

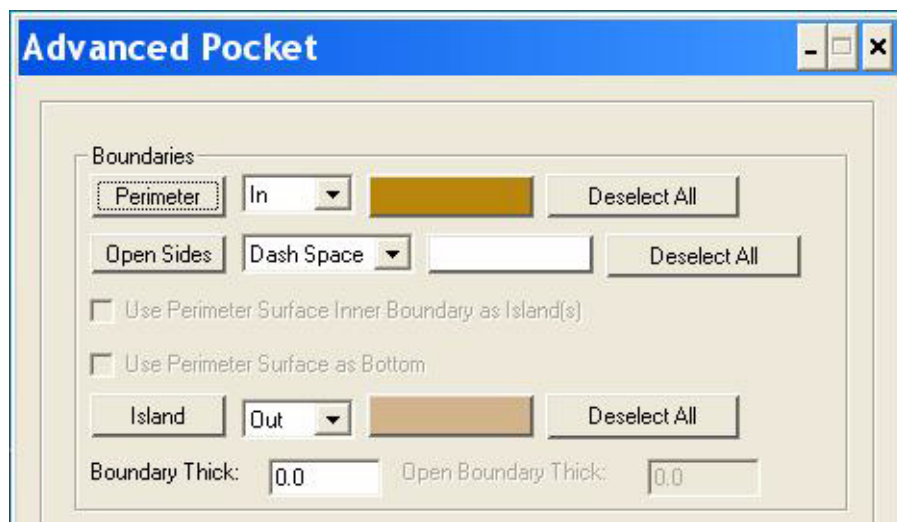
New in *NCL*

Motion enhancements

Advanced Pocket enhancements

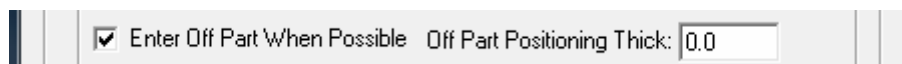
Added the option to specify open sided pocket boundary

- Open sides can be defined for pockets whose perimeter is defined using a composite curve or subscripted point array.
- There can be multiple open sides for a single perimeter entity.
- When defining multiple open sides in a single pocket perimeter the **Open Sides** button must be clicked for each open side being defined.
- Components of composite curves designated as open will be displayed in the user chosen color and line style.



Added the option to enter off the part through an open side.

- In addition an **OFFSET** distance an additional thick value can be specified to set the distance the cutter will be positioned from the boundary. The shortest entry through an open side will be used.



New Syntax:

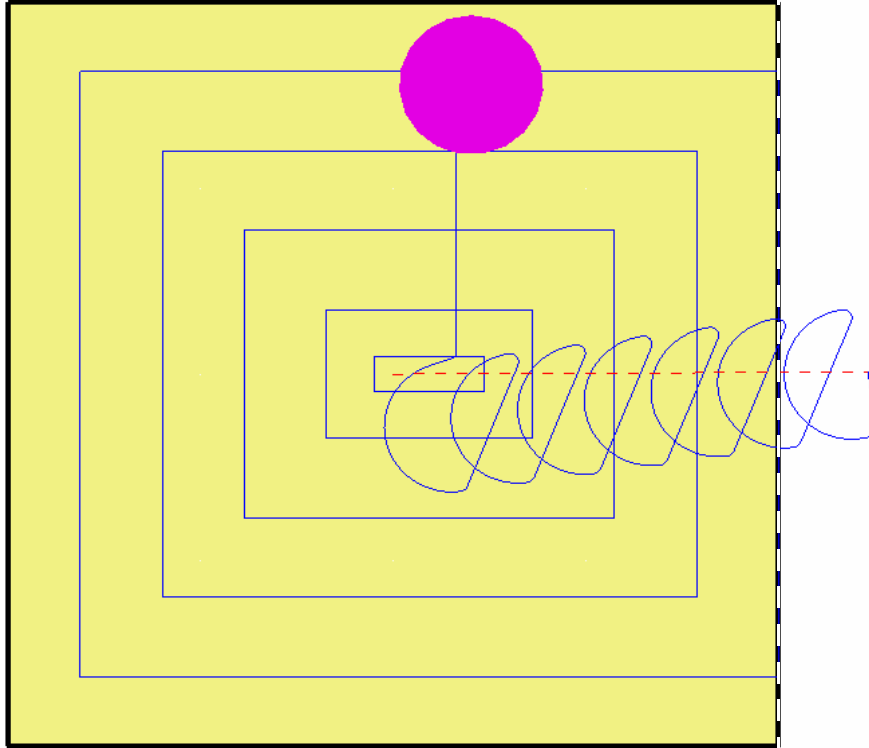
POCKET/sf1,[SAME],top-plane,OFFSET,sf,OFFSET,...,ofsthk

- Entering from off the part can revert back to the style used in previous versions of **NCL** using the version flag ***SET/VER,10.0** or earlier.




Added HELIX entry method for entry through an open pocket boundary

- The new method uses a flat helical motion to enter the part at the current pocket level.



In the above example the right side of the pocket boundary has been designated as open and the pocket is entered using flat helical motion (POKMOD Entry Method HELIX).

Click  **> Pocket > HELP** for more information on the new syntax

-- Please see the **NCL 10.1** reference manual for complete details.

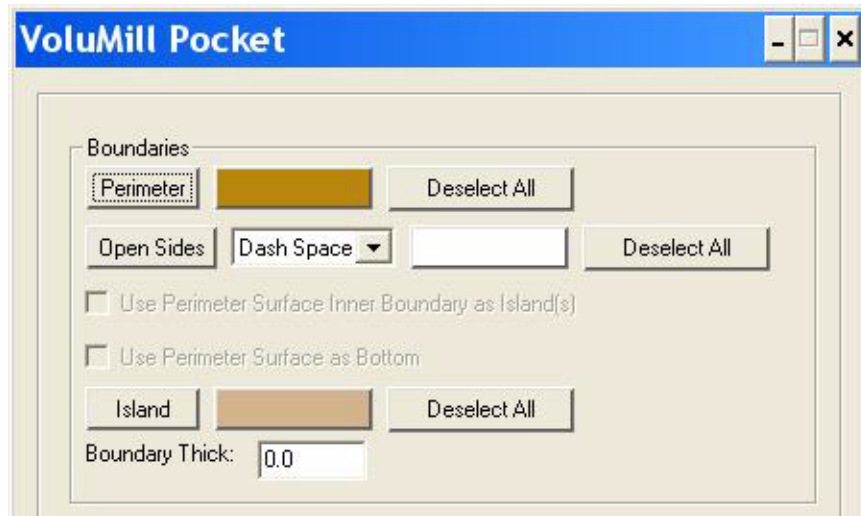
Enhanced selection of initial offset segment used for pocket motion

- The initial offset segment is used to determine which part of intersecting loops to keep and the starting point of the pocketing motion.
- Motion paths may be altered using the new method and therefore may not match older versions.


-- The ***SET/VER,10.0** command will cause pocketing to use the old logic.



Added option to specify open sided pocket boundary to VMPOCK



- Open sides can be defined for pockets whose perimeter is defined using a composite curve or subscripted point array.
- There can be multiple open sides for a single perimeter entity.
- When defining multiple open sides in a single pocket perimeter the **Open Sides** button must be clicked for each open side being defined.

Click  > **Pocket** > **HELP** for more information on this command.

-- Please see the **NCL 10.1** reference manual for complete details.

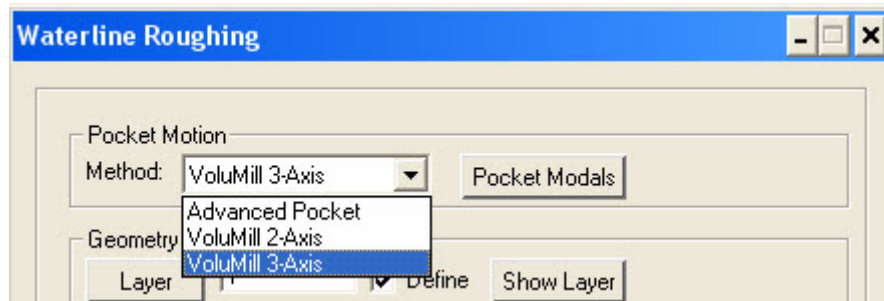
Waterline Roughing enhancements

Added the “Volumill 3-Axis” option for 3-Axis VoluMill™ style motion

- The Waterline routine can now be configured to generate 3-Axis VoluMill™ style motion.

New pocket command syntax parameter:

POCKET/LAYER=laynum [,VMP3AX],...

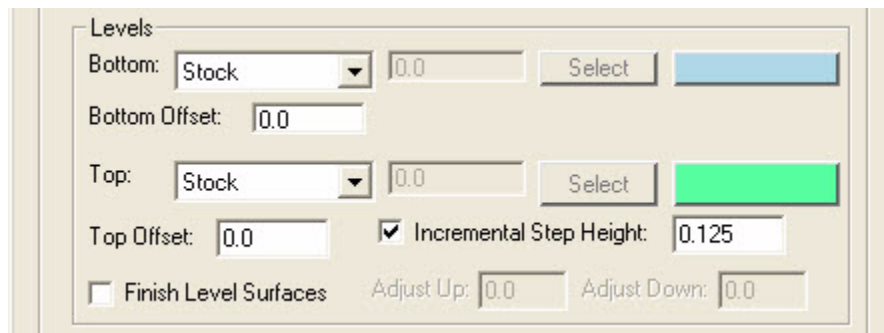


- The **VMP3AX** method differs from the existing **VoluMill 2-Axis** option in that **VMPOCK** (2-axis Volumill) slices the part surfaces at configurable Z-levels and uses the resulting contour(s) as the machining boundaries while the new **VMP3AX** uses boundaries that are created by VoluMill™

Added an Incremental Step Height feature

New pocket command syntax parameters:

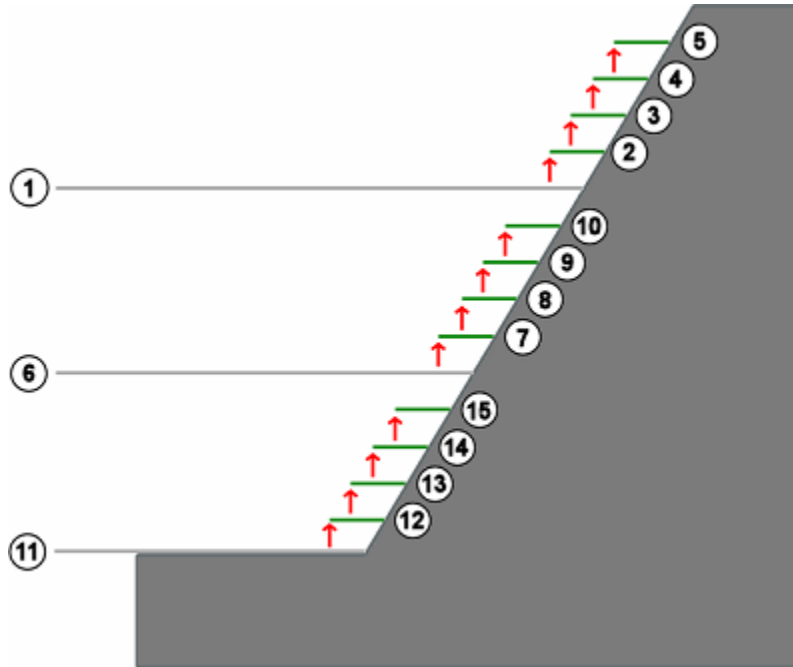
POCKET/LAYER=laynum,... [,STEP,UP,stp]



- Incremental Step Height defines the step up height for additional cutting levels made above the current cutting level after the current cutting level depth is finished.
- This feature allows for a finer step to be applied to open angle pocket walls without having to take smaller steps throughout the entire pocket.
- This parameter is only recognized when VMP3AX waterline roughing is used.



The following illustration is an example of the order of cutting levels where levels 1, 6 and 11 are the initial cutting levels and the remaining levels are the incremental step height cutting levels.



Additional cutting parameters supported by VoluMill™

New pocket modal syntax parameters:

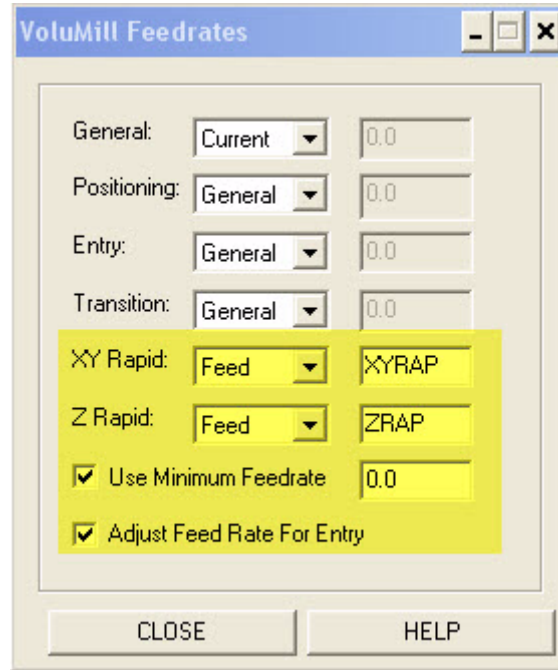
```
VMPMOD/...[ ,FLUTES,n,[flen[,tlen]]][ ,FEDRAT,f1,f2,f3,f4]
[ ,IN ][ ,SMALL,ON ][ ,MINFED,fd][ ,RAPID,xyrap[,zrap]]
OUT OFF OFF
```

New pocket modal form parameters:

- **Flute Length** **flen** defines the length of the cutting section of the cutter.
- **Tool Length** **tlen** defines the overall length of the cutter.



- **Contour Ramping Permitted** **SMALL,ON/OFF** specifies that when the **ON** parameter is given the configured entry style motion will be used to cut the regions that are too small to cut using standard motion. When the **OFF** parameter is given the regions that are too small to cut using standard motion will be omitted. **ON** is the default setting.



- **FEDRAT,...,IN/OUT** **Adjust Feed Rate For Entry** specifies whether or not to reduce the feed rate for entry moves. When **IN** is given the entry feed rate will be output as programmed. When **OUT** is given the entry feed rate will be reduced so the periphery of the tool moves at the entry feed rate instead of the center of the tool. **OUT** is the default condition.
- **MINFED,fd /OFF** **Use Minimum Feedrate** Defines the minimum allowed feed rate when **"MINFED,fd"** is given. When **OFF** is given the minimum feed rate will be ignored.
- **RAPID,xyrap[,zrap]** **xyrap** **XY Rapid** defines the rate the tool will feed for moves parallel to the XY-plane. **Z Rapid** **zrap** defines the rate the tool will feed for moves parallel to the Z-Axis. If **zrap** is not defined the value specified by **xyrap** will be used for moves parallel to the Z-Axis.

Tool axis mode enhancements

Enhanced the Tool Axis Perto Vector tool axis mode

Added an option to the **TLAXIS/...,PERPTO,vec** clause that allows the tool axis modifiers **FWD**, **RIGHT**, **GUIDE**, and **GOUCK** to be used while still maintaining perpendicularity to a supplied vector.

Below are the tool axis modes to which the **LAST** parameter has been applied:

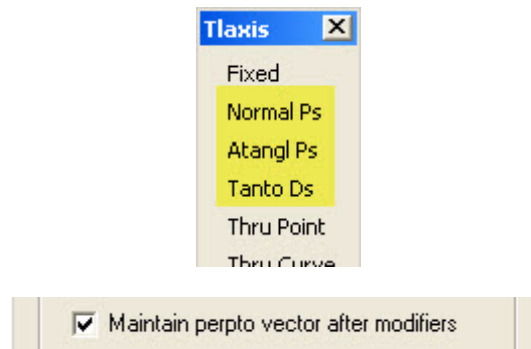
TLAXIS/NORMAL,PS,...,PERPTO[,LAST],vec

TLAXIS/ATANGL,PS,...,PERPTO[,LAST],vec

TLAXIS/TANTO,DS,...,PERPTO[,LAST],vec

- The “**LAST**” qualifier will enable this feature when specified with the **PERPTO** vector in the following tool axis commands.

A check box has been added to the following Tool Axis form menus to enable this feature:



Geometry Enhancements

Added ability to create composite curves from disconnected entities

A new command has been added that can create Composite Curves from entities that are not connected.

New command syntax parameters:

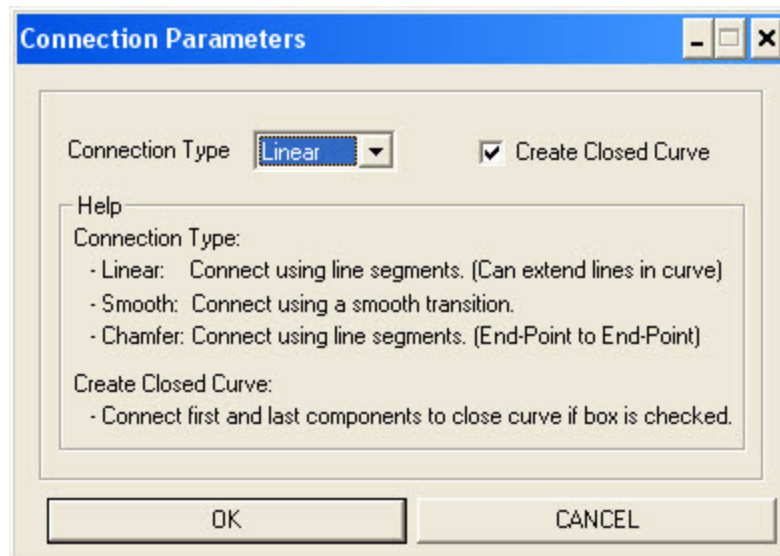
CV/COMPOS [,LINEAR [,CLOSE]], cv-list
SMOOTH OPEN
CHAMFR
OMIT

- When **LINEAR** is specified any required connections will be made by linearly extending the disjointed geometry so that they intersect. If the extensions do not

intersect then a line connecting the end points of the two entities is created (same as if **CHAMFR** is specified).

- When **SMOOTH** is specified all connections will be made using B-Splines to create a smooth transition. The B-Splines are defined using the tangent vectors at the ends of the entities.
- When **CHAMFR** is specified all connections will be made using a line between the end points of the two entities. Unlike **LINEAR** lines will not be extended.
- The default connection type is **OMIT** and no connections will be made. An error will be output if the geometry is not connected.
- **CLOSE** specifies that the beginning and end of the picked curve segments should also be joined so that the resulting curve is closed.
- **OPEN** is the default and specifies that no additional connection will be made to join the ends of the curve.

The following form has been added to interactively specify the command parameters:



Added the ability to offset components of a composite curve

The Composite Offset command has been added to define a new composite curve.

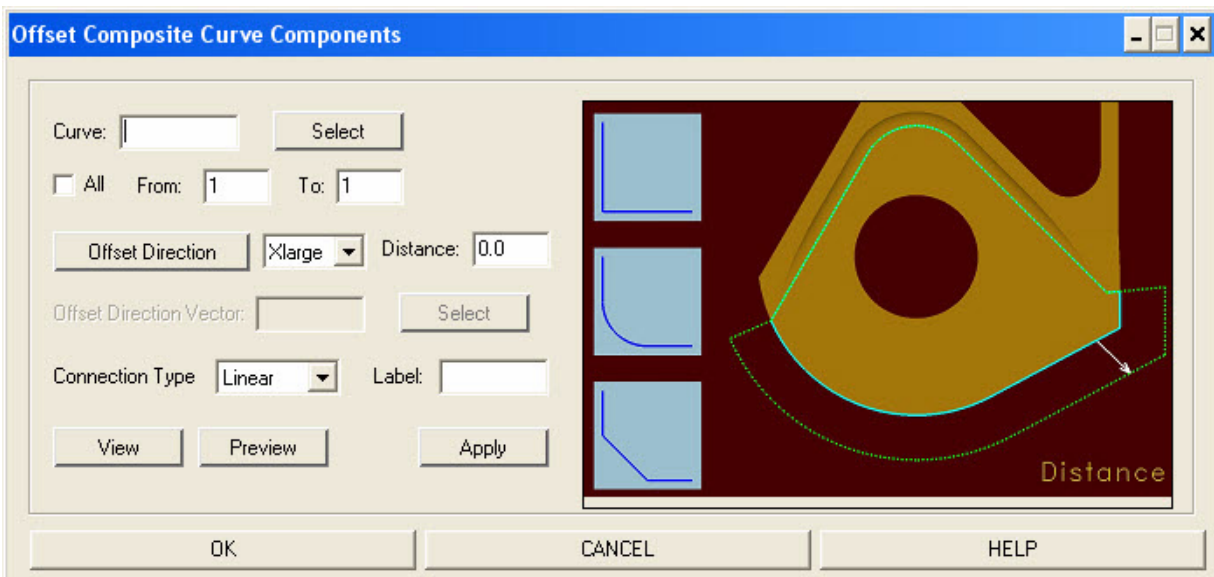


New command syntax parameters:

CV/OFFSET,curve, [comp1 [,comp2]], XLARGE,dist,LINEAR
ALL XSMALL SMOOTH
YLARGE CHAMFR
YSMALL
ZLARGE
ZSMALL
vc1

- **comp1** and **comp2** define the numeric index range for the components to be offset. For example: 4,6 specifies that only the fourth through sixth curve of the composite curve will be offset. If the parameter **ALL** is specified all components will be offset starting at the first component of the curve.
- The directional modifier (**XLARGE**, **YSMALL**, etc.) specifies the direction from the first component referenced to offset the curve. **vc1** specifies the label of a predefined vector or a nested vector to use for the offset direction. **dist** specifies the offset distance.
- **LINEAR**, **SMOOTH** and **CHAMFR** specify the connection type to be used if needed. The connection types are the same as defined above. **LINEAR** is the default connection type.
- When components of a composite curve are offset the component geometry of the original curve is maintained and new geometry is created and offset to define the new curve.

The following form has been added to the Curves menu to interactively specify the new commands' parameters:



Added the option to offset splines via a given vector.

CV/OFFSET, CV1, PV1, 0.25

- A vector or point vector can be given at the CMD line.

This command is source only.

-- Please see the **NCL 10.1** reference manual for complete details.

Create a Composite Solid from existing surfaces or solids

A new visual solid definition has been implemented that creates a composite solid from a list of existing surfaces and/or visual solids.

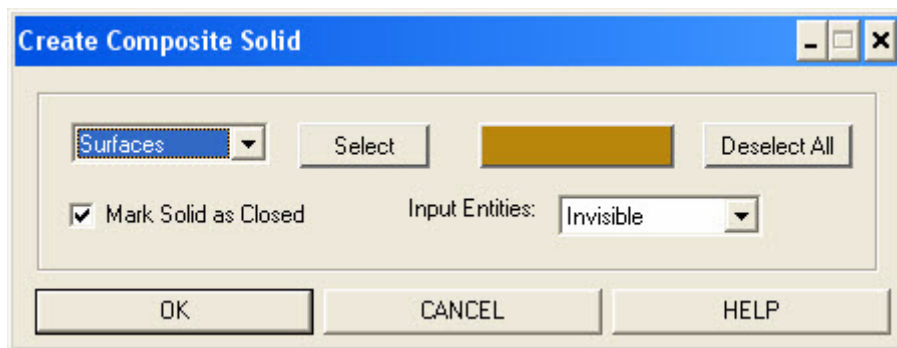
New command syntax parameters:

[name=]SOLID/COMPOS[,CLOSE][,INVIS],sf-list
OPEN RETAIN
REMOVE

- **COMPOS** is required for proper syntax. **CLOSE** will mark the solid as a closed solid. This setting currently has no effect on the solid, but will be used when exporting the Unibase using the **STEP** convertor. **OPEN** marks the solid as an open solid.
- **INVIS** will automatically invisible all of the surfaces/solids used to create the composite solid. This is the default setting. **RETAIN** will maintain the input surfaces/solids as they are and **REMOVE** will automatically remove the input surfaces/solids after the composite solid is created.
- **sf-list** is a list of surfaces/solids used to create the composite solid. It can be a list of geometry, a layer (**LAYER,n**), or the word **ALL**.

New Composite Solid form:

Click  >  >  to access the new form:



Create solids as external stock files

Added the **COMP** and **FCOMP** command parameters used in the external stock primitive file (.stk) creation to the **SOLID/LOAD,"file.stk"** command.

- Composite solids saved to an external stock primitive file will be stored as **COMP** and **FCOMP** commands. Include definition of COMP and FCOMP.....
- Only component solids will be stored in the stock file component surfaces contained in the composite solid will not be stored.

Utilize composite solids as IPV stocks and fixtures

A composite solid can be utilized as an **NCL/IPV** composite stock/fixture by using the **STOCK, FIXTUR/SOLID** command[s].

Extract surfaces/solids from NET surfaces and composite solids

Extract surf/solid command syntax parameters:

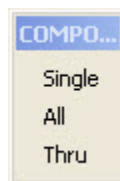
[name=]SOLID/OUT,geo,ALL **[,NUM,ncomp]**
SURF **id-list**

- It is recommended that you use the **SURF** definition when a *NET* surface is being referenced or the composite solid consists of a group of surfaces. The **SOLID** definition should be used for composite solids that contain a group of visual solids.
- **OUT** is required for proper syntax. *geo* can either be a *NET* surface or a composite solid.
- **ALL** will extract all surfaces/solids from the composite entity.
- **id-list** is a single value or range of values that specify which components will be extracted; the **THRU** clause is allowed.
- **NUM,ncomp** specifies a scalar that will receive the number of components extracted.

Extract surf/solid interface command:

Click  >  Box Pt-Pt >  Component to access:

- Clicking **Component** brings up the following menu:



- Choose **All**, **Single** or **Thru** to specify the entities that are to be extracted from the

solid.

- The interface behaves identically to the method used to extract components from composite curves; that is simply select on the component(s) of the composite geometry to extract then press the middle mouse button.

Enhanced the list of entities that can be saved to an external STL file

- The **SAVE/STL** command will now store a standard surface/solid list i.e. (a list of geometry, a layer, or the word ALL).
- A composite solid can now be specified and all components of the solid will be stored in the *STL* file.
-- Please see the **NCL 10.1** reference manual for complete details.

Selected surfaces multiple times when chain selecting

- Surfaces can now be selected multiple times when using the chain select feature.
- This feature is beneficial when chain selecting surfaces that contain a U-shaped surface with other surfaces connecting the uprights of the U.

The following image contains a sample of this type of geometry with the U-shaped surface displayed in Cyan and the in-between surfaces in magenta



- The user has the option of enabling this feature or disabling via the new check box on the **Chain Select Modals** form.

- As with all forms checking this box then hitting **OK** will write the new setting to the ncl_chain.mod file.

Added the ability to trim and extended composite curves

- The **REDEF** command can now be used with Composite Curves.
- Composite curve are now a valid entity pick when using the **Trim-Extend**, **Mid**

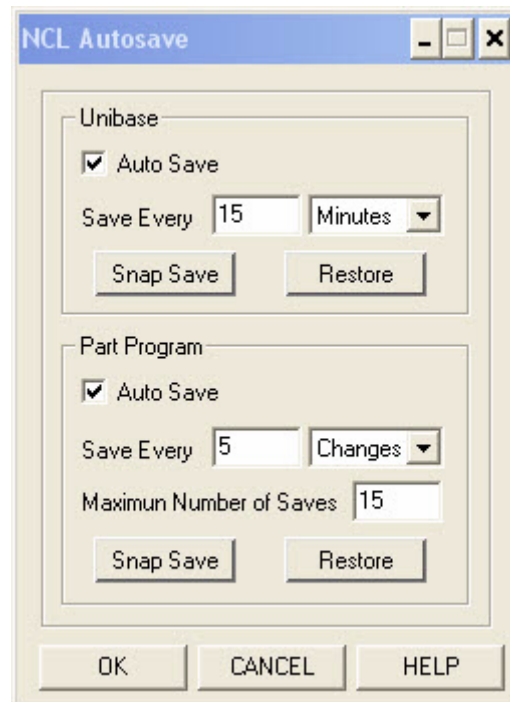
Trim and **Untrim** commands from the **NCL Edit** menu.



Interface Enhancements

Added an option to automatically save the part program

The form below has been added to the **NCL** "File" menu:



- Files can be saved to an external file based on either an elapsed time (Minutes) or number of changes (Changes) made to the file.
- The Unibase automatic save feature only generates one Unibase and overwrites it with each save during an **NCL** session.
- The Part Program automatic save feature can/will save multiple versions of the Part Program during an **NCL** session. The AutoSave files will be named as indicated below:
%ind_filename.ext
- Where: **ind** refers to the chronological number of the saved file and will start at the lowest number available (0 if no previous AutoSave file exist).
- filename** refers to the name of the part program currently loaded in **NCL**.

- **ext** refers to the file extension of the part program currently loaded in **NCL**.
- The **Maximum Number of Saves** field defines the maximum number of AutoSave files to maintain during the active session. Once the **Maximum Number of Saves** limit is reached AutoSave files will begin to be overwritten (recycled) starting with the oldest saved file.
- The **Snap Save** button will cause **NCL** to immediately generate a Unibase or Part Program AutoSave file. The **Snap Save** button can be used to save a Unibase or Part Program when the **Auto Save** checkbox is not checked.
- The Part Program **Restore** button will bring up a file browser so that the user can select a previously saved file to load into the current **NCL** session. Pressing the **Restore** button from the Unibase section behaves the same as loading a unibase (currently geo. will be overwritten).
- All automatic save files from the current session will be deleted at the end of the current session or when a new Unibase or Part Program is loaded into the current session.

Added a Statistics record to the **NCL** Unibase file

Click **Defaults/Unibase Statistics** menu to access the form:

Unibase Statistics

Originating System	
System: NCL V10.056	Organization: NCCS
System 2: *WINDOWS*NT*	Author: J. Sample
File: G:\test.u	Date: 04-FEB-2014 16:20:31
Translator:	
Updated	
System: NCL V10.056	Organization: NCCS
Author: J. Sample	Date: 04-FEB-2014 16:25:19
Notes	
This is a test unibase that was created with the following variables set in user.init	
OK CANCEL	

- The Statistics record stores the following information about the creation and modification history of the Unibase.



- Originating CAD/CAM system.
- Originating Company name.
- Originating Author.
- Original name of file.
- Date originally created.
- Translator used to import model.
-
- **NCL** version of last modification.
- Company name of last modification.
- Author of last modification.
- Date of last modification.
-
- User defined notes detailing the model and modifications.

This feature has been added to the *NCL/IGES* and *NCL/STEP* convertors so that the Unibase Statistics record is stored with the information provided in the header of the input *IGES* and *STEP* files.

Added an optional bottom Bar to the *NCL* window frame

The custom Bottom Status Bar is defined in the **NCL** layout file under the #STATUSBAR# section.

- Drag and drop any menu or individual menu button on to the new Bottom Status Bar.
- Up to 4 menus or status areas can be defined in the **Status Bar**.
- Each menu or status area will be the same size in the **Status Bar** and their size cannot be changed.
- The buttons defined in a status file (file.stat) can be assigned optional “**Tool Tip**” text that will override the default tool.

Example:

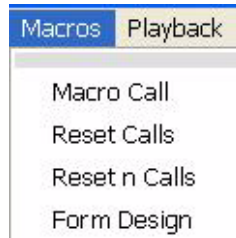
/Mode = / PROCESS_MODE, <Tool Tip>, CAM_COMMAND

- Status menus can be added to the **Bottom Status Bar** by manually adding the desired “.stat” menu to the *ncl.lay* file.
- **NCL 10.1** ships with a default Bottom Status Bar that contains the following items positioned from left to right:

Tool Position Menu; Tool Axis menu; CMD mode status buttons; as well as Analyze Geo., Invisible Geo., Geometry Status; and Motion Status, icons

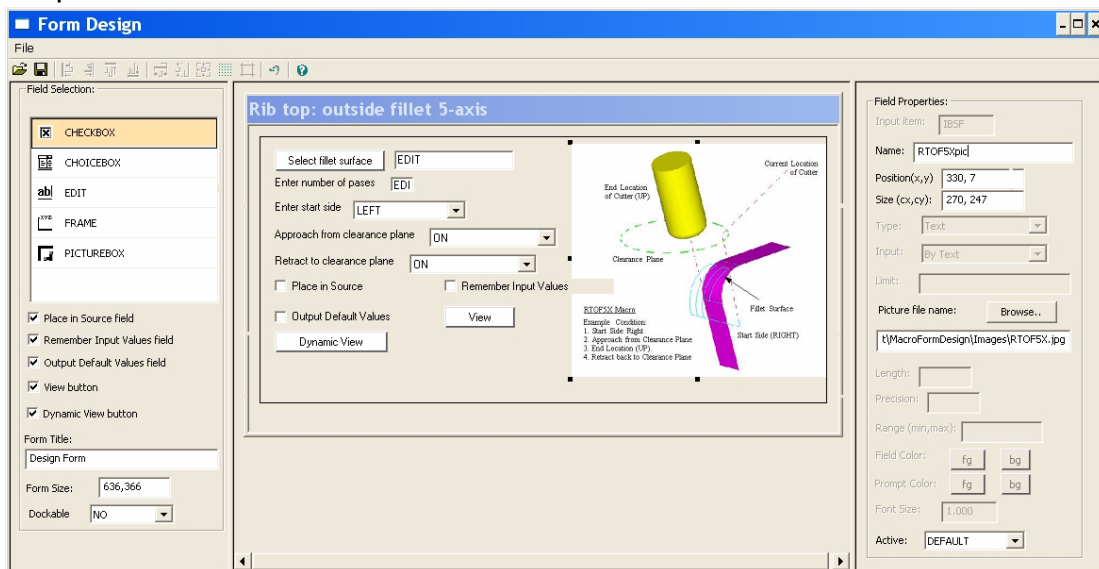
Added a tool to aide in the customization of macros forms

A new pick has been added to the **NCL** Macros menu



Choosing **Form Design** will bring up the **Select Macro** form that displays a list of available macros. If no macros are available to the current **NCL** session, (none appear on the form), the “**Load**” button can be used to locate and load an **NCL** macro. Once the macro is chosen and the “**OK**” button is clicked the new **Form Design** form will appear with the selected macro loaded.

Below is a screen shot of the new Macro **Form Design** tool being used to add a picture to an **NCL** macro:



- In the picture above a “**PICTUREBOX**” field type has been dragged from the “**Field Selection:**” column and dropped on to the RibTop macro form. The macro form and Picture Box were then stretched to accommodate a larger image then an image was loaded into the Picture Box by pressing the “**Browse**” button before finally stretching the Picture Box and form to the final desired size.
- Clicking the question mark at the top of the **Form Design** form will bring up a window into which “**Help Text**” can be entered. Clicking “**Accept**” then saving the form will add a button titled “**HELP**” to the bottom of the macros’ form.

Please note that the **HELP** button will not appear on the form when it is loaded in the Form Design tool. The **HELP** button along with the **OK** and **CANCEL** are only

visible when a macro is being used interactively.

- Clicking the Align (Left or Right) Icon will align the second field selected with the first that is selected.

Added ability to activate form fields by clicking on a forms pictures

- An area within a **PICTUREBOX** can be made interactive once a **PICTUREBOX** is added to a macro form by editing the customized macro form (.frm file) and adding the code below:

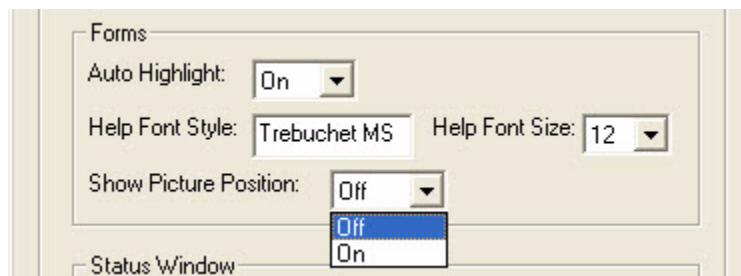
/PICTURE/ label [,"tooltip"], xmin,ymin,xmax,ymax [,param]

Where the parameters are defined as follows:

- **label** is the name of the PICTUREBOX this field refers to.
- **"tooltip"** is an optional quoted text string that contains the text that will be displayed when the cursor is moved over the defined (xmin,ymin,xmax,ymax) area in the picture.
- **xmin,ymin,xmax,ymax** defines the area of the picture that is used to activate this field.
- **param** is an optional parameter such as a label or text string that would serve as the default for the form field to which the picture is attached.
 - Please see the **NCL 10.1** reference manual for complete details.

ALERT: Please be aware that the (macro) Form Design tool does **NOT** write the **/PICTURE/** parameter to the .frm file when the macro form is saved therefore any **/PICTURE/** parameter that is manually added to a macros form file (*macroname.frm*) will be overwritten any time the Form Design tool is used to save a macros form.

Added a Show Picture Position field to the Interface Modals form



- Selecting **OFF** allows tooltip text to display when the cursor is within a defined region of the picture.
- **ON** will display the X,Y location of the cursor within the picture so that the user can define areas of the picture that will activate the form field to which the picture has been associated.

Miscellaneous *NCL* enhancements

Added support for temporary variables in loops

- Temporary variables will be defined in the same fashion as with Macros using the #label syntax.

Example:

```
-- loopst
-- #lab1=1
-- DO/mkpts,i=1,10
-- ppt(#lab1)=pt/i,0,0
-- #lab1=#lab1+1
-- mkpts:contin
-- loopnd
```

Enhance the list of valid Macro parameters

- **DATA** statements and **quoted text strings** can now be used as Macro parameters.

Modified macro parameter behavior when Modal Call is enabled

- When a macro form is called with **Modal Call** enabled the parameters will now be remembered only if the **Remember Input Values** check box is checked.

Added new variable to specify the name of the temporary clfile

In previous versions the temporary cl file generated during a verification session was always 'ncltemp.cl'. This fact limited the usage and abilities that could be added to custom post macros for specialized machine tools.

- The new environmental variable specifies the name of the temporary clfile used during motion playback and **NCL/IPV** simulation when a Simulated, Posted, or foreign clfile type is selected as the Source file in the Playback File form.
- Using the string %s the name of the active part program file is inserted into the temporary clfile name. Therefore the default setting is as follows:

```
-- NCL_TEMP_CLFILE = temp_%s
--
-- Part Program file = 1h760_7px.pp
-- Temporary clfile = temp_1h760_7px.cl
```

The **THRU** clause now skips missing entities

- Entities that are missing in a given **THRU** clause now skipped and the remaining entities processed instead of ending the list at the missing entity as in previous versions.

Fixed parsing of labels that begin with an "_" underscore

- This fixes various issues that could arise when running commands on imported symbols which are commonly labeled as "_longsymbolname" i.e. issues when attempting to group or decompose them.

Utilities (*RUN, *SHOW, etc.) now processed under REMARK/ON

- This addition addresses an issue where **NCL** would not respond to ***EDT** or ***RUN** commands after a **REMARK/ON** was encountered thereby forcing the user to exit **NCL** and manually remove the **REMARK/ON** statement in order to proceed with processing and or editing.

Please be aware that even though a **REMARK/ON** is issued **NCL** will now **STOP** processing at all ***STOP** and ***PAUSE** commands unless the ***RESET/STOP** and ***RESET/PAUSE** commands are issued before the **REMARK/ON**.

Fixed source formatting of commands that accept text strings

- Commands that can accept text strings as input (e.g. **UBFN**, **LOADU**, **SAVEU**,...) will no longer be formatted before being written to the part program. This fixes a problem where, for instance, a **UBFN** command that contains a long path name would be errantly broken into a continuation line if it were longer than the configured **NCL** line length.

Increased ISN numbers that can be stored in an external clfile

- Increased the maximum ISN number that can be stored in an external clfile from 32,767 (65,356 in some programs) to 2,147,483,467.
- A new Type 1001 record has been added to the external clfile that contains the ISN value when this value is over 32,767.
- Part program files that are less than or equal to 32,767 lines will output the clfile in the same format as previous versions of **NCL**.
- The addition of the Type 1001 CL record for ISNs over 32,767 should not adversely affect existing post-processors.

New in *NCL/IPV*

Added ability to display the MCD data lines during IPV simulation

- MCD data lines can now be displayed during *NCL/IPV* simulation when playing back an MCD or Posted file by selecting **Yes** from the **Show Source** menu on the **NCL/IPV Playback Control** form.

In order to play back an **MCD** or **Posted** file do the following: (Basic Steps)

- Set the **Source** field on the **Playback File** form to **MCD** or **Posted**.
- Click the **Options** button to reveal the **MDC Conversion Options** form then hit the **Scan** button to load the programs Post into the **MDF File:** field.
- Click **CLOSE** on the **MCD Conversion Options** form then click **OK** on the **Playback File** form.
- Click **Simulate** from the *NCL/IPV* menu and set the **Show Source** field to **Yes**
- Click **OK** on the **NCL/IPV Playback Control** form to begin simulation.

Enhanced PPRINT IPV error messages when generated in *NCL/IPV*

- Any error messages that are generated when processing **PPRINT IPV** commands during *NCL/IPV* simulation will now contain a more descriptive explanation of the error encountered.

Added support for composite stocks/fixtures

- Added support for stocks and fixtures defined as a composite of individual stocks/fixtures.

PPRINT IPV FIXTUR COMPOS id id-list
STOCK

- COMPOS** states that this definition is a composition of previously defined stocks/fixtures.
- id** is the numeric identifier for the composite definition. **id-list** is a list of predefined stocks/fixtures that will be combined into a single stock/fixture definition.

Once a stock/fixture is added to a composite definition it will be removed from the list of active stocks and can only be referenced as part of the composite stock/fixture.



Added support for decomposing composite stocks/fixtures

- Added support for the decomposition of composite stocks back into their individually referenced stocks by using the following command.

```
PPRINT IPV FIXTUR DECOMP id id-list  
STOCK
```

- id** is the starting ID number of the extracted stocks/fixtures. If **id** is set to 0 the ID numbers that the extracted stocks/fixtures had before they were stored in a composite solid will be used. **id-list** is a list of composite solids to decompose.

Added ability to move a stock/fixture by an incremental amount

Previously a stock/fixture could only be moved through a matrix from its' original position. In **10.1** stocks and fixtures can be moved through a matrix by an incremental amount from their current position.

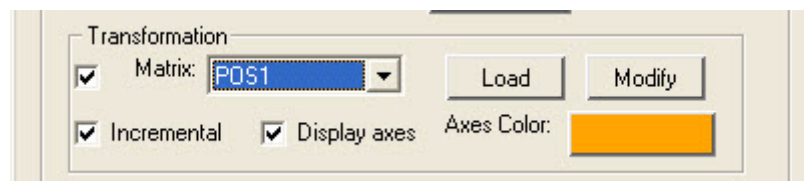
The following commands have been enhanced in support of this feature.

```
PPRINT IPV FIXTUR MOVE 0 ,mx-parms,id-list  
STOCK 1  
INCR
```

```
FIXTUR/MOVE,id-list,AT ,mx  
STOCK INCR
```

To move a stock/fixture incrementally from its current position a value of 1, (or the word INCR), must be specified on the **PPRINT IPV STOCK/FIXTURE MOVE** or the **NCL STOCK/FIXTURE MOVE** command. The standard value of 0 will position the stock/fixture using the specified matrix in relation to the original location of the stock/fixture.

Updated the Modify Stock/Fixture Attributes form for INCR moves



Added ability to translate a stock/fixture by specifying XYZ values

- Stocks and fixtures can now be moved by specifying XYZ translations as well as via the traditional 12-parameter matrix.

```
PPRINT IPV FIXTUR TRANSL 0 ,x,y,z,id-list
      STOCK      1
      INCR
```

- The new **INCR** option also applies to this new command form.
- The standard value of 0 will position the stock/fixture using the specified matrix in relation to the original location of the stock/fixture.

Added ability to rotate a stock/fixture about a major axis

```
PPRINT IPV FIXTUR XYROT 0 ,angle,id-list
      STOCK YZROT 1
      ZXROT INCR
```

- The absolute/incremental parameters are the same as they are in the **PPRINT FIXTUR TRANSL** command.
- As in the **NCL** matrix commands **XYROT** rotates the solid about the Z-axis, **YZROT** about the X-axis, **ZXROT** about the Y-axis and **angle** specifies the angle to rotate the solid.

Optimized the cloning of stocks and fixtures

The internal **NCL/IPV** primitive is now copied instead of creating a new stock definition for the cloned solid. This new method dramatically improves the speed of cloning solids especially when the cloned solid was imported from an external STL model.

Added ability to move a stocks/fixtures during the cloning process

PPRINT IPV CLONE has been expanded to support this feature.

```
PPRINT IPV FIXTUR CLONE id [AT, x1,y1,z1,d1, x2,y2,z2,d2, x3,y3,z3,d3]
      TRANSL,x,y,z
      XYROT,ang
      YZROT
      ZXROT
```

- A full matrix (**AT**), translation values (**TRANSL**), or a rotation plane, (**XYROT**, **YZROT**, **ZXROT**), with an angle are now valid command parameters.

- Stocks/fixtures moved using any form of the **CLONE** command will always be moved incrementally relative to their current position.

Added options to redefine/reset a stock or fixture's attributes

Added check boxes to the **NCL/IPV Geometry Modals options** form to reset all stock and fixture attributes when they are redefined and one to delete all defined stocks/fixtures when the **NCL/IPV** session is reset.

- Checking the **Reset on Redefinition** box causes all stocks and fixtures to take on the default attributes as defined in this form whenever they are redefined, (either interactively or when restarting the simulation).
- When this box is **not** checked all stocks and fixtures will maintain their current color, translucency, visibility, active state, tolerance and edge display when they are redefined.
- Checking the **Delete on Session Reset** box will cause all defined solids to be deleted when the **NCL/IPV** session is reset. This ensures that the session will start fresh with no defined solids.

If any stocks or fixtures are created interactively this setting should **NOT** be used. This is because interactively created solids will **NOT** be automatically re-created when the simulation is re-run. **NCL/IPV** will keep all solids defined interactively and those that are loaded via a Simulation file if this box is **NOT** checked. (Stocks and fixtures that are contained in a .sim or .cl file will be re-read and thus duplicated when the motion is simulated after the reset if this box is **NOT** checked.)

Position individual axes without having to program a motion block

Syntax:

PPRINT IPV POSITN label pos [label2 pos2 [...] labeln posn]

This command will position machine axes at the programmed positions.

- **label** is the label of the axis to position as defined in the **NCL/IPV** machine model file and **pos** is the new position for the chosen axis.
- Multiple axes can be specified in a single command and all of the *specified* axes will move at the same time.

Example:

PPRINT IPV POSITN X 10.0 A 90.0 C 0.

Control axes that do not have a corresponding linear or rotary axis

- These axes will be displayed and can be manipulated interactively in the **Axes Position** and **Axis Offsets** forms.

The standard machine positioning and offsets forms will display a maximum of 10 axes total. The forms displayed when using a custom Stringer machine will display a maximum of 3 user defined axes in addition to the 10 standard machine axes.

- They can only be positioned from the simulation file by using the **PPRINT IPV POSITN** command.

Added support for Work Coordinate offset Systems (WCS)

A (WCS) system is activated by the use of the **CUTCOM/ADJUST** and **TOOLNO/ADJUST** commands.

- The following command syntaxes are supported by **NCL/IPV**.

CUTCOM / ADJUST,ON
TOOLNO **OFF**
 n [,PLUS]
 MINUS

- **ADJUST** is required for proper syntax.
- **ON** will enable (WCS) offsets for Register 0.
- **OFF** disables (WCS) offsets of the types (**CUTCOM,TOOLNO**).
- **n** specifies the register to enable offsets for.



- **PLUS** enables offsets in the positive direction (default) and **MINUS** enables offsets in the negative direction.

The **CUTCOM** and **TOOLNO** (WCS) offsets are stored in separate registers and can be active at the same time.

External Workplane Coordinate System definition file

In order for the (WCS) offsets to have any effect on the machine axes positions an *external text file* must be created that contains (1) the type of offset register, (2) the register number and (3) the axis offsets for that (WCS) register.

The (Offset register File) “.ofs” must have the following format:

```
FIXTURE reg label1 ofs1 [label2 ofs2 ... labeln ofsn]  
TOOL
```

The “.ofs” file can contain multiple lines in the format above that define the axes offset values for the both the (WCS) registers.

- **FIXTURE** defines a **CUTCOM** style offset register and **TOOL** defines a **TOOLNO** style offset register.
- **reg** defines the register value that the offsets are being defined for.

Example: If an H15 code enables WCS offsets enter 15 for reg

- **label** specifies the label of the axis which will be offset and **ofs#** is the offset value to apply to this axis when the **reg** (WCS) offsets have been enabled. (If an “.ofs” file exists without the **CUTCOM/ADJUST** or **TOOLNO/ADJUST** commands being used the “.ofs” file will be ignored).
- **label** must match the label of one of the axes in the machines “machine definition file”.
- Multiple axes can be specified for each (WCS) register.
- An “.ofs” file must be loaded to enable (WCS) offsets

The following command is used to load the “.ofs” (WCS) offset file:

```
PPRINT IPV OFFSET "file.ofs"
```

Directly define axis offset values without an external “.ofs” file

Syntax:

```
PPRINT IPV OFFSET label1 ofs1 [label2 ofs2 ... labeln ofsn]
```

- The offset values and types are treated exactly the same as if they came from the WCS offset file.

All PPRINT IPV now processed before the first motion on reset

NCL/IPV will automatically process all **PPRINT IPV** commands up to the first motion when starting and resetting a session.

- This is so that any stocks defined at the beginning of the program are displayed and in their initial orientation before the start of any motion.

Previously stock and fixtures would not be located to their initial orientation until the cutter began to move thus, sometimes, causing the user to miss the initial entry cut.

PPRINT IPV Tool and Modals cmds. processed on Tool-list scan

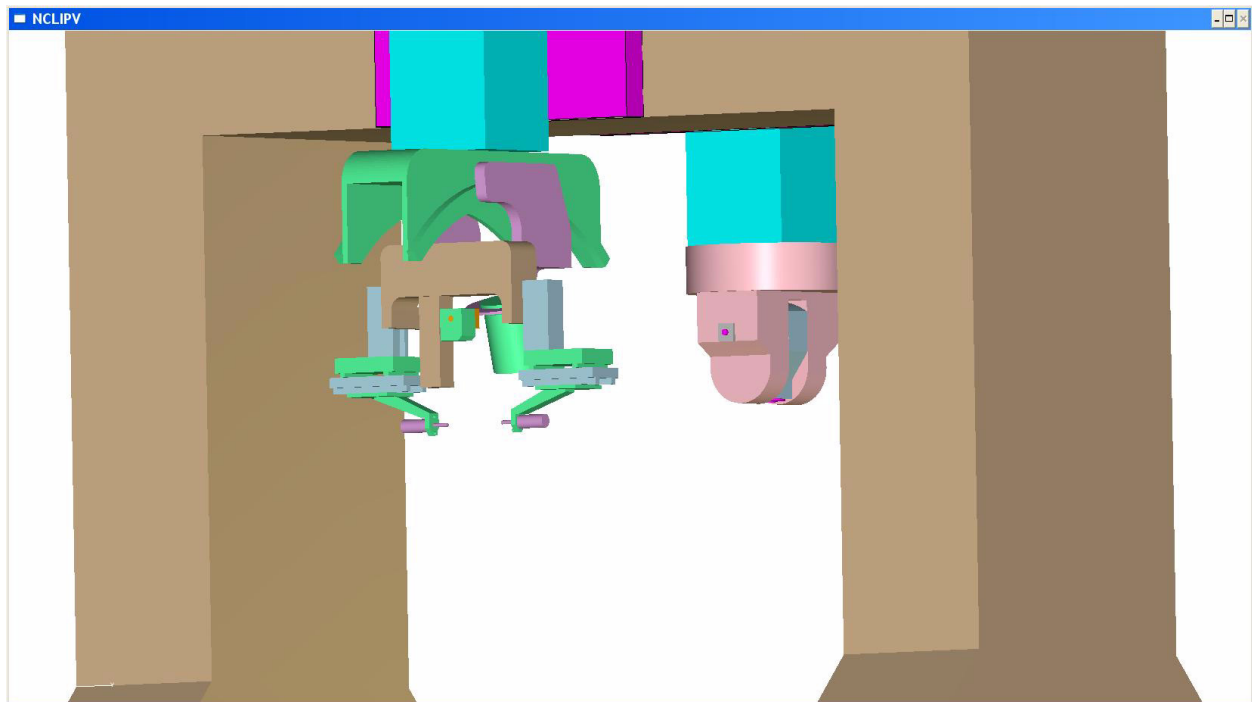
The **PPRINT IPV MODALS TOOL** and **PPRINT IPV TOOL** commands are now processed when scanning (loading) the simulation file. This will ensure that the default settings set using **PPRINT IPV MODALS** commands are used when processing cutters, especially, when an “.ofs” file has been defined.

New in *NCL/IPV* Machine Simulation

Added support for custom multi-axis/multi-spindle mach. simulation

The TORRESDRILL (Stringer Drilling) machine

Support has been added for the TORRESDRILL (Stringer Drilling) machine used in the drilling of structural components. This machine has separate milling and drilling heads where the drilling head has two sets of linear axes along with two spindles and a separate clip holder axis (Q).



The machine above is an example of a multi-axis/multi-spindle machine that can be simulated using *NCL/IPV Machine Simulation*.

Please contact **NCCS** if your company is interested in a custom multi-spindle/axes machine.

- A custom **Stringer Axes** positioning form has been created for this machine to accommodate the extra axes.

Enhanced the *NCL/IPV* Monitor Options form

- Extra entries have been added to the *NCL/IPV Monitor Options* form to accommodate the axes.

Updated PPRINT IPV commands to control additional axes

- The **NCL/IPV** post-processor command that defines the machine type has been updated to support this machine as follows:

PPRINT IPV MODALS MACHINE STRINGER.

- Added support for multiple active spindles within machine simulation.

PPRINT IPV SPINDLE m [,n,o,...]

- Added support for multiple active spindles to the **SPINDLE LOAD** command.

- The **LOAD** command determines the spindle(s) to load the next cutter definition into. By default the cutter will be loaded into the currently active spindles. Using the **LOAD** command you are able to load different cutters in different spindles without changing the active spindles.

PPRINT IPV SPINDLE LOAD m [,n,o,...]

Tooling Pin form now has a label field rather than name field

- Multiple Tooling Pins can be configured and each can be attached to a different machine axis independently of one another.

The screenshot shows the 'Tooling Pin' dialog box. It has a title bar with the text 'Tooling Pin' and standard window controls (minimize, maximize, close). The dialog is divided into two main sections: 'Machine' and 'Part'.
In the 'Machine' section, there is a 'Label' dropdown menu with 'BasePin' selected. Below it is a list of axes: PIN1, PIN2, PIN3, PIN4, and PIN5. To the right of this list are input fields for each axis, with '0.000in' visible for PIN1.
In the 'Part' section, there are three input fields: 'Origin' with the value '<0.000in,0.000in,0.000in>', 'Axis' with the value '<0.000,0.000,1.000>', and 'Flat' with the value '<1.000,0.000,0.000>'.
At the bottom of the dialog, there are four buttons: 'Place Stock', 'Reset Stock', 'CLOSE', and 'HELP'.

- The **PPRINT IPV TOOLPN** command has also been updated so that individual stocks and fixtures can be assigned to a specific tooling pin on the machine.

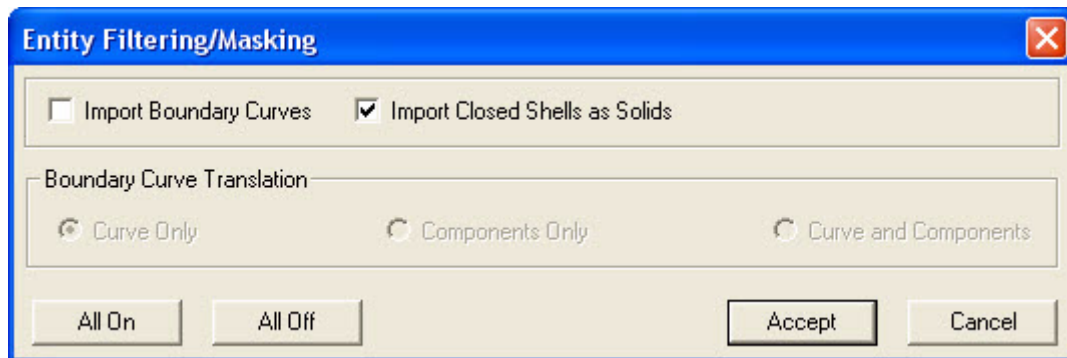
PPRINT IPV TOOLPN [label] [x,y,z,i,j,k,u,v,w]

New in *NCL/STEP* converter

Create composite solids from Closed Shells

Added a feature to the *NCL/STEP* convertor that will create composite solids from Closed Shells.

- This feature is enabled by checking the **Import Closed Shells as Solids** box in the **Entity Filtering/Masking** form.



- This feature allows the user to manipulate closed solids imported from a **STEP** file as a single entity once loaded in *NCL*.
- A **SOLIDS** fields has been added to the **Geometry Naming Modals** form to define the default labeling method for solids.



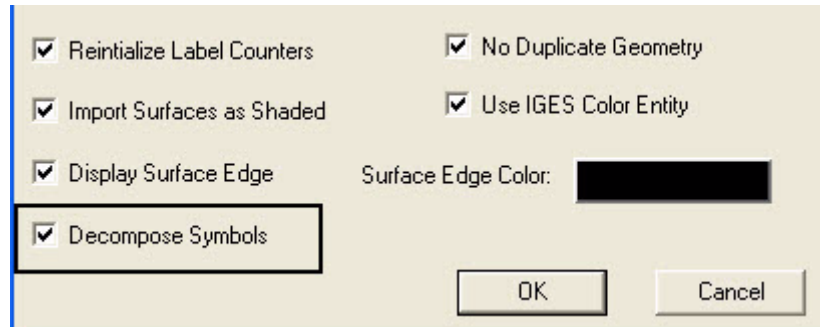
- Using the new “**Import Closed Shells as Solids**” will result in a significantly smaller unibase when importing a complex STEP file.

New In IGES

Added the option to decompose symbols

Added the option to decompose symbols during the translation of an **IGES** file.

- To enable the decomposition of symbols check the **Decompose Symbols** box on the **IGES IN OPTIONS** form.

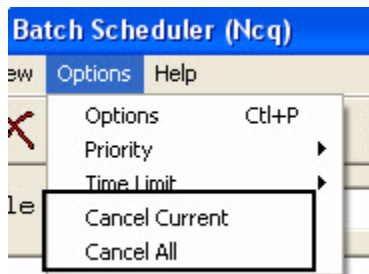



- To configure the default condition to decompose symbols set the /DECOMPOSE/ parameter in the “[ncliges.mod](#)” modal file to *YES

New in *NCQ*

Added two new options to the *NCL* Batch Scheduler (Ncq) form

Added two new options to the **NCQ Batch Scheduler** form on the **Options** menu that will allow the user to cancel the current file being processed.



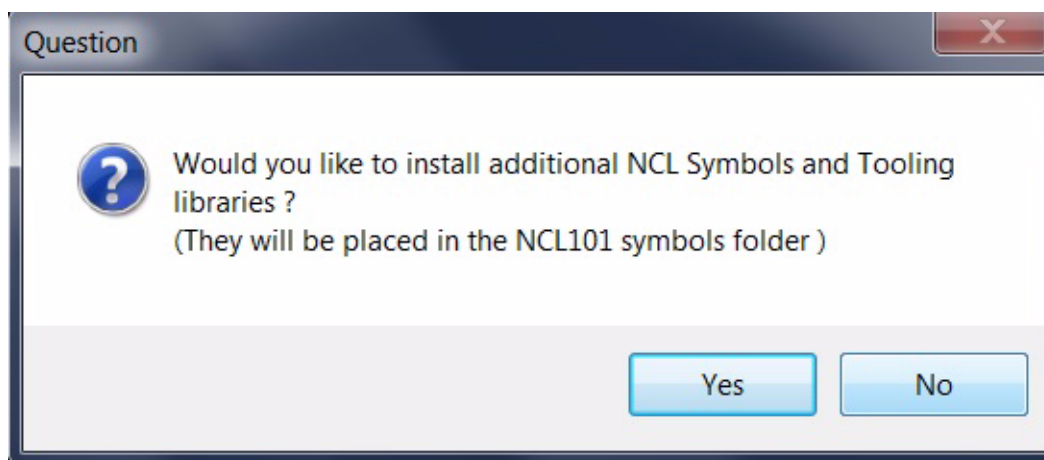
- The menu choice **Cancel Current** will cause **NCQ** to cancel the current file and move to the next file in the queue.
- The menu choice **Cancel All** will cause **NCQ** to cancel the current file and pause the batch processing. Batch processing can be resumed at the next file in the queue by pressing the  button.

New in **NCL/TOOLIB**

Tooling libraries now included on the **NCL 10.1** install media

Three distinct tooling libraries that include tools, fixtures and clamps have been included as an option on the **NCL 10.1** install.

- The option to install the tooling libraries will appear at the end of the **NCL 10.1** install.



- Please contact **NCCS** for guidance on how to unpack and utilize the tooling libraries.

Download the VoluMill™ Advisor for practical Materials/Speeds/Feeds

VoluMill™ offers a handy calculator that contains typical as well as aggressive speed and feed suggestions that take into consideration the type of material being used as well as the strength with which the material is fastened to the machine.

To download the Celerative Milling advisor click on the following link, fill out the customer information and click: Submit:

<http://www.volumill.com/content/download-millingadvisor-desktop-version>

Once downloaded a button to activate the Celerative Technologies Inc. MillingAdvisor Desktop Version can be added to your **NCL** motion menu by following the steps below:

1. Copy the file “motion.menu” that is currently in the “NCCS\NCL101\doc” folder to your **NCL 10.1** user area. The **NCL 10.1** user area is located at the following path “C:\Users\YOURNAME\NCL101\menu” folder. (Where the text “YOURNAME” is your log in name.)
2. Run the previously downloaded MillingAdvisorSetup.exe file and install to the default “C:\Program Files (x86)\Celeritive Technologies\Milling Advisor\MillingAdvisor.exe” folder.
3. Create a shortcut named “MillingAdvisor” that points to the default install location of the MillingAdvisor in the folder “C:\NCCS”. (Shortcut Target “C:\Program Files\Celeritive Technologies\Milling Advisor\MillingAdvisor.exe”)
 - To create the shortcut navigate to the “C:\NCCS” folder, click the right mouse button and place the cursor over “New” then choose “Shortcut”. Enter the location of the program to create a Shortcut to (Copy and Paste the “Shortcut Target” from step 3 above, (this is the default install location), click “Next”, and make certain the name is “MillingAdvisor” (name must be “MillingAdvisor” for the **NCL** menu pick to work) then click “Finish”.
4. An entry named **VMCalc** will appear on your **NCL 10.1** motion menu the next time **NCL 10.1** is started.
 - The method above has been tested and confirmed to work on all versions of the Windows operating system.