



## **What's New in Post Builder Version 7.5**

### ***Summary***

- **Template Controller Posts** – Template posts configured and customized for three distinctive types of NC controllers (**Sinumerik 840D**, **Fanuc30i**, and **Heidenhain Conversational**) are included in the Post Builder's library.

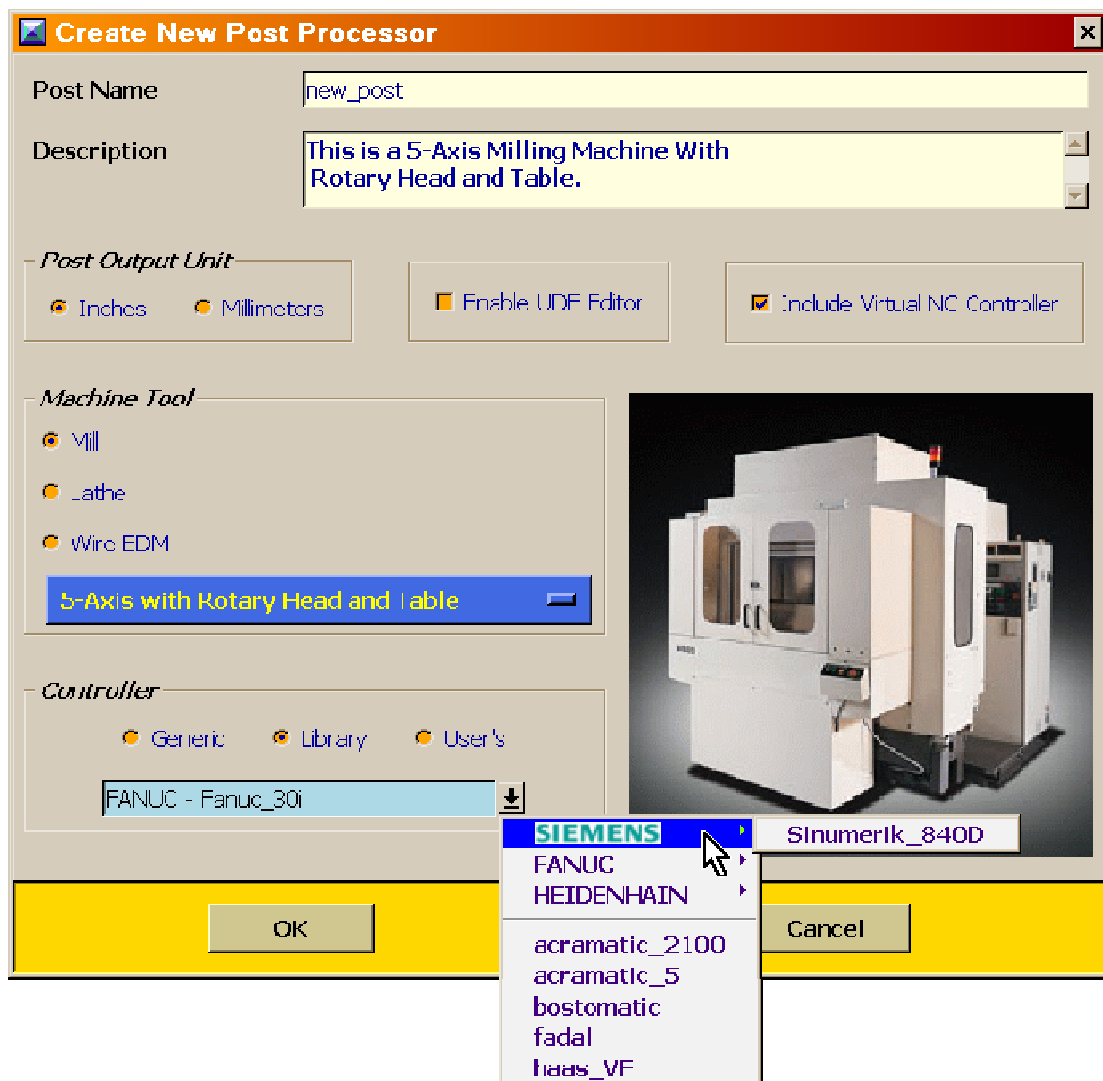
New mechanism implemented in this version of Post Builder enables the customization to the posts for different controllers to be preserved when these posts are used to create other posts or upgraded to the newer versions of Post Builder in the future.

- **Macro/Function Constructor** – New constructor enables the users to compose block templates for making calls to macro, cycle, or function in the NC program.
- **New UDE Elements** – UDE editor has been enhanced to support the new UDE elements implemented in NX/CAM.
- **Other Changes & Enhancements**

## Template Controller Posts

Template posts configured and customized according to the requirements of three major families of controllers are included in the posts library of Post Builder:

- **Siemens' Sinumerik 840D**
- **Fanuc's 30i**
- **Heidenhain's Conversational**



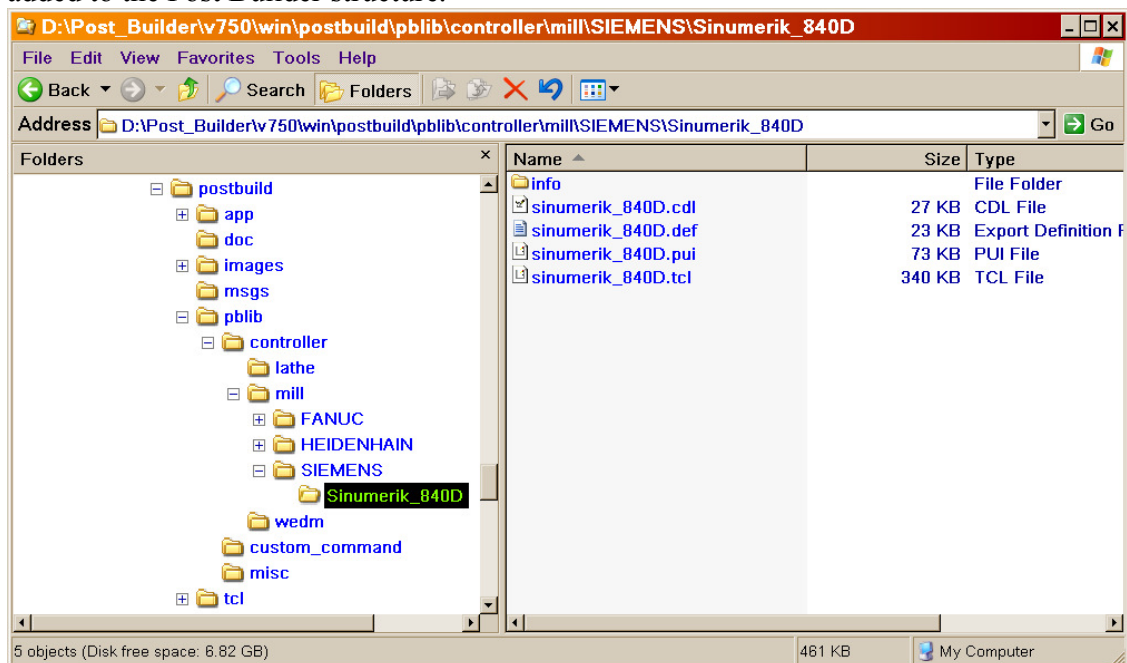
## Create New Posts Using Controller Posts

- You can create new posts by using one of these template posts.
- Each controller post will supply the elements that have been customized for the specific controller in question to the new post:

- **Presentation and interaction** (look & feel) of the Motion events (linear, circular, rapid and/or NURBS), Machine Control events and Cycles.
  - The help information for specific events or cycles of the controller will be provided. (Currently only graphical information for Sinumerik 840D is provided due to copyright consideration)
  - Parameters for the user-defined events (**UDE**) and user-defined cycles (**UDC**)
  - Addresses and Formats
  - M codes and G codes tables
- ☐ In the previous versions of Post Builder, some of these elements could be customized by editing the PUI file manually. Nevertheless, the changes will be restored when the post is upgraded to a newer version of Post Builder or used to create other posts.
- If a library post has been built with UDE enabled, the enabling switch for UDE editor will be toggled on by default. You may choose to override it. Similarly, if the post has been created with the VNC (simulation driver) enabled, the new post will also have the capability available. You can also choose to disable it.

## Add New Controller Posts to Library

- You can also introduce your own favored families of controllers to Post Builder's library. New controller posts will appear on the selection list for creating new posts.
- Additional controllers for an existing family or new families of controllers can be added to the Post Builder structure.



- Each controller post should be packaged in a separated folder.
- Post Builder also provides textual and graphical help info in the “**info**” sub-folder for the cycles of Sinumerik 840D and Heidenhain iTNC530 template posts. It will require

manual changes to the PUI files to hook up the information. This is not mandatory for a controller post.

- A controller post should be placed under the desired family folder.
- New family folders may be added under each machine type (lathe, mill & wedm) folder.
- The name of a new controller post must be manually assigned in the PUI file of the post as its “controller type” attribute:

```
# CONTROLLER TYPE START  
sinumerik_840D  
# CONTROLLER TYPE END
```

This attribute should have the same name as the post.

- A controller post can also limit its usage to specific machine types and/or axes configuration using the following 2 attributes:

```
# MACHINE TYPE START  
Mill  
# MACHINE TYPE END  
# MACHINE AXIS START  
4  
# MACHINE AXIS END
```

MACHINE TYPE	MACHINE AXIS
Mill	3, 3MT, 4, 4H, 4T, 5HH, 5TT, 5HT
Lathe	2
Wedm	2, 4

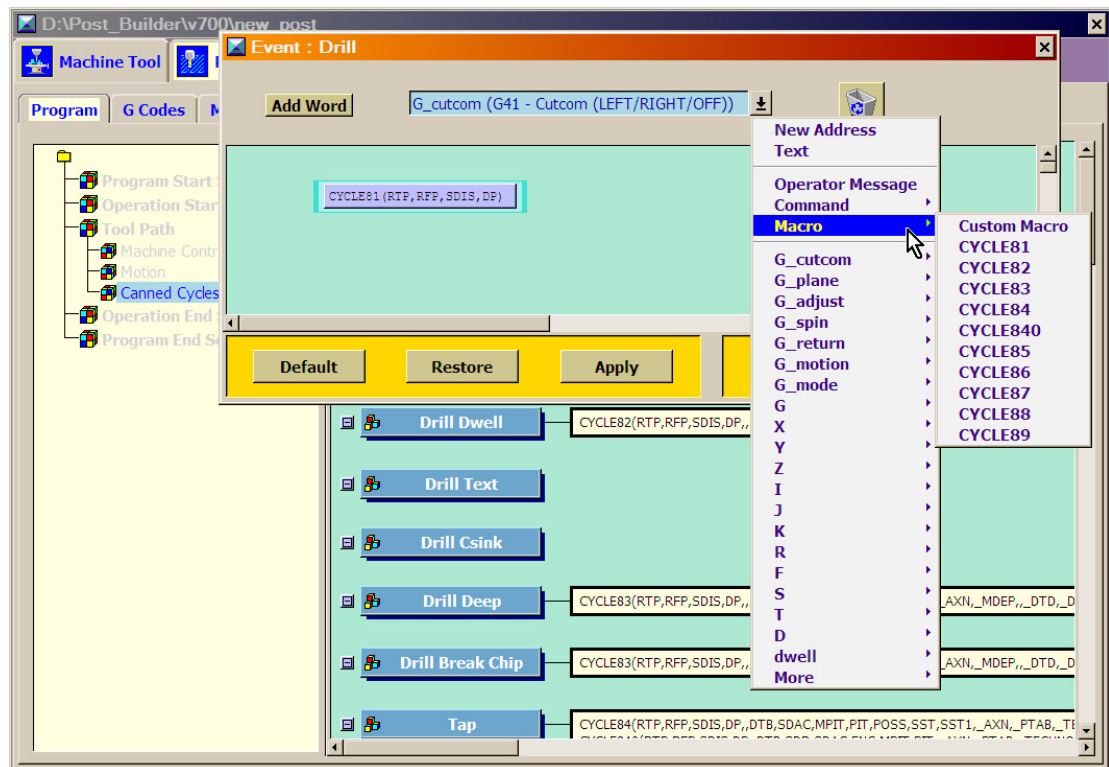
These 2 attributes are optional. A “machine axis” of “4” of a mill post indicates that the template post can be used to create posts for 3-axis, 3-axis mill-turn and both 4-axis head & 4-axis table. You can also restrict a candidate controller post to a specific kinematics configuration such as 4H or 5TT.

## Macro/Function Constructor

New building elements and mechanism have been implemented to facilitate the users to compose the posts that need to output NC codes in the form of cycle or macro calls, such as CYCLE81(...) for Sinumerik 840D or “CYCL DEF 204 Q200=...” for Heidenhain iTNC530 controller, or other high level function calls in the similar fashion.

### Construct Cycle and Macro Calls

- As an example, CYCLE81 statement can be defined for the drilling cycle of a Sinumerik 840D post.



Out of the box, the output constructs for the known hole-making cycles will be defined for the controller posts and assigned to the appropriate events. You can create additional macros of cycles as desired.

**CYCLE81**

Macro Name : CYCLE81

CYCLE81 ( 1.234, 1.234, 1.234, 1.234 )

**Macro**

Display Name CYCLE81

**Parameters List**

Separator , Comma (,)

Start Character ( Left Parenthesis (

End Character ) Right Parenthesis )

Parameter	Expression	Data Type	Integer	Decimal (.)	Fraction
RTP	\$mom_cycle_retract_to_pos(\$mom_cycle_spindle_axis)	Numeric Text	5	<input checked="" type="checkbox"/>	3
RFP	\$mom_pos(\$mom_cycle_spindle_axis)	Numeric Text	5	<input checked="" type="checkbox"/>	3
SDIS	\$mom_cycle_rapid_to	Numeric Text	5	<input checked="" type="checkbox"/>	3
DP	\$mom_cycle_feed_to_pos(\$mom_cycle_spindle_axis)	Numeric Text	5	<input checked="" type="checkbox"/>	3

Default Restore Apply OK Cancel

- Macro Name

**Macro**

Display Name CYCLE81

This option defines the name of a cycle or macro to be output.

- Format of Parameters List

**Parameters List**

Separator , Comma

Start Character ( Left Parenthesis

End Character ) Right Parenthesis

Three options in this group define the form of the arguments list for the cycle or macro call. You can choose any of these options from the pull-down list such as

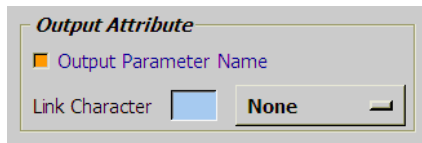
None  
Space  
Decimal .  
Comma ,  
Semicolon ;  
Colon :  
Text String

and

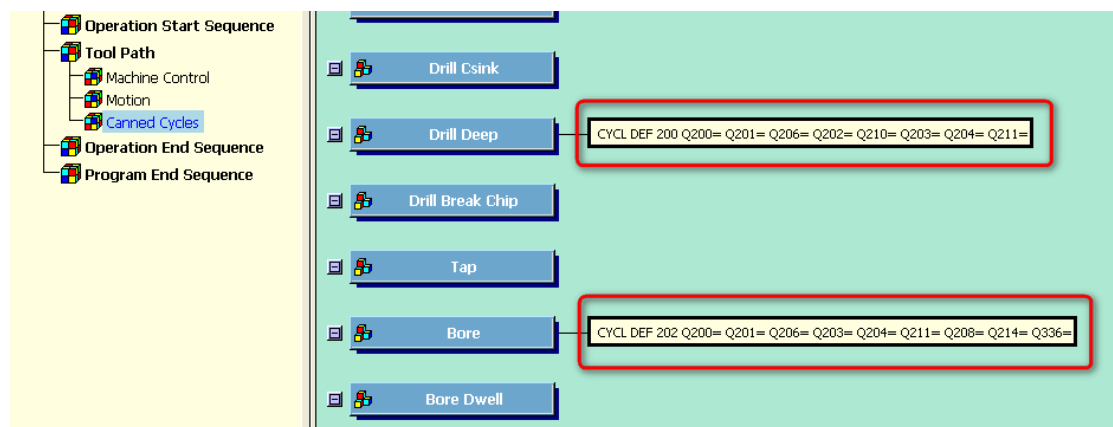
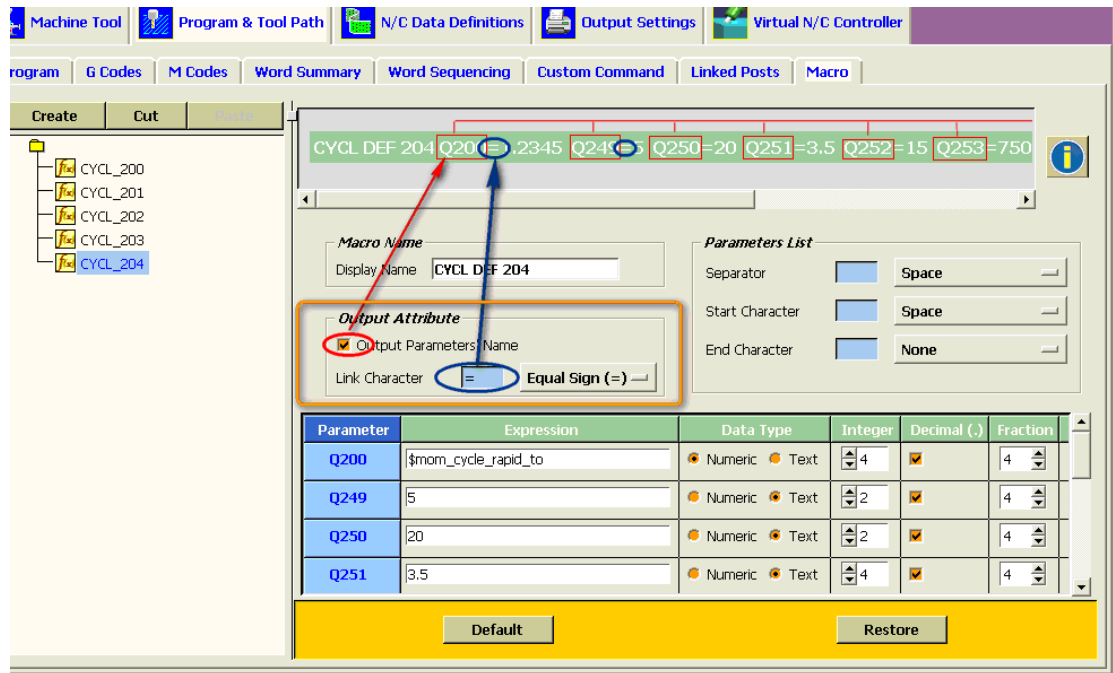
None  
Space  
Left Parenthesis (  
Right Parenthesis )  
Pound Sign #  
Asterisk \*  
Comma ,  
Semicolon ;  
Colon :  
Slash /  
Text String

option. or enter your own string by selecting “Text String”

- **Additional Output Attributes**

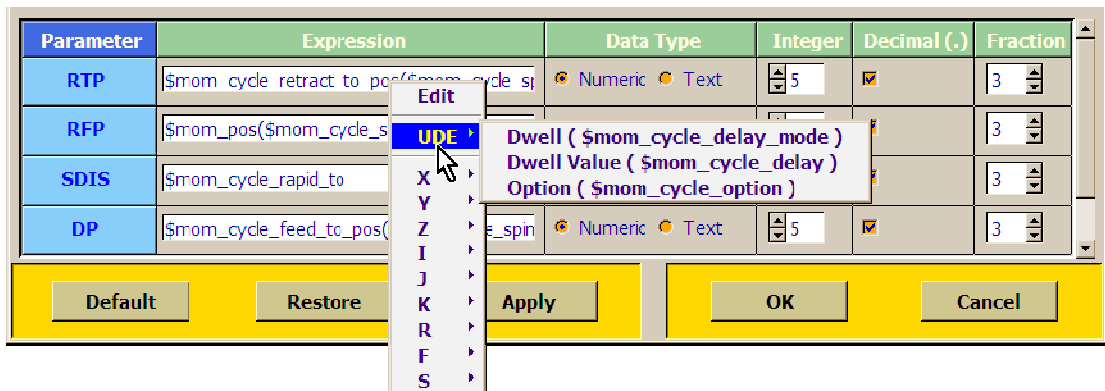


These options can be activated to output the name of parameters along with their values. As an example, these options are used to compose the cycle statements for the Heidenhain iTNC530 controller.



☐ This group of options is customizable. Since they are not applicable to the posts for Sinumerik 840D; they will not appear on the Macro constructor for the Sinumerik family of post.

- Macro Parameters

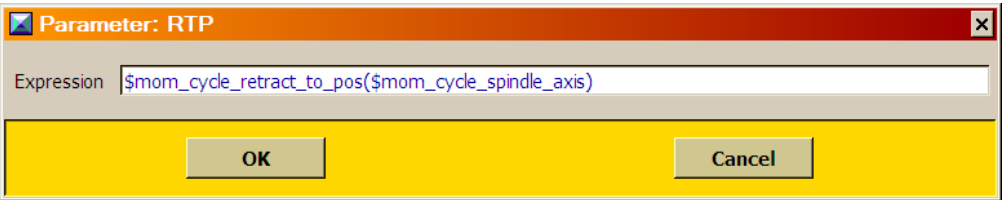


As many parameters as needed can be defined for each macro.

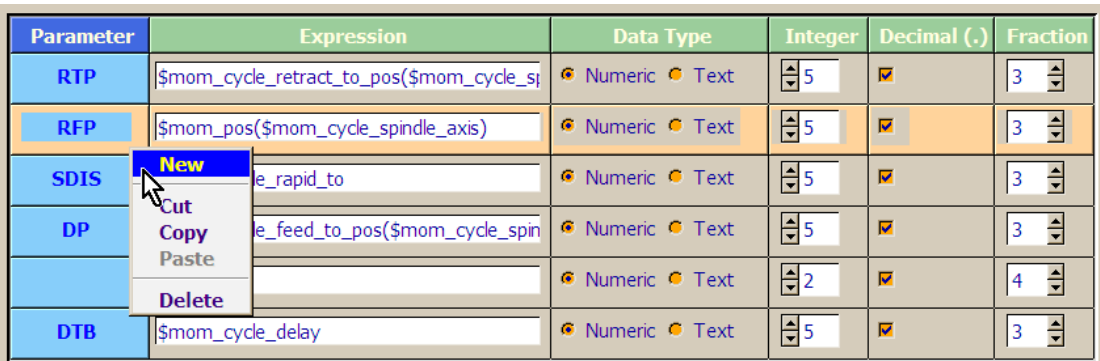
If a parameter is not to be output, its name is used only in Post Builder to display the representation of an argument.

**Expression** for each parameter can be configured using constants and global MOM & other Tcl variables. Nevertheless, the variables used should represent data of the same type (number or string) as specified by the “**Data Type**” option.

A right-mouse click on an expression will bring up a list of commonly applicable MOM variables including coordinates and UDE & UDC parameters of each individual cycle. Selecting the “**Edit**” option from the pull-down menu or double-click on the expression will display a sub-dialog allowing the user to more conveniently modify the expression.



A right-mouse click on the expression entry of the sub-dialog will also bring up the pull-down options of the common MOM variables.



A right-mouse click in the peripheral area of the parameter name will bring up the options allowing current parameter to be manipulated or a new parameter to be inserted after the current one.



A parameter without the expression will result in additional separator symbol (a “,” in this case) to be output in the call statement. This type of parameters can be defined with or without the name.

Macro Name : CYCLE82

CYCLE82 ( 1.234, 1.234, 1.234, 1.234, , 1.234 )

**Macro**  
Display Name : CYCLE82

**Parameters List**  
 Separator : Comma (,)   
 Start Character : Left Parenthesis (   
 End Character : Right Parenthesis )

Parameter	Expression	Data Type	Integer	Decimal (.)	Fraction
RTP	\$mom_cycle_retract_to_pos(\$mom_cycle_s	Numeric Text	5	<input checked="" type="checkbox"/>	3
RFP	\$mom_pos(\$mom_cycle_spindle_axis)	Numeric Text	5	<input checked="" type="checkbox"/>	3
SDIS	\$mom_cycle_rapid_to	Numeric Text	5	<input checked="" type="checkbox"/>	3
DP	\$mom_cycle_feed_to_pos(\$mom_cycle_spin	Numeric Text	5	<input checked="" type="checkbox"/>	3
		Numeric Text	2	<input checked="" type="checkbox"/>	4
DTB	\$mom_cycle_delay	Numeric Text	5	<input checked="" type="checkbox"/>	3

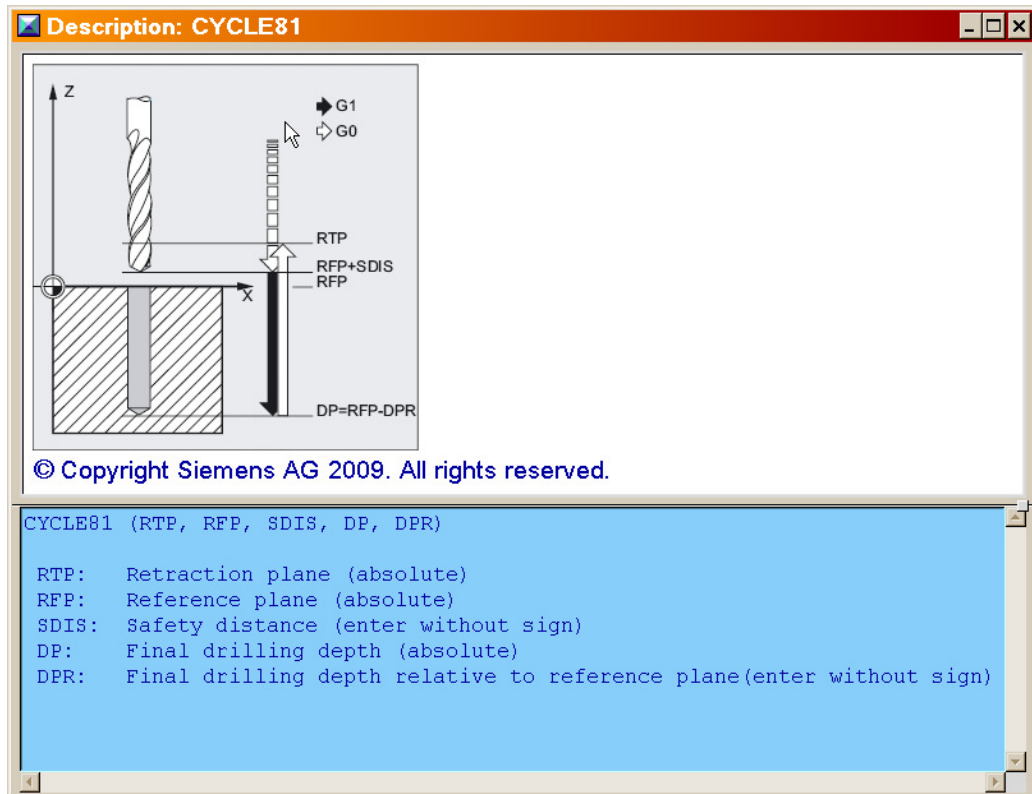
Default Restore Apply OK Cancel

- **Help Info**



Textual and/or graphical information, when permitted, for the known cycles of a particular controller may be provided. By clicking the button, you can display the info page for the macro or cycle in question.

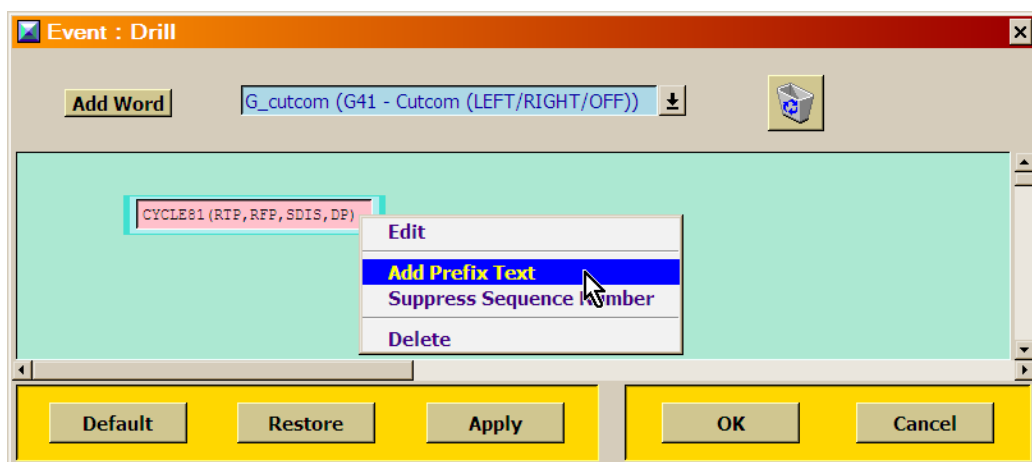
Info page for CYCLE81 of the Sinumerik 840D controller is shown below as an example:

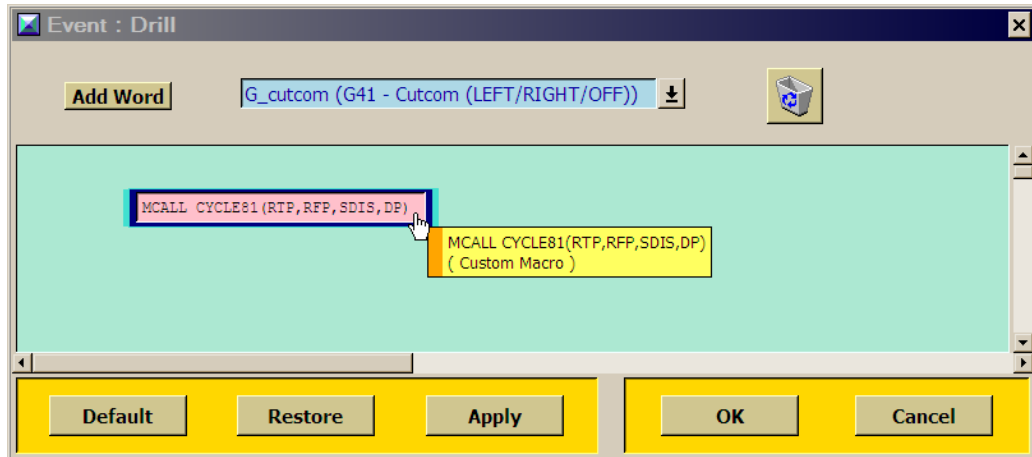
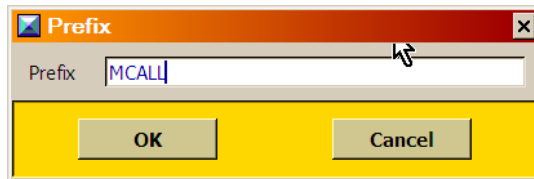


- Any copyrighted material for the help information will only be used when proper permission has been granted.

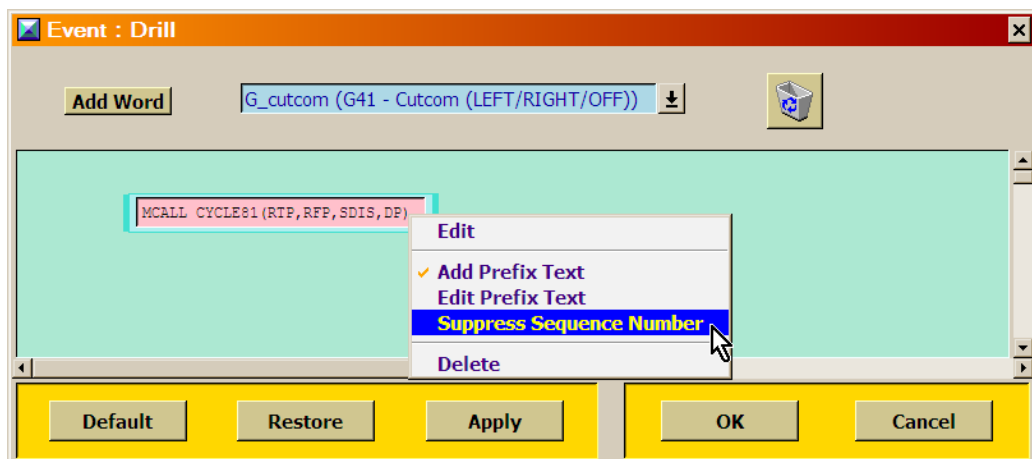
## Use and Manage Macros

- When a cycle or macro is used in a post, prefix string can be assigned and output with the call. This is illustrated in the following figures:

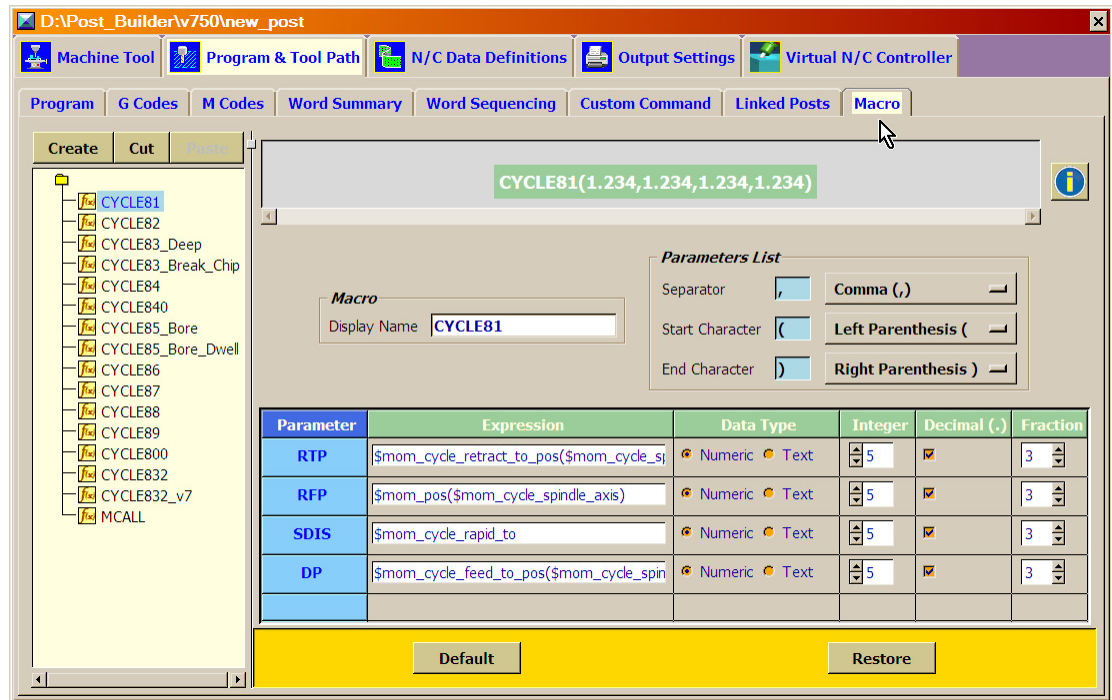




- You can also choose to suppress the sequence number from being output for a particular cycle or macro call at the time of post processing.



- All macros or cycles for a post are managed on the page below:



- A macro object can be copied to create a new one. This allows you to create a new macro with slight variations from an existing one or to output calls to the same macro with different arguments such as omitting parameters under certain conditions.

## New UDE Elements

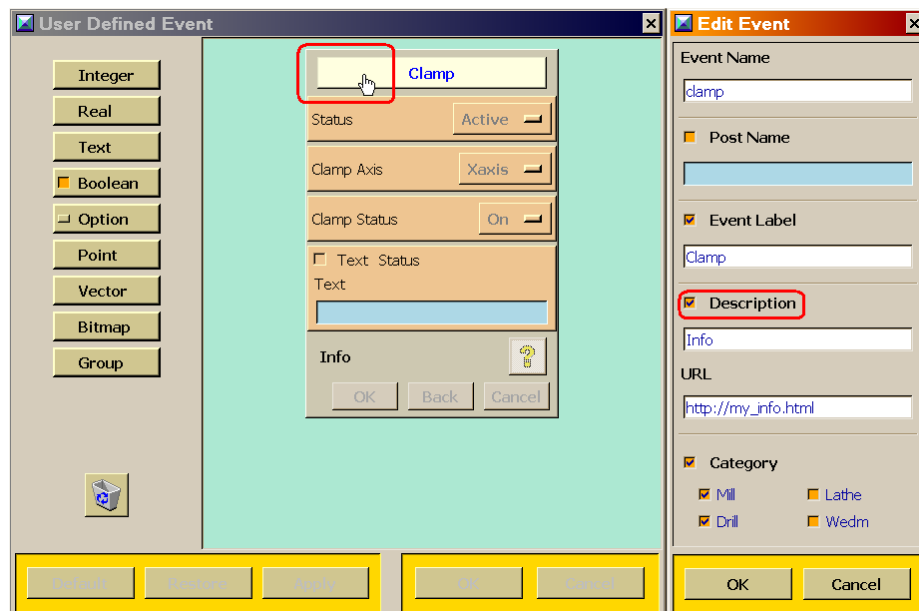
This version of Post Builder has been enhanced to support the new elements implemented in NX7.5 for the UDE (User Defined Event) mechanism. Three new elements are added to the UDE editor:

- **Context Sensitive Help** for event
- **Bitmap** (legend) parameter
- **Grouping** elements

New elements are available for both machine control events and user-defined cycles.


### Context Sensitive Help

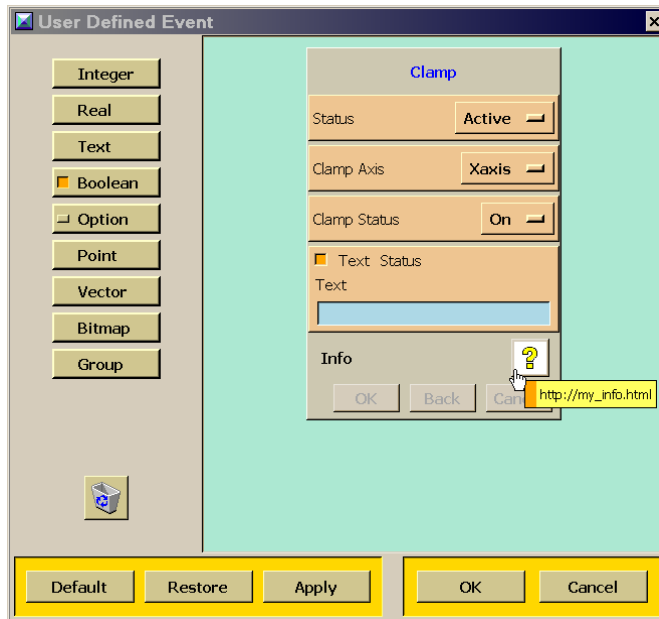
- The context sensitive help item can be added to an event. The “**Description**” toggle button on the event’s parameters dialog can be switched on to activate the help info for the event.



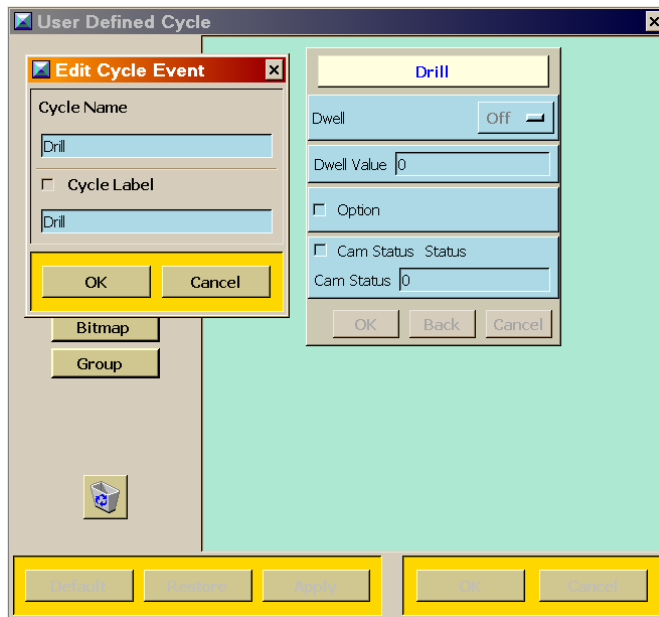
- Both label and URL must be specified.
- A legitimate URL format must be used.

Post Builder will display an error message if any of the rules has been violated.

- A question mark  icon will appear on the event dialog. You can exam the URL by traversing the mouse pointer to the icon.

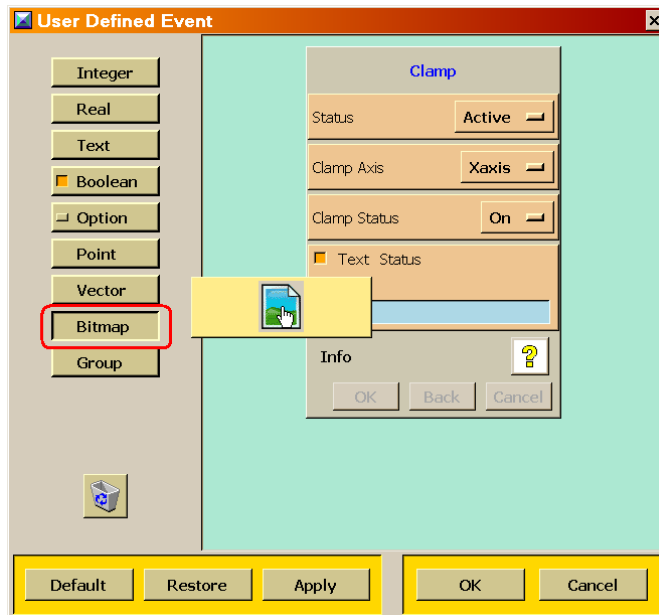


- This item can be removed by editing the event's parameters again and toggling off the "Description" switch.
- Due to the restriction in NX7.5, the help info element is not available for either system defined or user defined hole-making cycles.

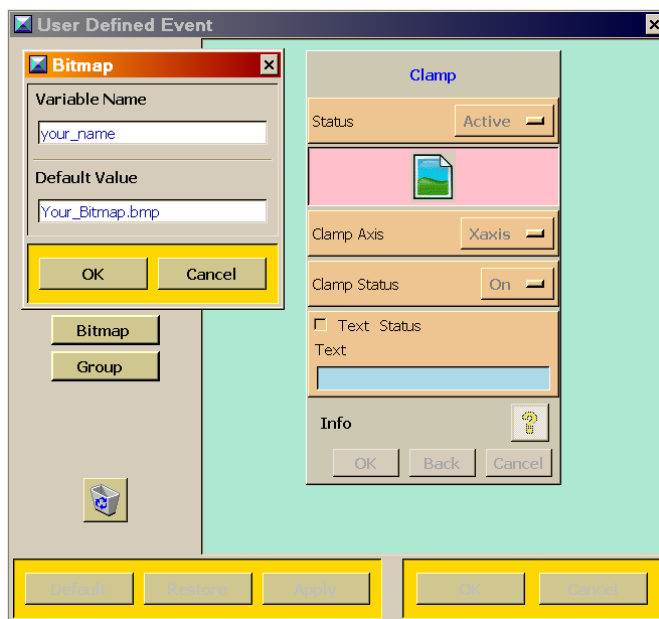



## Bitmap

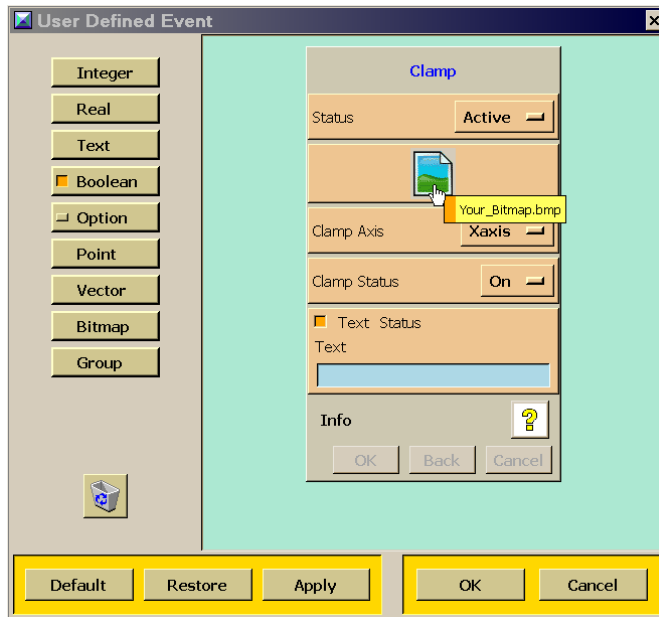
- Bitmap (legend) parameters can be added to an event by dragging the "Bitmap" button and dropping it to the desired location in the simulated event dialog.



- You will be prompted to specify the variable name and bitmap file name for this item.

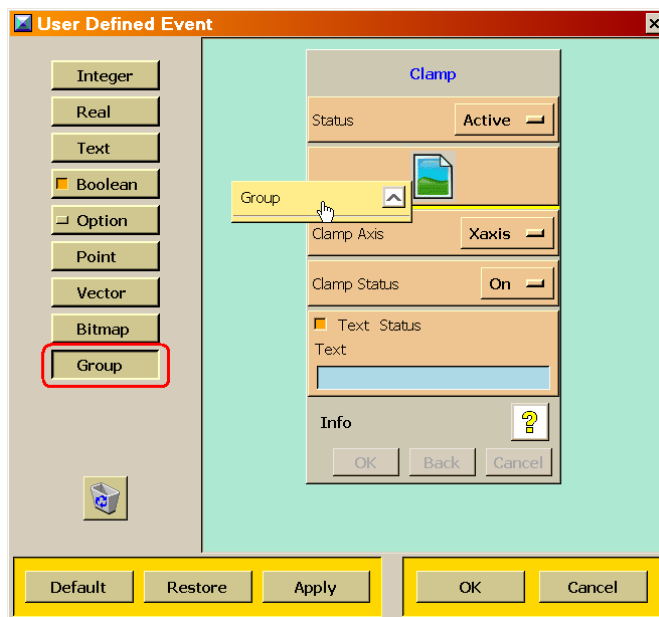


- The file name must be of format “**.bmp**”; otherwise Post Builder will issue an error message.
- When the item is in place, you may traverse the mouse pointer to the image  icon to verify the bitmap file name.



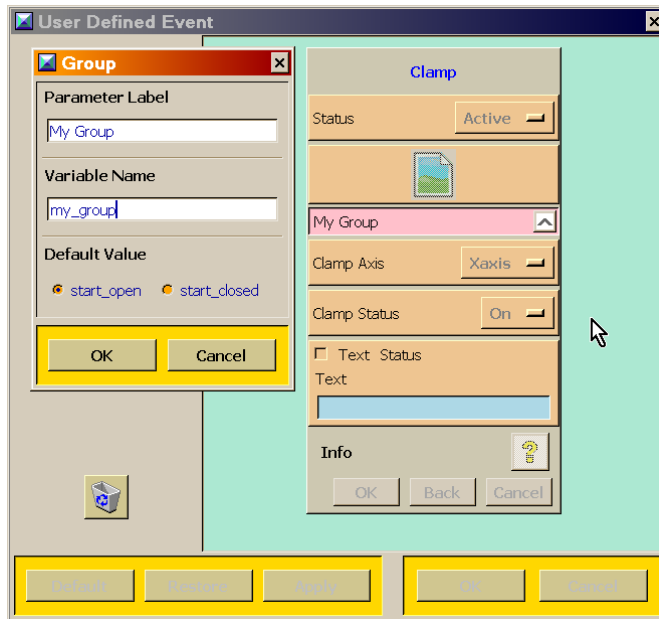
## Grouping



- The grouping elements can be added to the event dialog by pressing the “**Group**” button. You can drag the representative image to the desired location on the dialog.

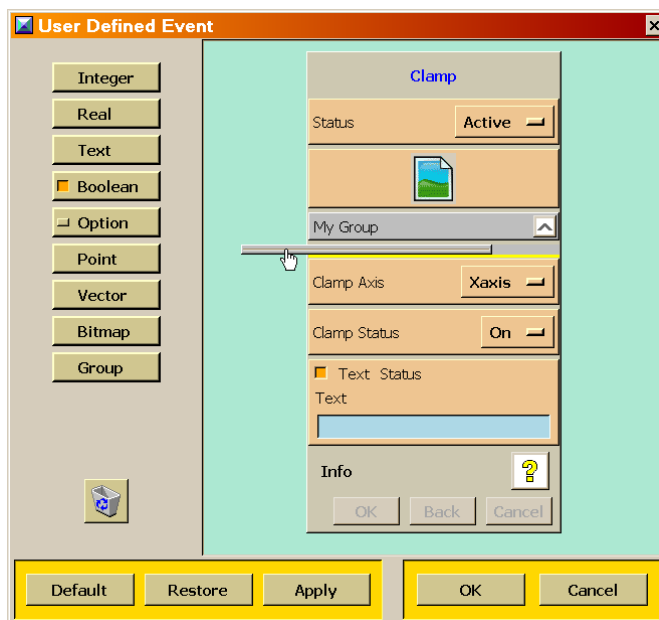


- You will be prompted to specify the label, variable name, and the default value (initial state) of the grouping elements.

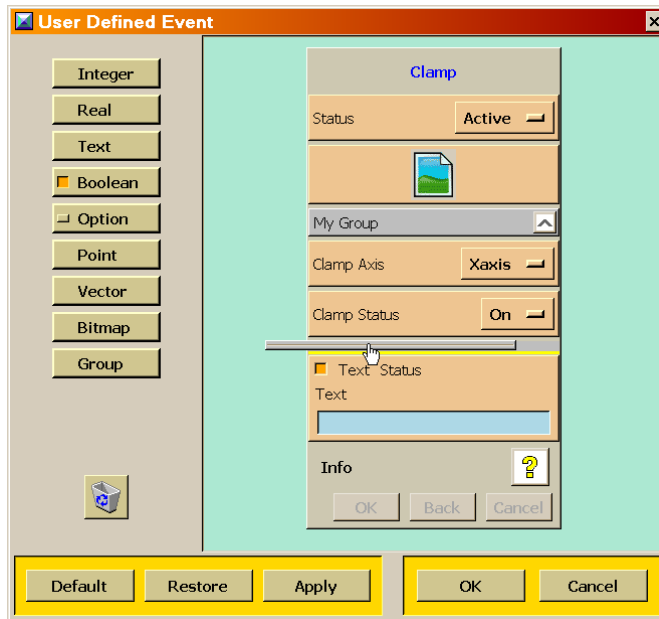





- The parameter label can be left blank. No title for the group will be displayed.
- The initial state is indicated by either “open”  or “close”  icon at the right end of the group item.
- When a grouping parameter is defined, 2 elements representing *group-start* and *group-end* will be added to the simulated event dialog.



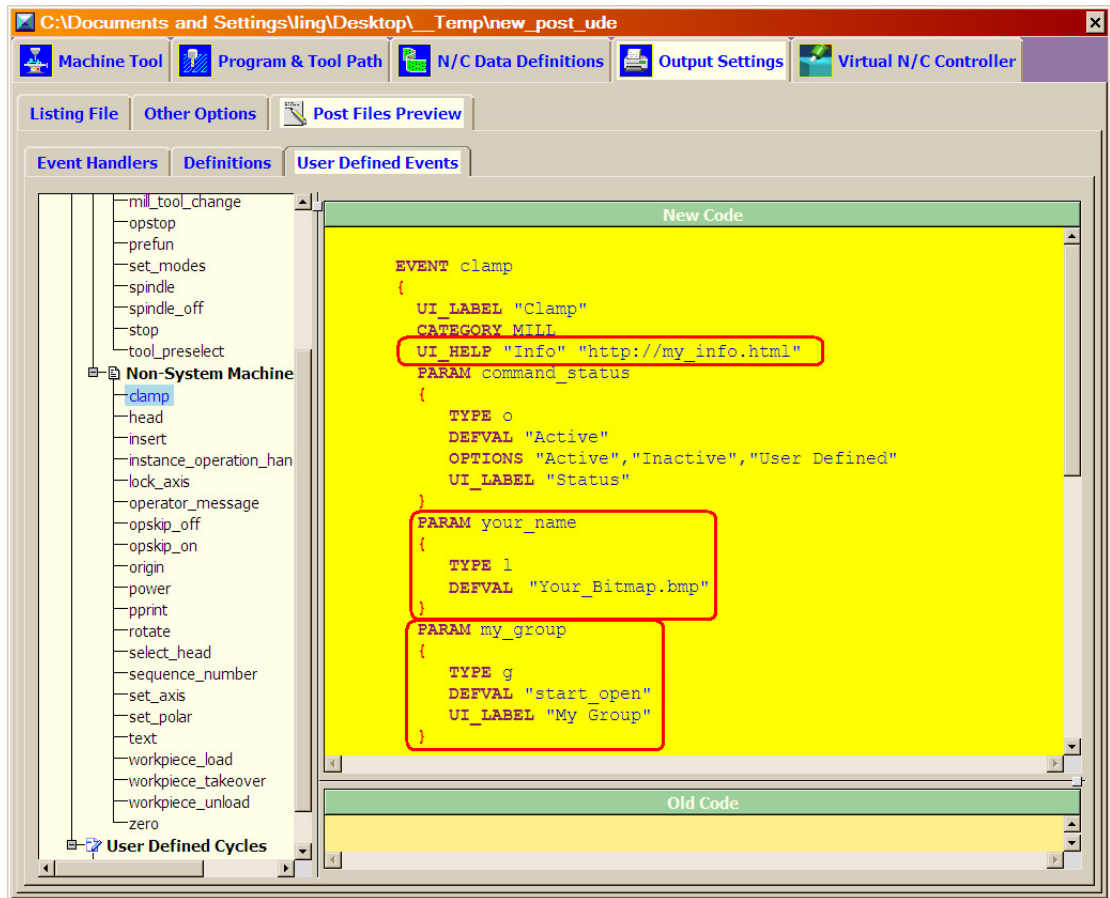
You can then drag the group-end element to a desired location on the dialog to form the group. The group-start element can also be repositioned afterward.



- Post Builder will not simulate the “open” and “close” actions of the groups on the dialog.
- The group-end element can not be edited. It can be deleted by dragging it to the trash can  icon.
- When either a group-start or a group-end is deleted, its associated element is also deleted.
- Multiple groups can be defined on an event dialog.
- A grouping item can not be added (nested) inside another group. Warning message will be displayed.

### Preview UDE Definitions

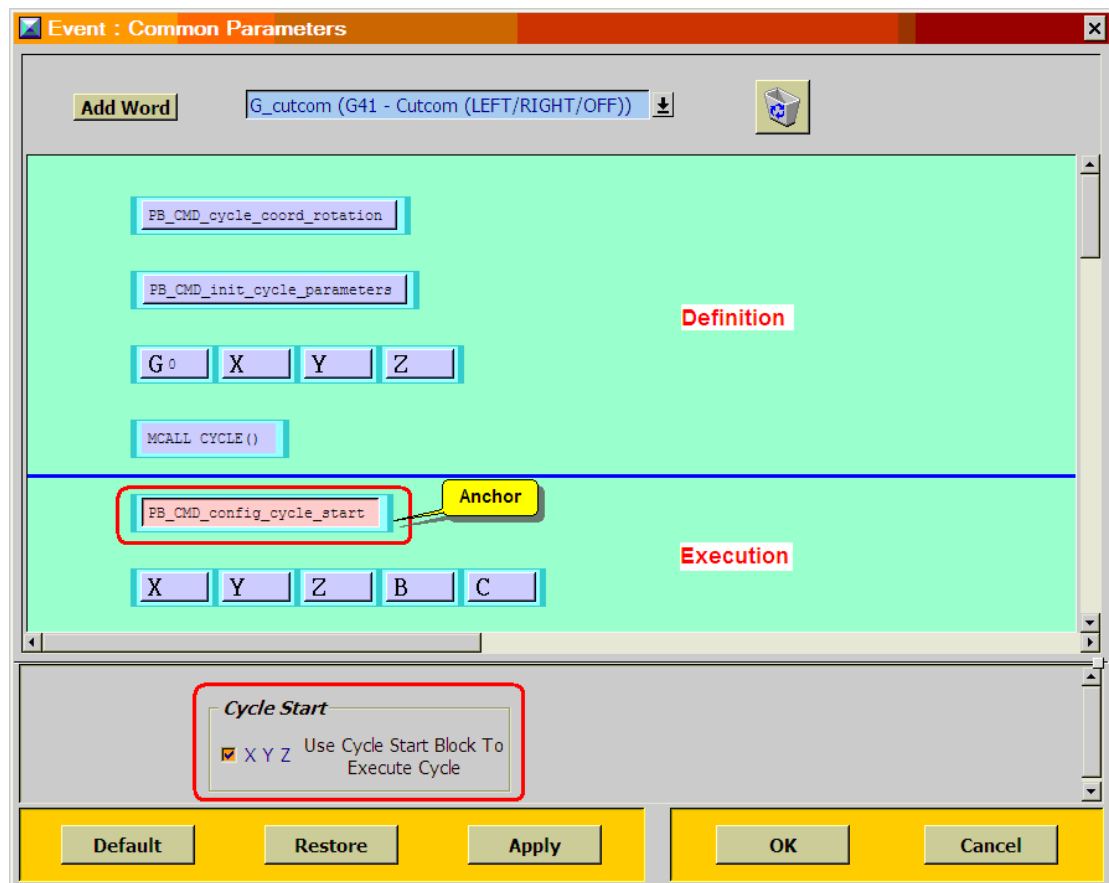
- As with the existing functionality of Post Builder, definitions of the new UDE elements, to be output to a CDL file, can be previewed as depicted in the figure below:



## Other Changes & Enhancements

### Enhanced Configuration for Canned Cycles

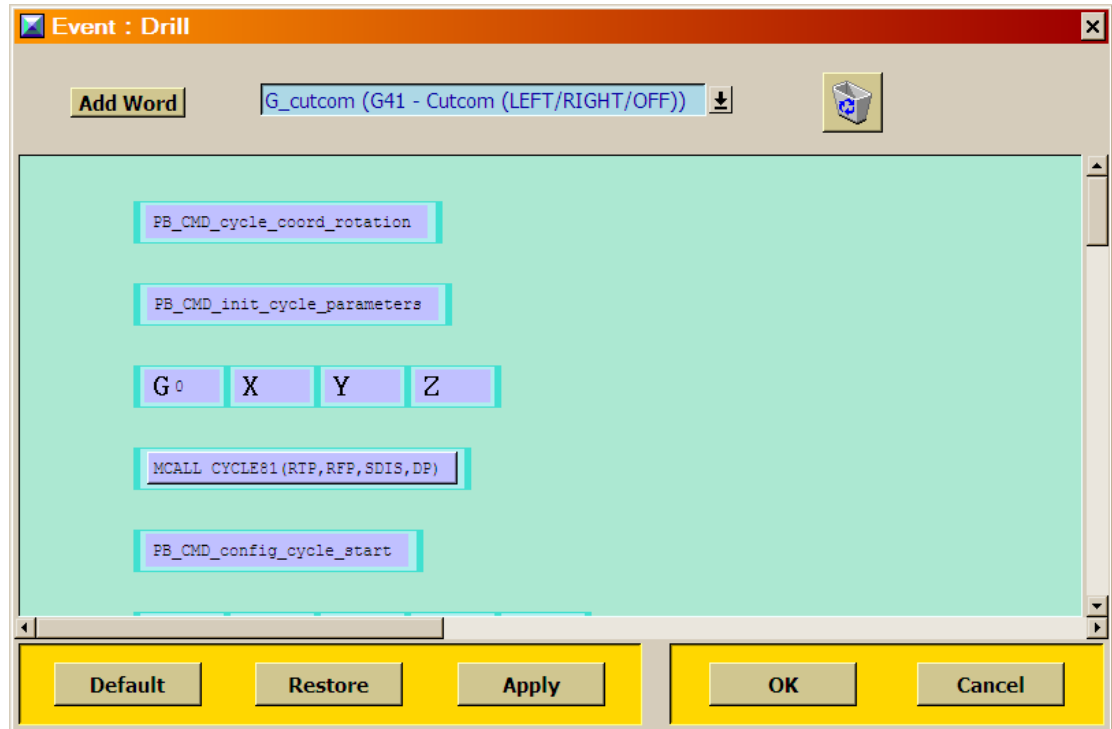
- The configuration of “Cycle Start” handling for the canned cycles has been enhanced. New mechanism removed the restriction that the cycle-start element can only be a special block template generated by Post Builder. It not only allows the cycle-start block (BLOCK\_TEMPLATE **post\_startblk**) to be configured in the definition file of a post, but also allows the “block” to be defined (in the PUI file) using a designated custom command (**PB\_CMD\_config\_cycle\_start**) instead.
- When a controller’s cycles are to be carried out via the cycle-start action, the pre-defined cycle-start block (post\_startblk) or the assigned custom command (PB\_CMD\_config\_cycle\_start) will be automatically inserted to the output window of the cycle’s common parameters as the **anchor** element. You can then add other blocks or custom commands after this anchor element to be processed upon the execution of cycles for each hole.



- Use of PB\_CMD\_config\_cycle\_start command allows you to adjust and configure the execution of cycles according to different conditions or cycle types.

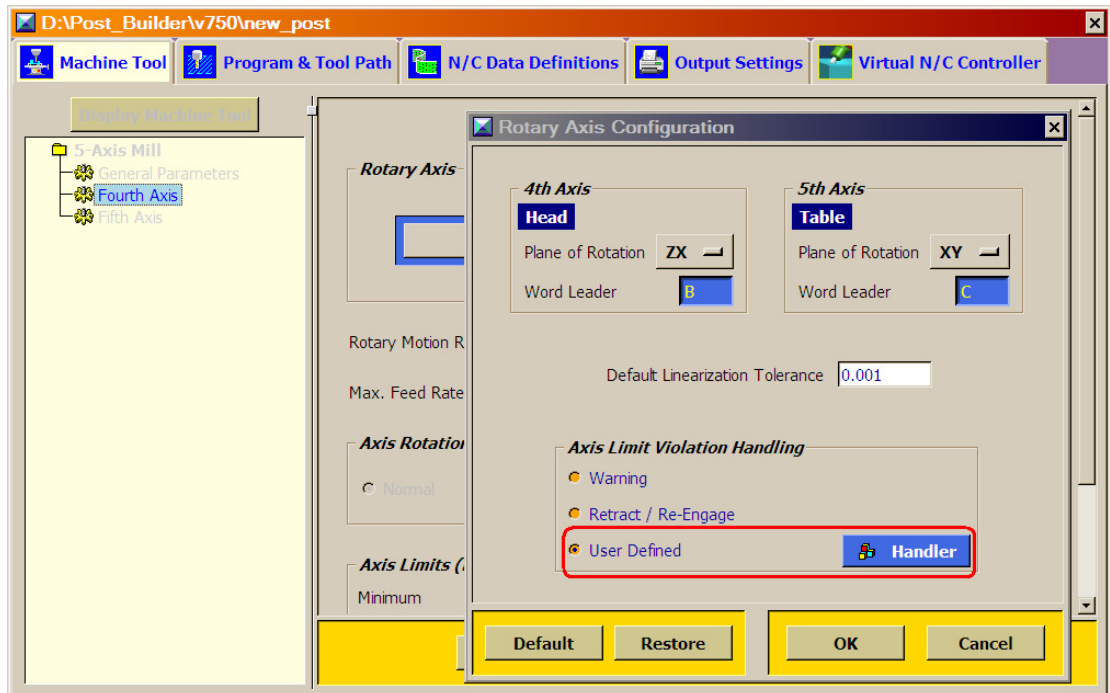
### Enhanced Display of Custom Commands, Macros & Operator Messages

- Instead of only 4 characters in all previous versions of Post Builder, the name of custom commands , the function signature of macros and the string of operator messages will be displayed in full in the event's output window to enhance the readability.

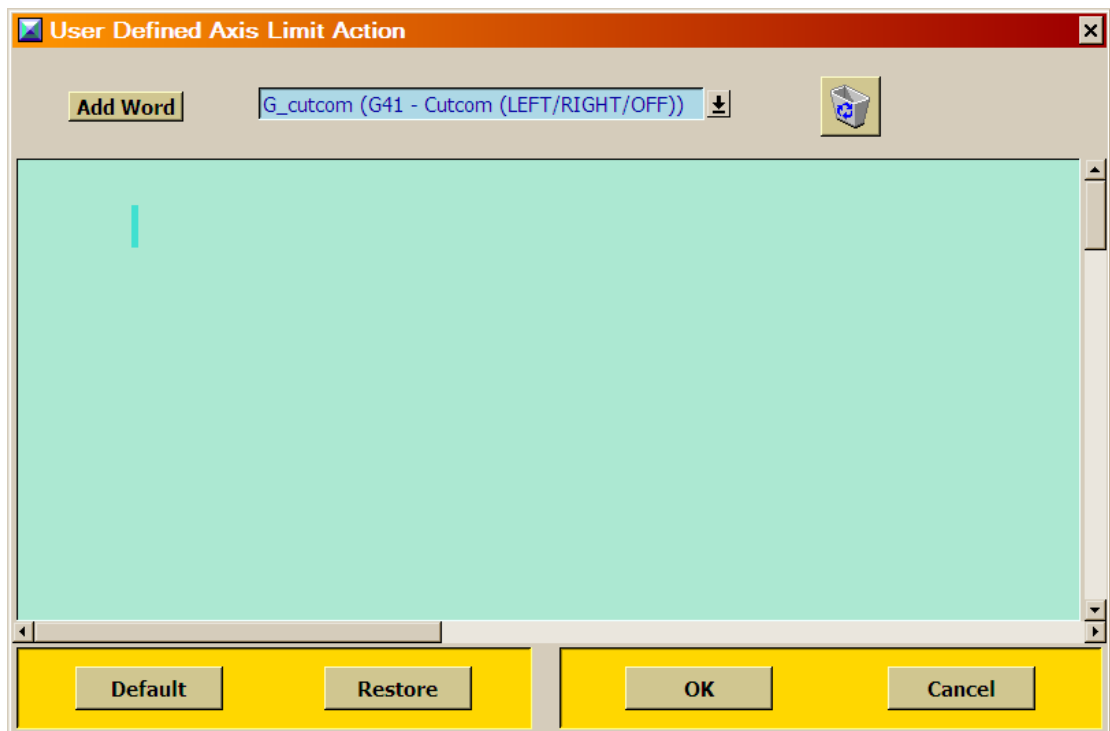


### User Defined Handler for Rotary Axis Limits Violation

- New option “**User Defined**” is added for handling the rotary axis limits violation. You will be able to construct the desired procedures and outputs for handling the situation when it occurs.



Clicking the “**Handler**” button, you can compose the procedures and output as the same way as for other events.



### **vnc\_base\_v750\_tcl.txt**

- New version of the encrypted VNC base file is included in this release of Post Builder. This file will also be released with NX7.5.

- Machine tool simulation drivers created with this version of Post Builder will require this VNC base file to work. If you are using an older version of UG/NX, you need to copy this file from the POSTBUILD directory to the desired **UGII\_CAM\_POST\_DIR** directory.

## **New Custom Commands**

### **1. PB\_CMD\_activate\_Fanuc\_turbo\_mode**

This custom command can be imported from POSTBUILD/pplib/custom\_command/**pb\_cmd\_activate\_turbo\_mode.tcl** and added to the **Start of Program** event marker to enable the turbo post-process mode. You can use this command with fixed or multi-axis mill posts. Multi-axis turbo mode is only available from NX7.0 and on.

As the name implies, this function is only suitable for the FANUC style of controllers. This custom command was first released in the version 7.0 of Post Builder.

## **Legacy Posts Conversion**

### ***Pre V3.4 Mill Posts***

#### **Preserving New IKS Parameters**

In Post Builder version 3.3 or earlier, custom commands such as “**PB\_CMD\_init\_new\_iks**” were used to enable the new IKS functionality. Parameters specified in any of these custom commands will be extracted and preserved with the posts when saved in this version of Post Builder. Interactively, these parameters will be presented in the appropriate dialogs when a post is open. However, the parameters will not be preserved, if the new IKS service was not enabled (**mom\_kin\_iks\_usage** = 0) in the custom command. Default values for the new IKS parameters will then be presented instead.

These legacy custom commands will no longer be needed but will be retained with the posts only for your reference. You may delete them manually.

#### **Swapping Rotary Axes of Dual-Head Posts**

The assignments of rotary axes in all **dual-head 5-axis mill** posts created before this version of Post Builder were intentionally switched to accommodate certain short comings with the legacy inverse kinematics solver. This erroneous setting will be corrected when a post is saved with this version of Post Builder.

Kinematic parameters for the 4<sup>th</sup> rotary axis will be swapped with that of the 5<sup>th</sup> axis.

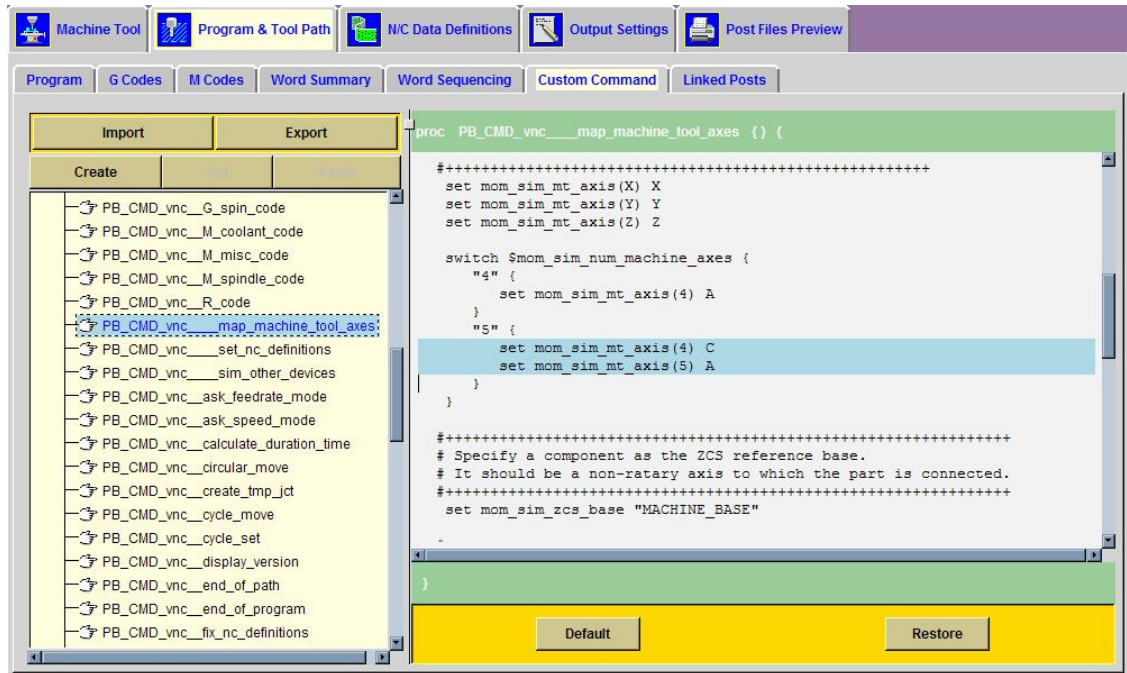
#### **PB\_CMD\_revert\_dual\_head\_kin\_vars**

This custom command will be provided for all 5-axis mill posts and used to swap the kinematic parameters of the dual rotary heads. This custom command should also be called whenever the new IKS is disabled.

#### **VNC of Dual Rotary Heads**

Prior to **NX4**, the NC axes assignment in a **VNC** file is done in a custom command. There’s no easy way to carry out an automatic conversion. You will have to edit the custom command manually.





## Legacy Posts Using ugvadvkins.dll

New IKS service will be disabled for dual-head mill posts using “**ugvadvkins**” shared library. The kinematic parameters assigned for the dual rotary heads will be swapped as described above.

## Pre V3.3 Lathe Posts

Custom command **PB\_CMD\_output\_spindle** used by any legacy lathe posts to output a block for setting the maximum spindle speed needs to be modified as follows:

```

#=====
proc PB_CMD_output_spindle {} {
#=====
  global mom_spindle_mode
  global spindle_is_out
  global mom_spindle_maximum_rpm

  if {[info exists spindle_is_out]} {

    if {$mom_spindle_mode == "RPM"} {
      MOM_force once M_spindle S G_spin
      MOM_do_template spindle_rpm
    } elseif {$mom_spindle_mode == "SFM" || $mom_spindle_mode == "SMM"} {
      MOM_force once M_spindle S G G_spin
    }

    # The following line has been changed.
  }
}

```

```

        if {$mom_spindle_maximum_rpm > 0} {
            MOM_do_template spindle_max_rpm
        }

        MOM_do_template spindle_css
    }
    set spindle_is_out 1
}
}

```

## To Run ISV



To use the Machine Tool Drivers (MTD) built with this version of Post Builder in ISV before NX3.0, you need to copy "**mom\_source.dll**" (Windows users only) from \POSTBUILD directory to \$UGII\_CAM\_AUXILIARY\_DIR.

All legacy Machine Tool Drivers will need to be resaved with the new version Post Builder.

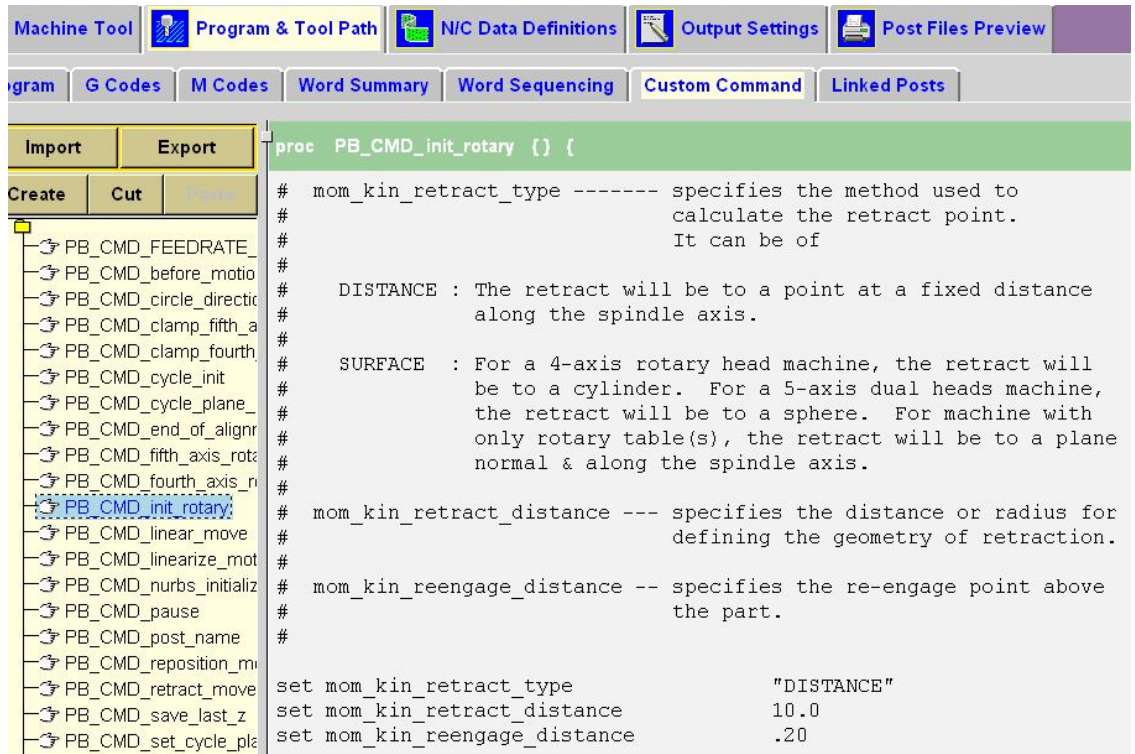
You need to copy "**vnc\_base\_v340\_tcl.txt**" in \POSTBUILD directory to your \$UGII\_CAM\_POST\_DIR to use the MTDs created with this version of Post Builder.

## Pre V3.1 Multi-Axis Posts

The option of the **Axis Limit Violation Handling** for a rotary axis specifies how the postprocessor behaves when the system reaches a rotary axis limit. The option can be of

- **Warning** means that the postprocessor outputs a message to the warning file, but takes no action to correct the axis limit violation.
- **Retract/Re-engage** means that the postprocessor generates N/C codes that enable machine-tool to retract to clearance geometry (plane, cylinder or sphere), reposition the rotary axis to a valid position if possible, re-engage to a specified distance above the part, feed to the previous position where it retracted from and then continue moving to the programmed destination. The feed rate for the retraction will be at the retract feed rate, the feed rate for the move from the clearance back to the re-engage point will be at the approach feed rate and the move from the re-engage point into the part will be at the engage feed rate.

Parameters for this control are specified in the custom command **PB\_CMD\_init\_rotary** under Program & Tool Path -> Custom Command. As depicted in the graphic below, you can modify the variables **mom\_kin\_retract\_type**, **mom\_kin\_retract\_distance** and **mom\_kin\_reengage\_distance** to configure how this functionality behaves.



The parameter **mom\_kin\_retract\_plane** used in previous releases of Post Builder has been replaced by **mom\_kin\_retract\_distance**. **PLANE** used as one of the options of **mom\_kin\_retract\_type** in some pre-release versions of v3.1 has been replaced by **SURFACE**. After you have saved your legacy posts in Post Builder v3.1, unless you make changes in the custom command **PB\_CMD\_init\_rotary**, the value of **mom\_kin\_retract\_plane** will be transferred to **mom\_kin\_retract\_distance** and **PLANE** will be interpreted as **SURFACE** automatically.

## Pre V3.1 XZC Mill & Simple Mill-Turn Posts

Prior to version 3.1, C axis limits (Machine Tool -> Rotary Axis) of a XZC mill or simple mill-turn post were not observed.

Axis Limits (Deg)	
Minimum	0
Maximum	360

No checks were made against the values that were on the dialog for the rotary axis limits. Only the limits of the **fourth\_axis** Address -9999.9999 ~ 9999.9999 were used. Starting from version 3.1 the actual limits will be enforced.

Example of a post with rotary axis limits of 0 ~ 360:

### **V3.0.1**

G81 X1 Z1 R2.1 C180.0  
C260.0  
C340.0  
C420.0  
C500.0

### **V3.1**

G81 X1 Z1 R2.1 C180.0  
C260.0  
C340.0  
C60.0  
C140.0

When saving a legacy XZC or simple mill-turn post in Post Builder v3.1, if you need to preserve the effect of having the limits of the rotary axis being -9999.9999 ~ 9999.9999, you must change the values on the dialog accordingly.

## ***Pre V3.0 XZC Simple Mill-Turn Posts***

The information in the custom command **PB\_CMD\_init\_mill\_turn** and **PB\_CMD\_init\_mill\_xzc** will be converted to dialog items on the **Machine Tool** pages and removed. The custom commands **PB\_CMD\_start\_of\_mill**, **PB\_CMD\_end\_of\_mill**, **PB\_CMD\_start\_of\_turn** and **PB\_CMD\_end\_of\_turn** have been converted into event handlers. You can now edit the event handlers on the **Linked Posts** page.

## ***Pre V3.0 Linked Posts***

Custom command **PB\_CMD\_init\_multiple\_post** of all legacy posts has been converted to dialog items on the Linked Posts page and removed. The custom commands used to start and end each head have also been converted into event handlers. You can now edit these event handlers on the linked posts page.

## ***Pre V2.0.2 XZC Mill & Simple Mill-Turn Posts***

You will need to follow the following steps to rectify existing XZC mill posts created in Post Builder version 2.0:

1. Create a new XZC mill post using the existing post as the “controller” post. Make sure the new post has the same output units as the existing ones.

2. Transfer your changes of the variables, if any, from the custom commands `PB_CMD_kin_mill_turn_initialize` and `PB_CMD_kin_mill_xzc_init` to `PB_CMD_init_mill_turn` and `PB_CMD_init_mill_xzc` respectively.
3. Add a line of code containing the “return” (without quotes) statement into the beginning of both `PB_CMD_kin_mill_turn_initialize` and `PB_CMD_kin_mill_xzc_init` commands.
4. Examine and save the post.