
- Activate and deactivate correctly in NC code, e.g. when switching back to milling mode





```
N40 TC(56,70,180)
N50 G54
N60 TRANS X0. Y0. Z0.
N70 GETD(C4)
N80 TRAORI
N90 DIAMON
N100 SETMS(4)
N110 SPOS[1]=180.
N120 G0 X110. Y0. Z232.6 D1
N130 G95 S4=2000 M4=3
N140 X58.2
N150 G18 G3 G90 X53.4 Z230.2 I-2.4 K0. F.4
N160 G1 X52.4 Z230.2 B50.
N170 X52.4 Z235.885 B60.
N180 X52.4 Z244. B80.
N190 G3 X54. Z244.8 I0.8 K0.
N200 G1 X68.703 Z244.8 B70.
N210 X72.793 Z244.8
N220 G2 X74.844 Z245.225 I0. K1.45
N230 G1 X76.551 Z246.078
N240 X76.722 Z246.172
N250 G3 X83.484 Z246.467 I1.839 K-1.543 F.5
N260 G0 X110.
N270 Z323.
N280 TRAFOOF
N290 DIAMOF
N300 TRANS
N310 M4=5
N320 G97
N330 SUPA X598.7 D0
```

Task 3
Solution – Implement B-axis turning



https://community.plm.automation.siemens.com/t5/NX-CAM-Postprocessor-Group/Webinar-post-processing-2019-Session-04/gpm-p/606377

1. Webinar session for B-axis turning is available online and how to proceed
2. With NX1872 the sim15 millturn Sinumerik example contains a sample implementation of B-axis turning

```
LIB_GE_command_buffer_edit_prepend MOM_end_of_path_LIB MOM_end_of_path_LIB_ENTRY_start {CheckTurnMode} _CheckMode
LIB_GE_command_buffer_edit_prepend MOM_linear_move_LIB OUTPUT {CalculateBRotation} _CalculateB
LIB_GE_command_buffer_edit_prepend MOM_circular_move_LIB OUTPUT {CalculateBRotation} _CalculateB

#-------------------------------------------------------------------
proc Detect_B_axis_mode {} {
#-------------------------------------------------------------------
if {[string match "Turn Finishing" $::mom_operation_type]} {
    if {[info exists ::mom_use_b_axis] && $::mom_use_b_axis == 2} {
      MOM_output_literal "GETD(C4)"
      MOM_do_template traori
      MOM_enable_address fourth_axis
    } else {
      MOM_disable_address fourth_axis
    }
} else {
   MOM_disable_address fourth_axis
}
}

#-------------------------------------------------------------------
proc CalculateBRotation {} {
#-------------------------------------------------------------------
#set initial B-axis init value for turning
if {[info exists ::mom_use_b_axis]  && $::mom_use_b_axis == 2} {
   set initial_baxis_value 0

   set B_rotation_caluclation [expr $::RAD2DEG*atan2($::mom_tool_axis(0),$::mom_tool_axis(2))]
   set ::B_rotation [expr $initial_baxis_value + $B_rotation_caluclation]
   MOM_force once X Z
}
}

#-------------------------------------------------------------------
proc CheckTurnMode {} {
#-------------------------------------------------------------------
if {[info exists ::mom_use_b_axis]  && $::mom_use_b_axis == 2} {
    MOM_do_template trafoof
    LIB_TURNING_mode "end"
    }
}
```

2

Task 4

SIEMENS
Ingenuity for life

If the motion type is changing the customer request the output of the current motion type in the NC code. (e.g. for deposition tools in stepover moves):

Conditions:

- It is **not allowed** to output this request with **Turbomode off option**
- Option is needed to switch on and off
- Property settings for M-code Deposition tool ON/ OFF needed
- M code is outputted on the correct location in NC code
- Count the numbers of change for stepover, engage, retract, cut, traversal and output this formatted in end of program (for sure option is needed in UI ☺)

Task 4
Solution 1/4 – Add Properties for reuse

SIEMENS
Ingenuity for life

1

```
949
950 set id "cst_tree_node_deposition"
951     set $id "0"
952     set datatype($id)       "NODE"
953     set access($id)         222
954     set dialog($id)         {{Deposition Settings}}
955     set descr($id)          {{Node description}}
956     set group_status($id)   0
957     set ui_parent($id)      "cst_tree_root"
958     set ui_sequence($id)    -1
959
960 set id "cst_tree_group_deposition"
961     set $id "0"
962     set datatype($id)       "GROUP"
963     set access($id)         222
964     set dialog($id)         {{Deposition Settings}}
965     set descr($id)          {{Group description}}
966     set group_status($id)   1
967     set ui_parent($id)      "cst_tree_node_deposition"
968     set ui_sequence($id)    -1
969 }
970
```

2

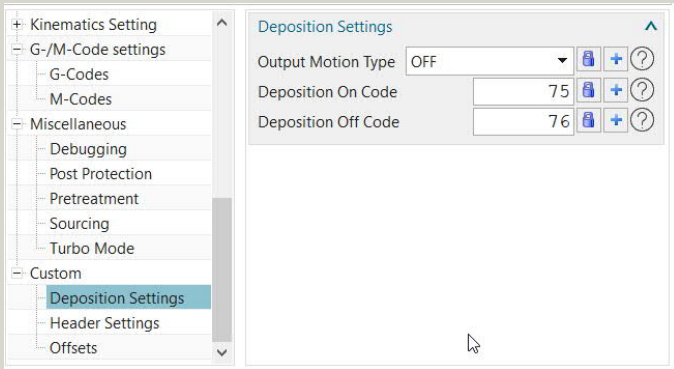
```
993     set ui_sequence($id)    -1
994
995 set id "motion_output_turbo"
996     set $id 0
997     set options($id)        {OFF|ON}
998     set options_ids($id)    {0|1}
999     set access($id)         222
1000    set dialog($id)         {{Output Motion Type}}
1001    set descr($id)          {{Output motion type in Turbo mode}}
1002    set ui_parent($id)      "cst_tree_group_deposition"
1003    set ui_sequence($id)    -1
1004 }
```

3

```
1004
1005 set id "deposition_on_code"
1006     set $id 75
1007     set options($id)        {*VALUE*}
1008     set options_ids($id)    {*VALUE*}
1009     set datatype($id)       "INT"
1010     set access($id)         222
1011     set dialog($id)         {{Deposition On Code}}
1012     set descr($id)          {{A numeric Property}}
1013     set ui_parent($id)      "cst_tree_group_deposition"
1014     set ui_sequence($id)    -1
1015
1016 set id "deposition_off_code"
1017     set $id 76
1018     set options($id)        {*VALUE*}
1019     set options_ids($id)    {*VALUE*}
1020     set datatype($id)       "INT"
1021     set access($id)         222
1022     set dialog($id)         {{Deposition Off Code}}
1023     set descr($id)          {{A numeric Property}}
1024     set ui_parent($id)      "cst_tree_group_deposition"
1025     set ui_sequence($id)    -1
1026
```

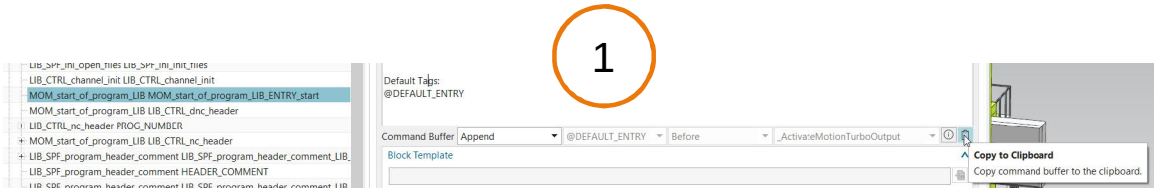
4

1. Create a new node and group in Custom_tree object
2. Create a property to switch on/off the output of motion type in Turbo mode
3. Create a property for deposition on with default value 75
4. Create a property for deposition on with default value 76



Task 4
Solution 2/4 – Activate MOM_before_motion in Turbo mode

SIEMENS
Ingenuity for life

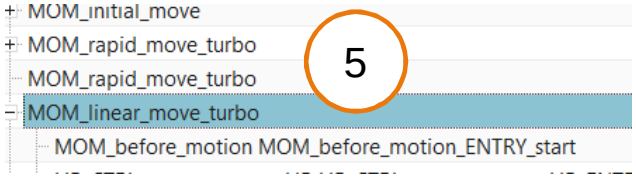



```
LIB_GE_command_buffer_edit_append MOM_start_of_program_LIB MOM_start_of_program_LIB_ENTRY_start
ActivateTurboMotionType _ActivateMotionTurboOutput

#-------------------------------------------------------------
proc ActivateTurboMotionType {} {
#-------------------------------------------------------------
#if property is set to off return this procedure
if {[CST_CTRL motion_output_turbo] == 0} {return}
#if property is set to on it's needed to switch to advanced turbo mode
CONF_CTRL_setting set turbo_mode 3
#this MOM command activate the MOM_before_motion customization
MOM_set_turbo_before_motion ON
}
```

2

3

4

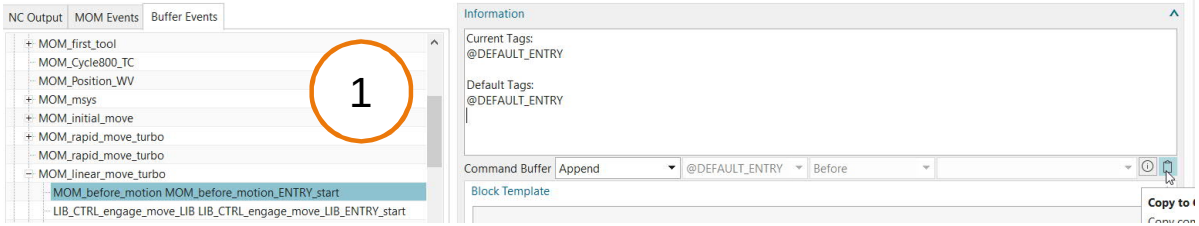


5

1. Run Inspect Tool and append new command in start of program to activate extended functionality with Turbo mode
2. Call procedure and define a Tag name with the additional customization
3. Dependent on property setting return this procedure or activate the motion output
4. To influence Turbo mode the turbo mode must switch to Advanced mode, set the property by Tcl and activate the Motion type with MOM_set_turbo_before_motion ON
5. Run the post process with Inspect Tool and instead of Turbo event Turbo events for each motion appears with customization possibilities

Task 4
Solution 3/4 – Output Motion types with Turbo

SIEMENS
Ingenuity for life

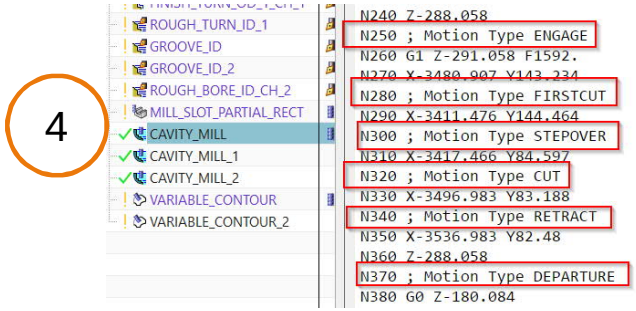


LIB_GE_command_buffer_edit_append MOM_before_motion MOM_before_motion_ENTRY_start TurboMotionTypes _TurboMotionTypes

2

```
#-------------------------------------------------------------
proc TurboMotionTypes {} {
#-------------------------------------------------------------
MOM_output_literal "; Motion Type $::mom_motion_type"
}
```

3



1. Run post process with Inspect Tool and use MOM_before_motion Buffer from the Turbo event to extend
2. Create procedure and call this in the Buffer. Add a Tag as identifier
3. Output the Motiontype simply by a MOM_output_literal
4. Run post process and check that now each motion change is outputted. Be aware of the fact that only motion changes will be outputted, means if multiple cut motions than only the first one will be outputted if there is no further motion type change

Task 4
Solution 4/4 – Add counting of changed motion types

SIEMENS
Ingenuity for life

```
#-------------------------------------------------------------
proc ActivateTurboMotionType {} {
#-------------------------------------------------------------
#if property is set to off return this procedure
if {[CST_CTRL motion_output_turbo] == 0} {return}
#if property is set to on it's needed to switch to advanced turbo mode
CONF_CTRL_setting set turbo_mode 3
#this MOM command activate the MOM_before_motion customization
MOM_set_turbo_before_motion ON

set ::mom_cst_motionchange_cut 0
set ::mom_cst_motionchange_firstcut 0
set ::mom_cst_motionchange_traversal 0
set ::mom_cst_motionchange_approach 0
set ::mom_cst_motionchange_departure 0
set ::mom_cst_motionchange_engage 0
set ::mom_cst_motionchange_retract 0
set ::mom_cst_motionchange_stepover 0
}
```

1

```
#-------------------------------------------------------------
proc TurboMotionTypes {} {
#-------------------------------------------------------------
MOM_output_literal "; Motion Type $::mom_motion_type"
switch $::mom_motion_type {
    "CUT" {incr ::mom_cst_motionchange_cut}
    "FIRSTCUT" {incr ::mom_cst_motionchange_firstcut}
    "TRAVERSAL" {incr ::mom_cst_motionchange_traversal}
    "APPROACH" {incr ::mom_cst_motionchange_approach}
    "DEPARTURE" {incr ::mom_cst_motionchange_departure}
    "ENGAGE" {incr ::mom_cst_motionchange_engage}
    "RETRACT" {incr ::mom_cst_motionchange_retract}
    "STEPOVER" {incr ::mom_cst_motionchange_stepover}
}
}
```

2

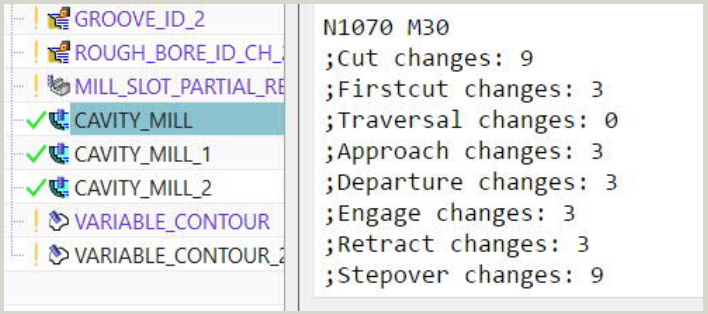
3

```
LIB_GE_command_buffer_edit_append MOM_end_of_program_LIB MOM_end_of_program_LIB_ENTRY_end Motionchanges _stepover

#-------------------------------------------------------------
proc Motionchanges {} {
#
if {[CST_CTRL motion_output_turbo] == 0} {return}
MOM_output_text ";Cut changes: $::mom_cst_motionchange_cut"
MOM_output_text ";Firstcut changes: $::mom_cst_motionchange_firstcut"
MOM_output_text ";Traversal changes: $::mom_cst_motionchange_traversal"
MOM_output_text ";Approach changes: $::mom_cst_motionchange_approach"
MOM_output_text ";Departure changes: $::mom_cst_motionchange_departure"
MOM_output_text ";Engage changes: $::mom_cst_motionchange_engage"
MOM_output_text ";Retract changes: $::mom_cst_motionchange_retract"
MOM_output_text ";Stepover changes: $::mom_cst_motionchange_stepover"
}
```

4

5

1. Add initial counter in the ActivateTurboMotionType procedure for each motion type
2. In the TurboMotionTypes add a switch condition and increment the counter by one if the motion type is changed
3. Use Inspect Tool and last Buffer in end of program to output the motion type changes
4. Activate this output only if the property is switched On, otherwise it may will result in Tcl error
5. Output all motion type changes counter variables



Task 4
Conclusion Solution and Benefits

SIEMENS
Ingenuity for life

1. Using advantage of Turbo mode for high performance with possibilities to customize even in the motions, especially for additive

2. Should be reused in a seperate layer to minimize implementation effort for next post processor

3. One time effort higher, benefit of extremely time decrease for next projects

4. Further information to Turbo mode can be found in documentation

5. This solution not contains the call of M-commands for deposition but can be easily added by your own now

6. Learned in this task:
   - Using Turbo mode advanced
   - Using Tcl editor and Inspect Tool
   - Create and connect created properties

Task 5

SIEMENS
Ingenuity for life

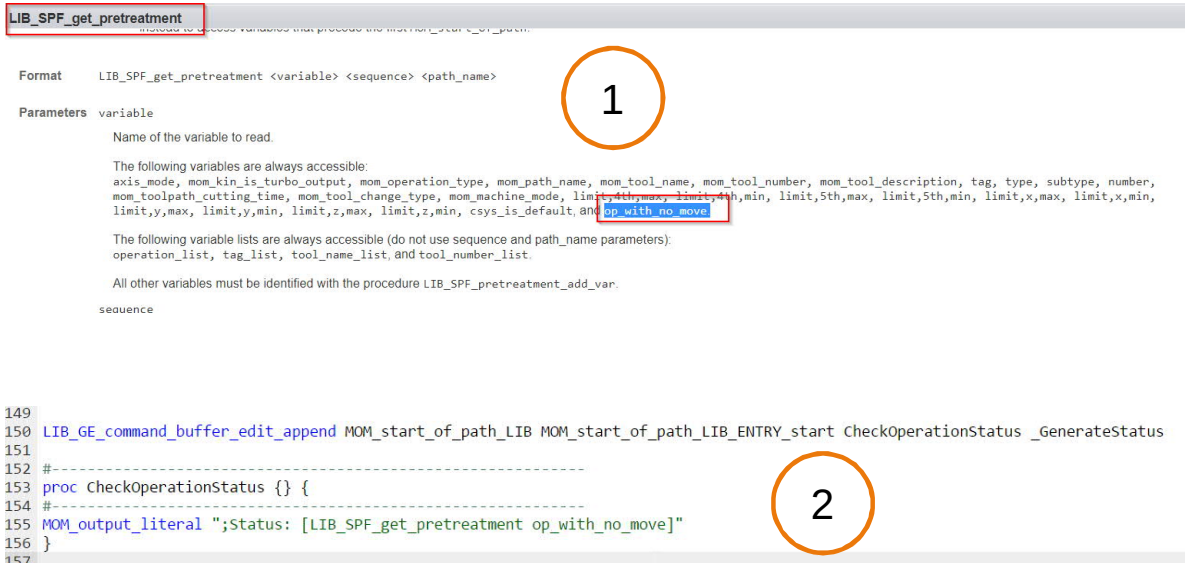
Warning if toolpath is empty:

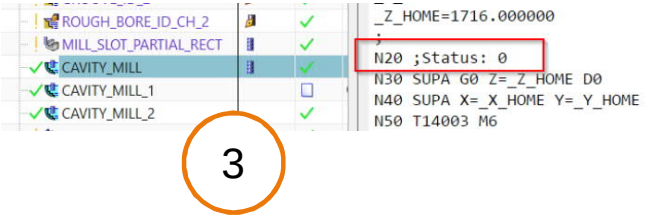
Conditions:

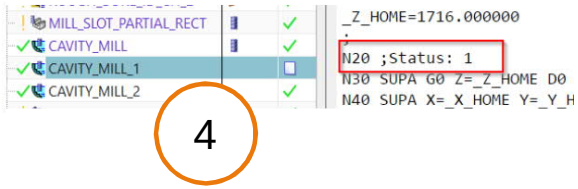
- Using standard Messagebox from Post Configurator library
- Option is needed in UI

Task 5
Solution 1/3 – Search needed variables

SIEMENS
Ingenuity for life



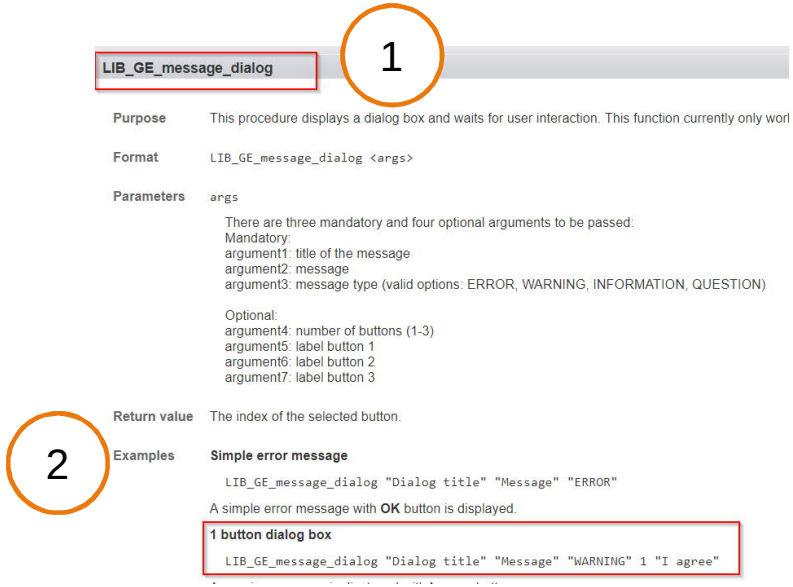


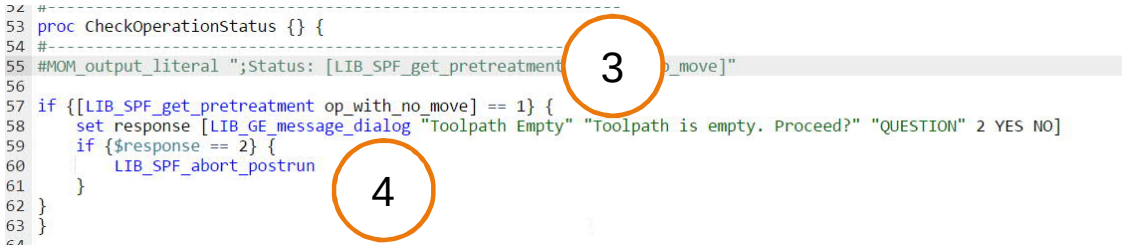


1. Search in NX documentation for variable which contains this information, if not found anything search standard functionality from Post Configurator documentation, in this case in pretreatment there is a variable op_with_no_moves which indicates toolpath is empty
2. Add a procedure in Start_of_path buffer and output the value from pretreatment to find out which values are returned
3. Post process an operation with toolpath will return 0
4. Post process an operation without toolpath will return 1

Task 5
Solution 2/3 – Ask status and create Message dialog

SIEMENS
Ingenuity for life

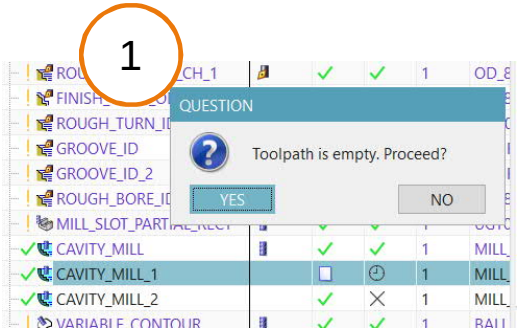


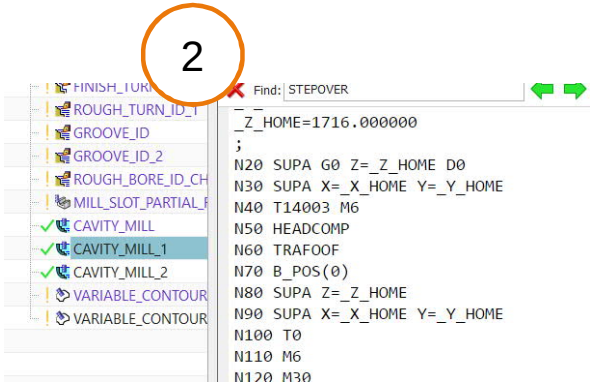


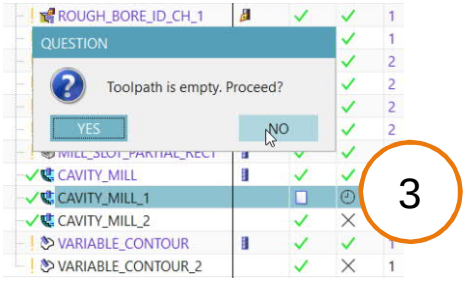
1. Use documentation of Post Configurator to find standard Message dialogs, LIB_GE_message_dialog
2. In the examples copy the sample with 2 buttons into service file
3. Create condition if op_with_no_moves the dialog comes up with yes and no question to proceed
4. If response no abort the post process run with additional standard command from Post Configurator libraries

Task 5
Solution 3/3 – Validate modifications

SIEMENS
Ingenuity for life







1. Run post process with operation and empty toolpath
2. If question is answered with yes some output should appear
3. If question is answered with no the post process is aborted without any output

Task 5
Conclusion Solution and Benefits

SIEMENS
Ingenuity for life

1. Using documentation to find standard commands instead of reinvent the wheel

2. Should be reused in a seperate layer to minimize implementation effort for next post processor

3. One time effort higher, benefit of extremely time decrease for next projects

4. Standard methods for messages are available

5. Learned in this task:
   - Using Documentation
   - Using Tcl editor and Inspect Tool

Task 7

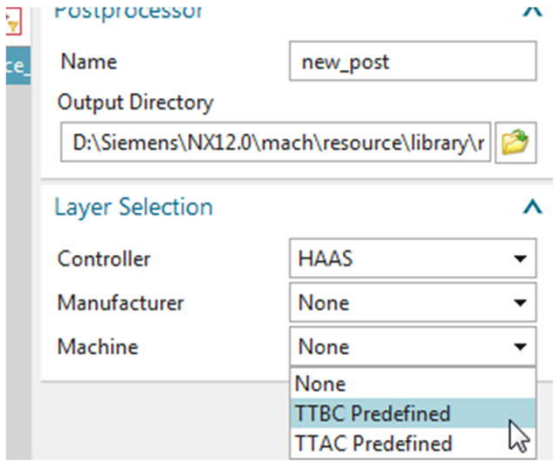
SIEMENS
Ingenuity for life

Generic kinematic layer which can be reused to configure predefined kinematic types (HT, HH, TT, 3axis, 2-axis lathe)

Conditions:

- Layer contains all important settings (mom_kin_xxx variables)
- Is integrated in Post create environment in machine section

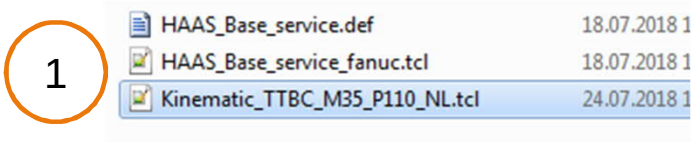
Task 7
Solution 1/15 – General purpose

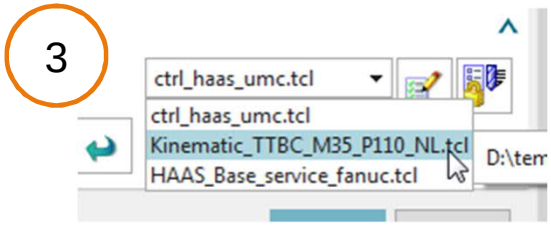


Post Configurator use MTB informations. If no model is loaded it will use a 45°-BC Headtable kinematic. To create Postprocessors without MTB but with preconfigured kinematics it‘s possible to add them during create process. These layers can be reused by different controllers.

Task 7
Solution 2/15 – Additional kinematic layers (TTBC)

SIEMENS
Ingenuity for life





1. Create a new file with information about the kinematic, e.g. kinematic type _ MinLimit4th_MaxLimit4th_MinLimit5th_MaxLimit5th/ nolimits
2. Include this file in the temporary postprocessor behind the mtb layer. When a postprocessor will be created the new one created will be located here automtically
3. Open Post Configurator and select the new created file and open it with the Tcl editor

Task 7
Solution 3/15 – Additional kinematic layers (TTBC)

SIEMENS
Ingenuity for life

```
##############################################################################################
#
#   Kinematic Layer TTBC
#
#   Company         : Siemens Industry Software
#   Contact person  : Thomas Jenensch
#   Mail            : thomas.jenensch@siemens.com
#
##############################################################################################
#
#   Copyright 2018 Siemens Industry Software
#               All Rights Reserved.
#
##############################################################################################
# Description
# This layer contains predefined kinematic condi
# kinematic model.
#
# Kinematic type: Table Table
# 4th axis: B
# 4th axis Max limit: 110
# 4th axis Min limit: -35
# 5th axis: C
# 5th axis limits: no limits
##############################################################################################
# History
# 24-07-18   TJ   Initial version
##############################################################################################
#
### General kinemtic settings ###
set ::mom_machine_mode MILL
set ::mom_kin_machine_type 5_axis_dual_table
set mom_sys_leader(X)                 X
set mom_sys_leader(Y)                 Y
set mom_sys_leader(Z)                 Z
set mom_kin_spindle_axis(2)           1.0
### General kinemtic settings ###

### 4th axis settings ###
set mom_kin_4th_axis_leader           B
set mom_kin_4th_axis_min_incr         0.001
set mom_kin_4th_axis_point(0)         0.0
set mom_kin_4th_axis_point(1)         0.0
set mom_kin_4th_axis_point(2)         0.0
set mom_sys_4th_axis_has_limits       1
set mom_kin_4th_axis_max_limit        110
set mom_kin_4th_axis_soft_max_limit   110
set mom_kin_4th_axis_min_limit        -35
set mom_kin_4th_axis_soft_min_limit   -35
set mom_kin_4th_axis_type             Table
set mom_kin_4th_axis_vector(0)        0
set mom_kin_4th_axis_vector(1)        1
set mom_kin_4th_axis_vector(2)        0
set mom_kin_4th_axis_zero             0.0
### 4th axis settings ###

### 5th axis settings ###
set mom_kin_5th_axis_leader           C
set mom_kin_5th_axis_min_incr         0.001
set mom_kin_5th_axis_point(0)         0.0
set mom_kin_5th_axis_point(1)         0.0
set mom_kin_5th_axis_point(2)         0.0
set mom_sys_5th_axis_has_limits       0
set mom_kin_5th_axis_max_limit        3600000
set mom_kin_5th_axis_soft_max_limit   3600000
set mom_kin_5th_axis_min_limit        -3600000
set mom_kin_5th_axis_soft_min_limit   -3600000
set mom_kin_5th_axis_type             Table
set mom_kin_5th_axis_vector(0)        0
set mom_kin_5th_axis_vector(1)        0
set mom_kin_5th_axis_vector(2)        1
set mom_kin_5th_axis_zero             0.0
### 5th axis settings ###
```

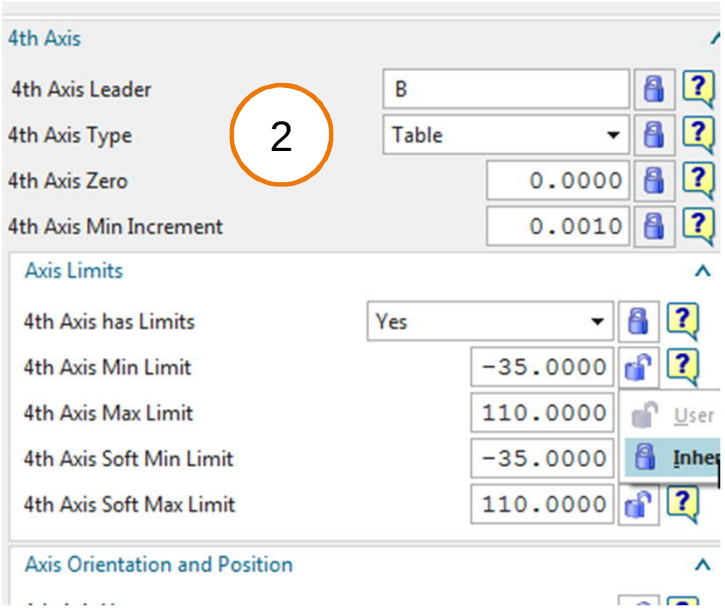
1

2

1. Add description of the layer and generic information about the kinematic type
2. Add all necessary mom-variables which are needed for the kinematics. The information which mom-variable is needed is located in the Tooltip help.

3. After finishing of set up the layer save and close the Tcl-editor.

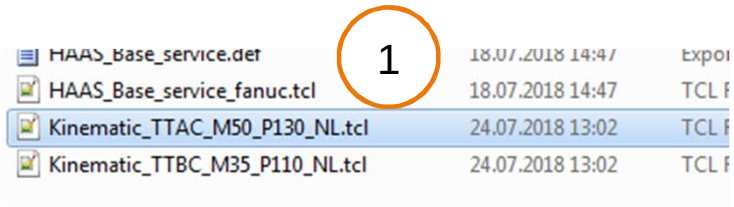
Task 7
Solution 4/15 – Additional kinematic layers (TTBC)

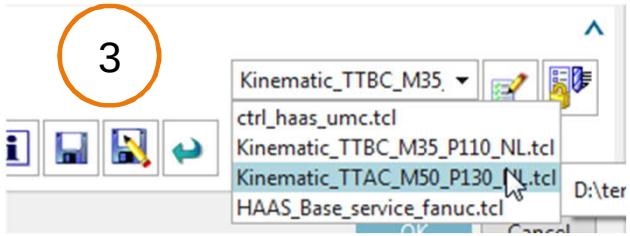


1. Now check that this layer is sourced correctly and all changed properties get now the new default value from the layer for the machine type.
2. Repeat this for all properties which was changed in the UI of Post Configurator (open lock).

Task 7
Solution 5/15 – Additional kinematic layers (TTAC)

SIEMENS
Ingenuity for life

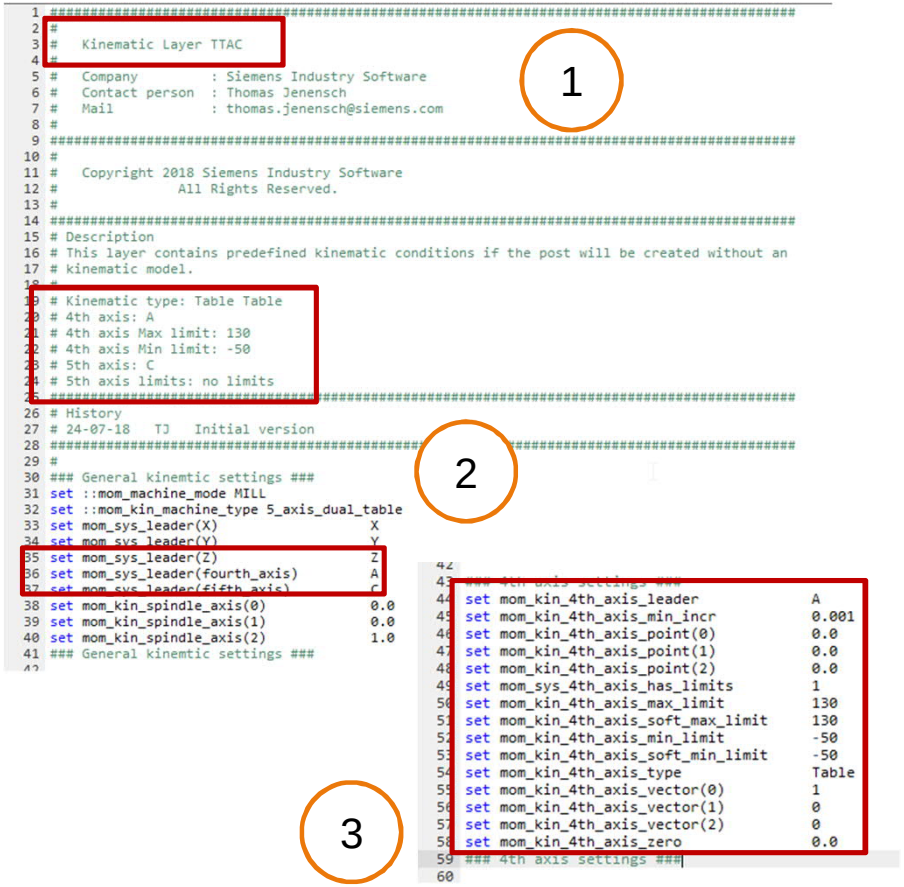




1. Now concept of reusing running very well. Copy the first created kinematic file
2. Add this file also in sourcing in psc file (Important: behind the TTBC-kinematic)
3. Open the file with Post Configurator

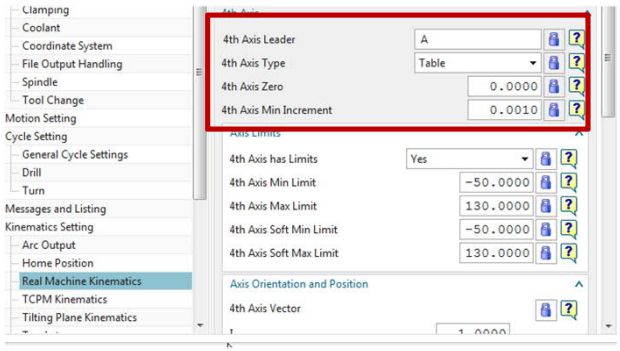
Task 7
Solution 6/15 – Additional kinematic layers (TTAC)

SIEMENS
Ingenuity for life



1. Add additional information in header
2. Change general settings, e.g. leader
3. Change all needed mom-variables for 4th axis (leader, vector, limits, type)

Task 7
Solution 7/15 – Additional kinematic layers (TTAC)

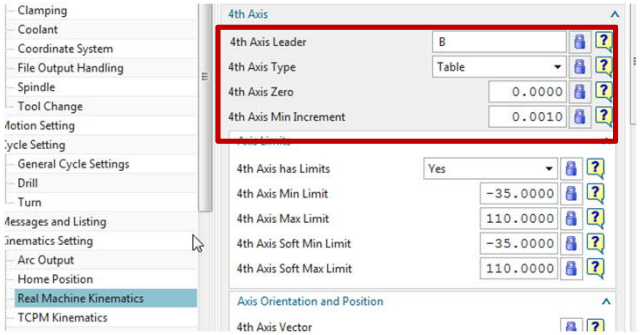


1

```
N18 G234 H1
N20 G17 G0 G90 X-245.026 Y-95.398 S2228 M3
N22 Z244.136 A90. C119.972
N24 Z221.911
N26 G94 G1 X-245.01 Y-96.202 Z220.369 F1203.
```

```
<Filename Name="HAAS_Base_mtb" Processing="true"/>
<Filename Name="Kinematic_TTBC_M35_P110_NL" Processing="true"/>
<Filename Name="Kinematic_TTAC_M50_P130_NL" Processing="false"/>
<Filename Name="HAAS_Base_service_fanuc" Processing="true"/>
```

2




```
N16 G54
N18 G234 H1
N20 G17 G0 G90 X320.398 Y-20.026 S2228 M3
N22 Z244.136 B90. C29.972
N24 Z221.911
N26 G94 G1 X321.202 Y-20.01 Z220.369 F1203.
```

3

Summary:

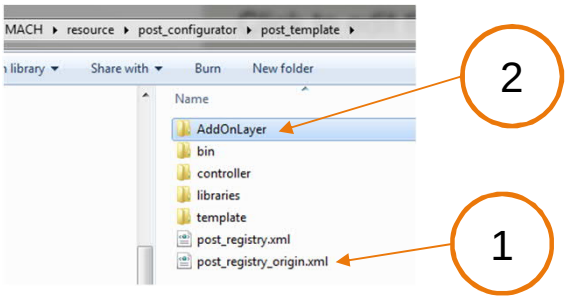
Now it‘s possible to switch the kinematics easily.

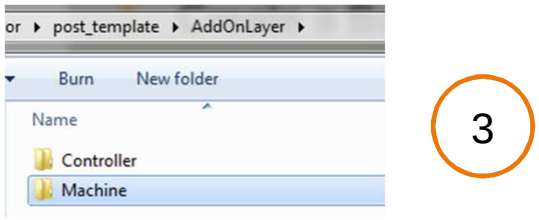
1. Postprocess an operation with TTAC kinematics file
2. In the psc-file turn off the processing of TTAC kinematics file
3. Reopen Post Configurator and postprocess again

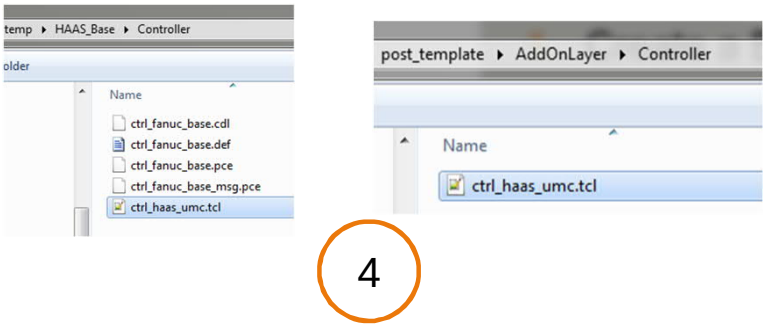
<u>Additional Background:</u>

This can be reused for each predefined kinematic definition. In next topic this will be transferred into standard environment.

Task 7
Solution 8/15 – Create structure in environment





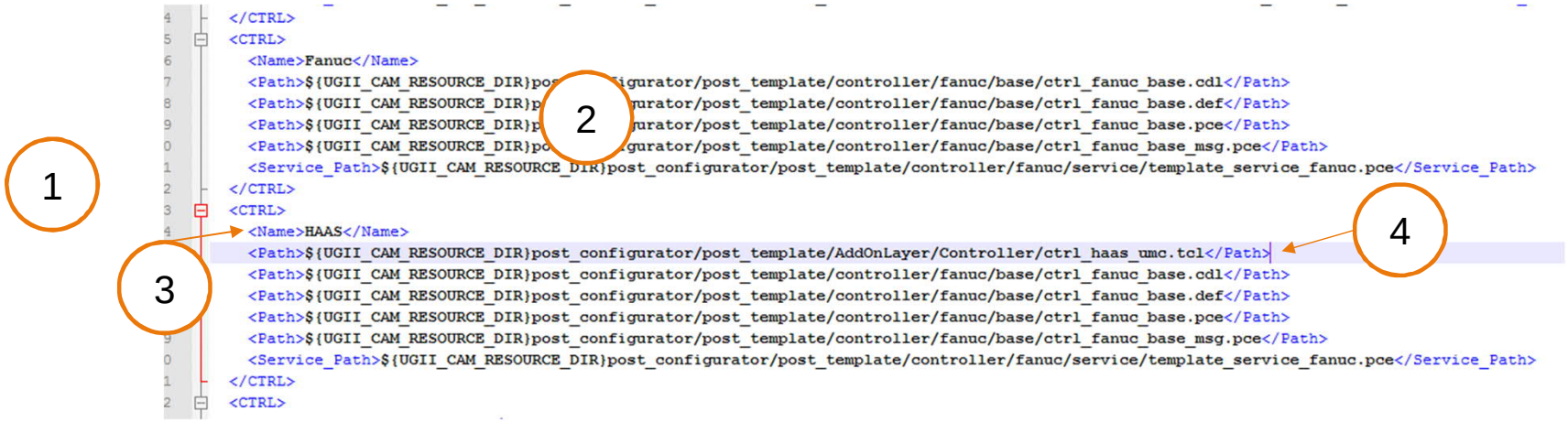


1. Create a Backup of your origin post_registry.xml
2. Create a new folder for the new created layers
3. In this folder create a sub structure, e.g. Controller, machine
4. Copy the created controller file from the temporary postprocessor into the folder

Task 7
Solution 9/15 – Create structure

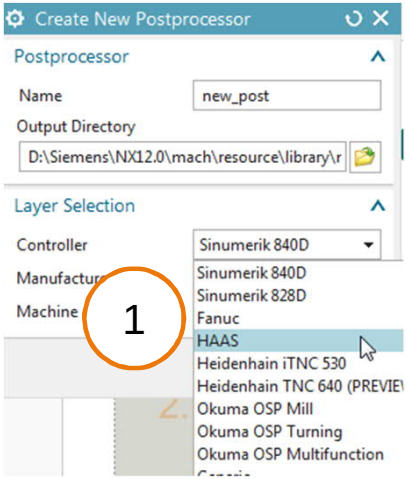
SIEMENS
Ingenuity for life

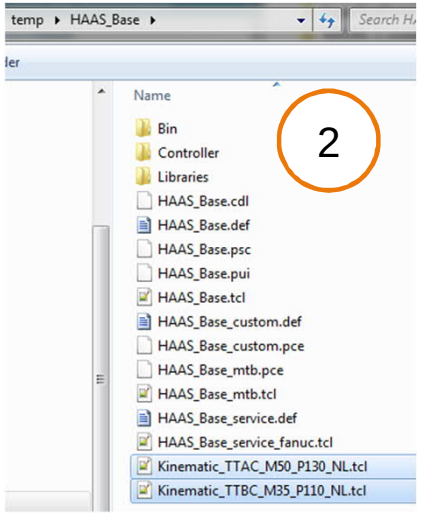
1. Open the post_registry.xml
2. Copy the existing Fanuc entry
3. Rename the copy to e.g. HAAS
4. Add the entry for the main-tcl file which will also source the Fanuc controller
5. Save the file

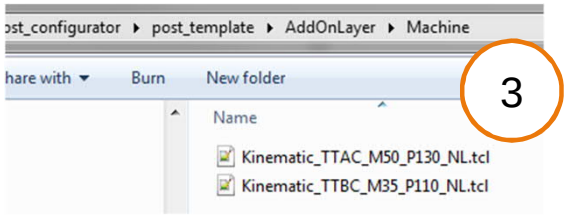


Task 7
Solution 10/15 – Create structure

SIEMENS
Ingenuity for life







1. Open Post Configurator and check that post_registry works fine and the added controller level is visible
2. Add the additional two kinematics files into the AddonLayer/Machine folder from the temporary postprocessor
3. Open the post_registry.xml to add theses files as machine layers for this HAAS controller

Task 7
Solution 11/15 – Create structure

SIEMENS
Ingenuity for life

1. Copy an existing machine level, e.g. Fanuc
2. Give it a unique name and assign the Base_CTRL (logic which files will be shown when select a controller)
3. Repeat the step for the second predefined kinematic and save the post_registry.xml

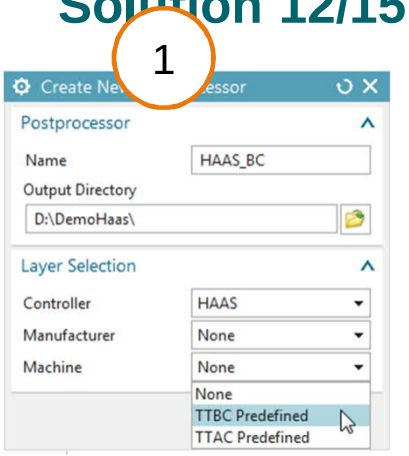
```
<MACHINE>
  <Name>Fanuc Sample Machine Level</Name>
  <Path>${UGII_CAM_RESOURCE_DIR}post_configura st_template/controller/fanuc/machine/machine_ootb_5ax_fanuc.pce</Path>
  <Base_CTRL>Fanuc</Base_CTRL>
</MACHINE>
<MACHINE>
  <Name>TTBC Predefined</Name>
  <Path>${UGII_CAM_RESOURCE_DIR}post_configurator/post_template/AddOnLayer/Machine/Kinematic_TTBC_M35_P110_NL.tcl</Path>
  <Base_CTRL>HAAS</Base_CTRL>
</MACHINE>
<MACHINE>
  <Name>TTAC Predefined</Name>
  <Path>${UGII_CAM_RESOURCE_DIR}post_configurator/post_template/AddOnLayer/Machine/Kinematic_TTAC_M50_P130_NL.tcl</Path>
  <Base_CTRL>HAAS</Base_CTRL>
</MACHINE>
```

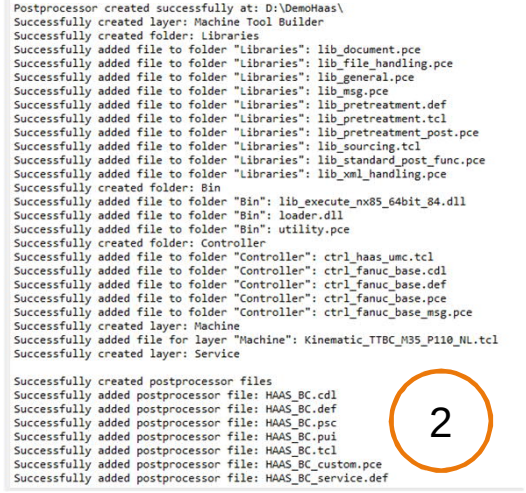
1

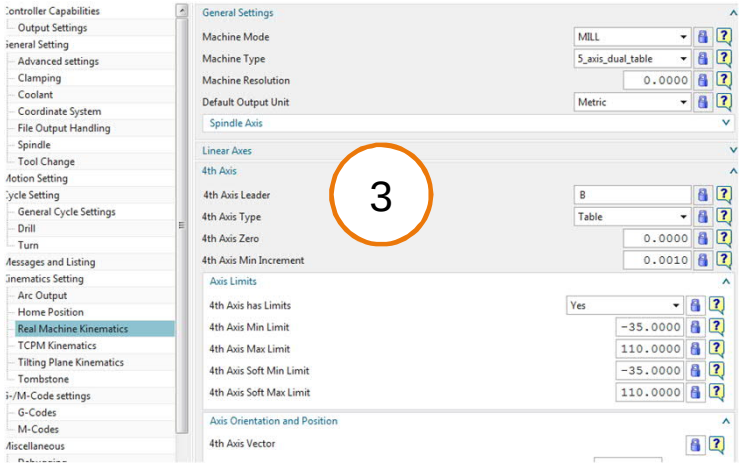
2

3

Task 7
Solution 12/15 – Create structure







1. Check that everything is fine and open the Create Postprocessor Dialog
2. Create a new Postprocessor based on this data
3. Check/ recognize that kinematics are set directly correct

<u>Additional Background:</u>

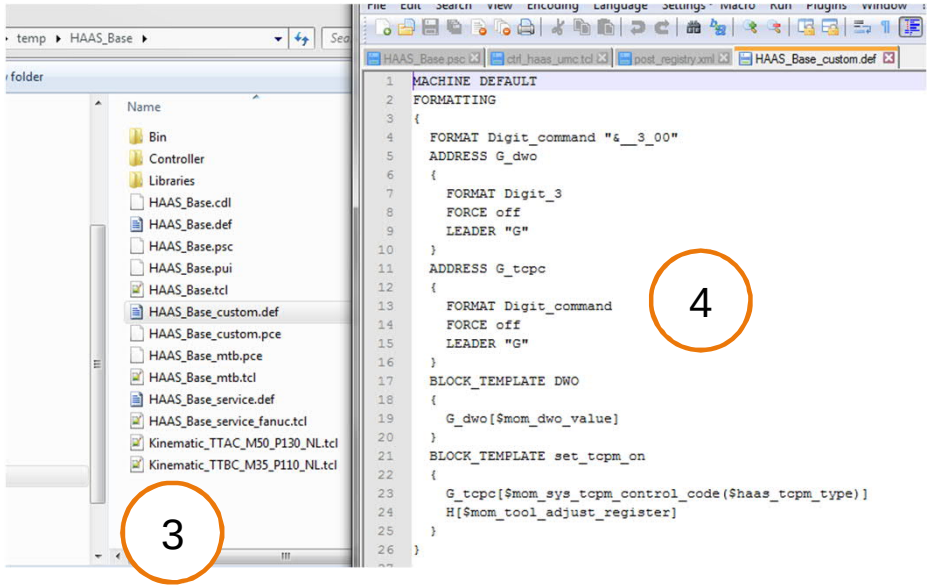
Even if a machine mdoel is loaded now, the predefined kinematics will be used. For this you can build a property which decides use machine model kinematics or not.

Task 7
Solution 13/15 – Create structure

SIEMENS
Ingenuity for life

```
********** MOM: TCL SCRIPT MESSAGE **********
User message: Error code 1745006: MOM given an invalid block name; MOM: DWO is an invalid block template name.
    while executing
"MOM_do_template DWO"
    invoked from within
"if {1} [subst $\{[subst ::buffer::${00_                ::${00_section}::code($00_tag)]\}]"
    ("foreach" body line 8)
```

1



3

4

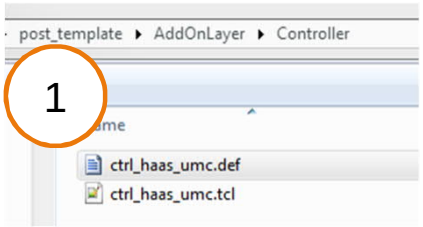
1. Select an operation and try postprocessing. You should get an error ☺.
2. Thinking…
3. Remember, we defined additional Blocktemplates and address.
4. Open the custom.def-file from the temporary postprocessor.

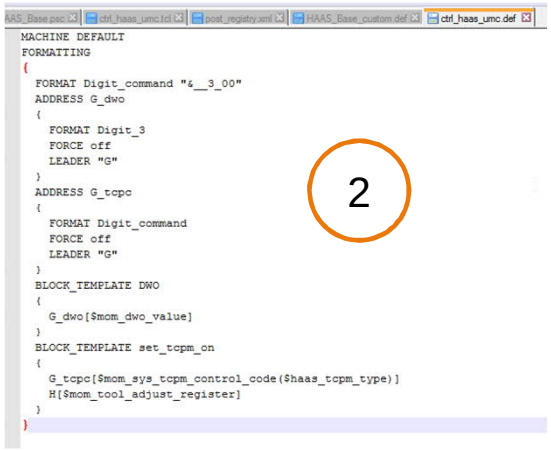
<u>Additional Background:</u>

All customization for address, Blocktemplates, Formats will be written in the xx_custom.def file of the postprocessor. To reuse it we have to copy the content in a def-file controller layer

Task 7
Solution 14/15 – Create structure

SIEMENS
Ingenuity for life





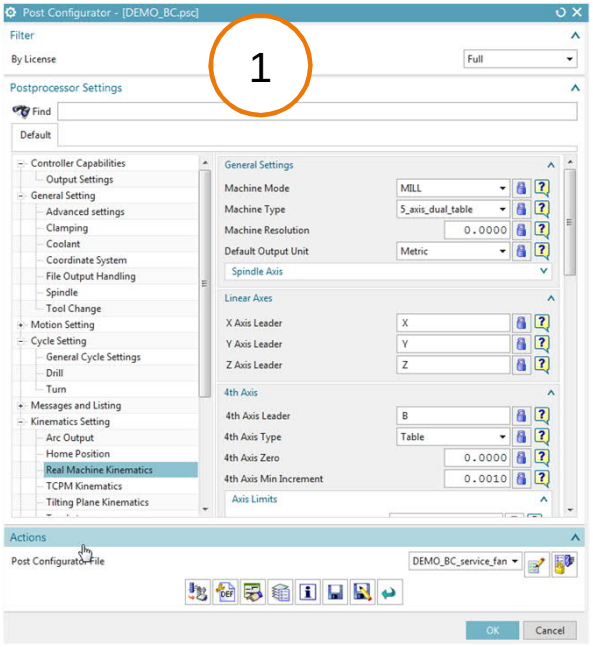
1. Create a new def-file in the AddOnLayer/Controller folder
2. Copy the content from temporary postprocessor/ custom def-file into that file
3. Link this file to the controller. Be aware of sourcing order if you overwritten Blocktemplates then the origin layer must be sourced before.

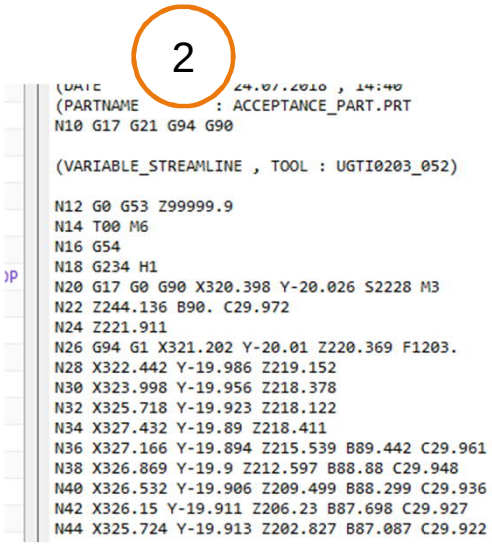
<u>Additional Background:</u>

The same is possible for cdl-files which contains UDE‘s.

Task 7
Solution 15/15 – Create structure

SIEMENS
Ingenuity for life





1. Create a new postprocessor again
2. After creation postprocess an operation.
3. Done.

Q&A

SIEMENS
Ingenuity for life

Thomas Jenensch
Product Portfolio Lead NX CAM Infrastructure
Manufacturing Engineering Software

Nonnendammallee 101 5. OG, Bauteil C
D-13629 Berlin, Germany
Tel. :+49 (30) 46777 535

thomas.jenensch@siemens.com
www.siemens.com/plm

Siemens Manufacturing Forum
www.siemens.com/plm/nxmanufacturingforum

Realize Innovation