



What's NEW in VERICUT 7.0

IMPORTANT! - Licensing is NOT included in software shipments. See "How to get a license" below for details.

August 10, 2009

Dear VERICUT® User:

Thank you for your continued investment in VERICUT, an important part of your NC programming and machining process!

The VERICUT 7.0's NC program simulation, verification, and optimization technology is packed with new features making it more powerful and easier to use. This letter describes important changes in VERICUT 7.0. Take a moment to review what's new and improved in this release.

Maintenance and Licensing Information

How To Get a License - All users must complete and return the License Request Form in the CD booklet, or submit the application at <http://www.cgtech.com/usa/support/license/>. Licensing is sent via Email only.

NOTE: This software requires a VERICUT 7.0 license.

Software maintenance keeps you on the cutting edge - CGTech provides update software to customers with current software maintenance. Your continued maintenance ensures that you have the most advanced verification technology available. If your maintenance has expired, please contact your CGTech representative (<http://www.cgtech.com/usa/cgtech/contact/>).

Sincerely,

Bill Hasenjaeger

CGTech Product Marketing

VERICUT 7.0

Release Notes

August 10, 2009

VERICUT 7.0 Enhancements

Highlights

The Project Tree is enhanced to reduce the number of pop-up dialogs that the day-to-day VERICUT user must navigate to set up a simulation. Setting up a simulation in VERICUT 7.0 is very different than VERICUT 6.2.

- The Project Tree must be used to configure a project.
- The features formerly in pop-up dialogs that are most commonly used for the day-to-day use of VERICUT are located in the Project Tree.
- An optional Configure menu appears at the bottom of the Project Tree for the selected branch in the Project Tree.
- A new file selection method is built into the Project Tree.
- Actions in the Project Tree Configure menus are applied immediately. There is no OK, Apply, or Cancel button to press.
- Machine components are optionally displayed in the Project Tree, thereby eliminating the Component Tree.

The basic philosophy of the Project Tree enables you to configure a project's setups using the Project Tree features as a guide, as follows:

- Use the Project Tree to step down through the tree structure, configuring each branch of a setup as you go.
- A "Configure" panel is optionally displayed at the bottom of the tree's dialog during setup configuration.
- The "Configure" panel displays settings and actions most commonly used for configuring the setup.
- The features on each "Configure" panel are specific to the branch or item selected in the tree.
- Filling-in text and numeric fields, checking options and pressing buttons in the panel cause an *immediate* action when used. There is no **OK**, **Apply**, etc.
- Less commonly used features are available in the right-mouse button shortcut menus associated with each branch or item selected in the tree or from pop-up dialogs accessed from the VERICUT Menu bar.

VERICUT uses a new method to calculate and animate the motion path from the NC data.

- Animated motion, in all view types, are coordinated.
- Tool images are consistent between views.
- Slow motion and skip cuts are identical for all motions and view types.
- Collision tolerance and motion display are independent of each other.

Machine and Control files are in XML format.

Enhancement Details

Verification

VERICUT is enhanced to support automatic corner rounding and chamfering lines and circles for all cases; approaching/departing, inside/outside, and CW/CCW.

The Toolbar is now fully customizable. You can now add or remove individual icons and define the order in which they are displayed in the toolbar.

Material removal, and Tool display, in the Workpiece view are enhanced to support adding an offset Driven Point (in X or Y) to a Milling Cutter.

Water jet tool display is enhanced in the VERICUT graphics window to display the three different ranges of the tool differently, similar to the way that water jet tools are displayed in the Tool Manager Tool Display area.

Block parsing/processing is enhanced to enable applying a new variable value on the same block in which it was calculated.

The "Preserve Stock Transition" button is moved to the Project Tree's cut stock model's right-mouse button menu and to the cut stock's Configure Model: Assemble tab.

VERICUT is enhanced to enable referencing an NX part file. VERICUT will open and import the part file via a NX/Open application.

VERICUT Machine and Control files syntax is now in XML format.

The Working Directory is now saved in the user preferences file.

The ability to rename a coordinate system in the Project Tree is added to the coordinate system right mouse button menu.

The Project Tree is enhanced to enable specifying G-Code Offsets (Program Zero, Work Offsets, etc.) within the Project Tree.

New feature, "Auto-set working directory to the current project folder", is added to the Preferences window to enable automatically setting the Working Directory to the project file's location.

VERICUT 7.0 Release Notes

New feature, “Suppress save and display thumbnail image”, is added to the Preferences window to enable suppressing the creation of the thumbnail image during the saving of a Project file and suppress the display of the thumbnail in file selection windows (Open, Save As, etc.).

A model highlighted in the Project Tree can now be un-selected by clicking in a blank spot (background) in the VERICUT graphics area.

Right clicking on any file “Save” icon (Save Project, Save Machine, etc.), changes the icon and behaviour to “Save As” mode. The state of the toggle is saved in the user preferences file.

The NC program file name is now written to the VERICUT Log File when an Error or Warning is reported.

Tap error messages are enhanced to enable more quickly identifying the problem (feed speed or tap definition).

"Error: Tap cycle advance 'xxx' is incorrect for tap tool 'yyy' pitch at line 'zzzz'".

Where: 'xxx' = feed distance per revolution.

'yyy' = tool ID.

'zzzz' = line number.

A right mouse button menu has been added to the Message Logger in NC Program Review mode to enable selecting whether Errors, Warnings, or both are displayed. When active (checked) Errors and Warnings are now both associated with the other NC Program Review windows as Errors were in the past.

Collision checking between Tool Holders the Stock/Fixture in a Workpiece View is no longer dependent on the holder being visible. Collision Checking settings and Tool/Holder visibility setting are now independent of each other. Tool/Holder to Stock/Fixture collision settings are now set in the Project Tree, Check Collisions Between menu accessed by clicking on a Collision branch. Cutter/Holder Visibility settings are set in the Project Tree, Configure Tooling menu or from a Tooling branch right mouse button menu.

Design component visibility in a particular view is now based on its based on its Visibility setting (Blank, Machine View, Workpiece View, or Both) except while cutting.

Syntax Check is enhanced to enable checking multiple NC programs at one time. In the Syntax Check window (Info > NC Program, Utilities menu > Syntax Check), The “Check Active” button checks the NC program that is currently “active”. The “Check All” button checks all NC programs, sub-programs, and subroutines referenced by the current setup.

A new function, atan2_d0to360, is added to returns values in the range of 0 to 360 degrees to match the capabilities of Fanuc controls.

atan2_d0to360

atan2_d(yvalue, xvalue)

This function returns the arc tangent of yvalue/xvalue. The returned value is in degrees, and is in the range of 0.0 to 360.0. If yvalue and xvalue are both zero, the return value will be zero.

VERICUT 7.0 Release Notes

A new function, `asin_d270to90`, is added to returns values in the range of 270 to 90 degrees to match the capabilities of Fanuc controls.

`asin_d270to90`

`asin_d270to90(value)`

This function returns the arcsine of the specified value. The specified value must be in the range of -1.0 to 1.0. The return value is in degrees, and is in the range of 270.0 to 90.0.

Dynamic Controls is enhanced to support PowerMill mouse action conventions.

Pan = Shift key + Middle Mouse Button

Zoom In = Mouse Wheel forwards (away from the operator)

Zoom Out = Mouse Wheel backwards (toward the operator)

Dynamic Rotate = Hold Mouse Wheel down

Project Tree is enhanced so that a unique name is assigned when a new setup is added.

The MDI window is enhanced so that once you activate the NC Block Entry text field by clicking in it; it remains active enabling you to continue adding blocks without re-activating it.

A new “Do Not Shorten Cutter” feature is added to Calculate Min. Cutter Extension that that when active, prevents VERICUT from changing the cutter height to be shorter than originally defined in Tool Manager.

The ability to optionally display the tool path line display in a Machine View when in NC Program Review is added.

The cutting motion in both the Workpiece View and the Machine view are now synchronized.

The Animation Speed slider is enhanced to enable controlling animation speed as well as setting Block Skip values.

The Project Tree is enhanced enabling you to unselect models in the Project Tree by clicking on a blank space in the VERICUT graphics window.

The Design component, by default, is only visible in a Workpiece view. It can be made visible in other views.

VERICUT is enhanced to enable assigning a variable and using the variable in the same block.

Design components that are automatically moved to a subsequent setup are now restored to the original setup when the project file is saved.

Fixture collision errors are enhanced to include the fixture component name.

A new Drill Cycle processing option, “No Motion” is added to enable drill cycle processing and material removal without animation.

Machine Simulation

An Ignore column is added to the Machine Settings: Collision Detect tab, similar to the one on the Machine Settings: Travel Limit tab, enabling to toggle checking "on" or "off" for each collision record. The on/off condition of the toggle is only retained during the current VERICUT session.

The Tool Change Retraction table is enhanced to enable controlling the order in which axes move to the tool change location.

Collision Detection is enhanced to highlight the component collision pairs, including sub-components when checked, when a row is selected in the collision table. The components are highlighted by shading them in the "error color". The highlight is removed when the tab is not raised and when the dialog is closed.

The Machine View, right-mouse button, Component Visibility list no longer contains components that do not have models attached.

Machine selection has been enhanced by the addition of the following features:

- When a machine file is saved, a thumbnail image of the contents of the first Machine View is also saved.
- The machine thumbnail image is displayed in the file selection window while browsing to open a Machine file, in the same manner that a thumbnail image is displayed in the file selection window when browsing to open a Project file.
- The machine thumbnail image is displayed when the cursor is held over a machine file in the in the Machine File pull-down list in the Project Tree, Configure CNC menu.

Machine Simulation is enhanced to behave more like the machine for situations where the tool assembly does not include a cutter (spin the holder, continue logging errors, etc.).

Collision checking is enhanced to enable setting a "near miss" value for components and the cut stock.

Collision reporting is enhanced so that if Animation Speed is slow, and Stop At Max Errors is toggled "On", simulation will stop at the intermediate point where the collision occurs.

Tool Manager

The ability to "stack" tool assemblies by default is returned to Tool Manager for all non-turn tool types.

The tool display in the Tool Manager is enhanced to shade the cutting portion of the tool differently than the shank portion of the tool.

VERICUT 7.0 Release Notes

The Turret Aid window is enhanced so that all settings are retained during the current session.

The file selection field is enhanced so that if a file name is specified without a path, Tool Manager searches for the file in the same three default paths that VERICUT uses (Working Directory, parent of Working Directory, Library).

Tool Manager is enhanced to display both “active” and “inactive” OptiPath records associated with the tool.

A “rapid” search tool is added to the Tool Manager Menu bar.

X-Caliper

X-Caliper Distance/Angle output is enhanced to enable measuring the polar angle of a cylinder axis from a defined center point and orientation.

X-Caliper is enhanced to retain the display and orientation of “From”, “To” and measurement markers during dynamic view changes (rotate, zoom in/out, pan).

OptiPath

Depth limit checking is enhanced and a warning message is output if the cutting depth exceeds a specific amount.

G-Code Processing

New macro, GlobalVariable is added to enable sharing variables, or variable ranges, across subsystems.

Right clicking on the Step VCR button displays a new feature enabling you to specify how VERICUT is to handle control subroutines encountered during processing. Choices include Step, Step Into Subroutine, Step Over Subroutine, Step to End of Subroutine.

Macro DynamicToolTipOnOff is enhanced to handle any turning tool, including flash tools with single insert, in the turning plane adding support for dynamic control point during multi-axis turning.

Support is added for true helical material removal when processing G-Codes, including a new X-Caliper feature “helix sweep”.

Two new macros, SetCycleFeedrate and SetCycleFeedrate2cycle, are added to support a main and secondary feedrate that is specific to drill cycles.

The Axis Priority logic is enhanced to allow for multiple groups. Each axis can now have an unlimited number of priorities designated by the group number. By default, each axis has a priority group designated as “0” and may not be deleted. All other groups can be

added or deleted. The priority group can be change at anytime during the simulation via the AxisPriorityGroup macro.

Support is added for multiple line comments. Two Special words: Multiline Comment, defining the comment start (for example, /*) and End Comment, defining the end of the comment (for example, */) must be defined. It is also important to remove the "/" word used as a skip character, and all workaround words, settings, macros etc.

A new capability of adding comments to describe what Word/Value pairs do in the control is added. These comments are also written out to G-Code and Control Report files.

New macros XRelationalOffsetCompName, YRelationalOffsetCompName, ZRelationalOffsetCompName, and RelationalOffsetRegisterName are added to support Relative From/To offsets for axes other than XYZ axes.

Support is added for the Siemens 840D variable type AXIS.

DEF AXIS

DEF AXIS will create a word of type "Special", with subtype "Variable Name". This word will be marked as being created in the current subroutine (or main), and be removed when exiting from this subroutine.

DEF AXIS will also create a variable of type "AXIS" or "axis array". This variable will be marked as being created in the current subroutine (or main), and be removed when exiting from this subroutine. All variables will be treated as global.

The following "AXIS" formats are supported:

```
DEF AXIS name
DEF AXIS name_1, name_2, name_3
DEF AXIS name_4 = X
DEF AXIS name_5, name_6=C, name_7=SET(Z)
```

The following "axis array" formats are supported:

```
DEF AXIS Name[m]
DEF AXIS Name[m] = SET (X, Y, Z,...)
Name[i] = SET(name1, name2[j],...)
Name[i] = REP(name1)
```

The actual axis variable is a string representing the AXIS name and its size is used to allocate the necessary space. Internally the AXIS variable is a string limited to contain only supported axis names.

- Commands SET and REP can be used with both simple axis variable or axis array.
- Parameters for SET, or REP commands can be an axis (constant), a simple variable, or axis array element. No axis expression is allowed.
- Word SET should be added to Word Format table (Type = Special, Sub Type = Sin840D SET).
- Word REP should be added to Word Format table (Type = Special, Sub Type =

VERICUT 7.0 Release Notes

- Sin840D REP).

Rules For Using:

1. DEF must be followed by one or more spaces or tabs.
2. AXIS must be followed by one or more spaces or tabs.
3. The end of the variable name is marked by a space, tab, comma, semi colon.
4. The end of a variable definition is marked by a comma or semi colon.
5. One or more spaces may exist prior to the =.
6. One or more spaces may exist after the =.
7. The left parenthesis will indicate an initial value. The right parenthesis will mark the end of the initial values. Expressions will not be allowed. The constant value must be a valid axis name without any quotes (exactly as it is used with motion commands (X5 Y10)).
8. Supported axis names are all 840D axis (X, Y, Z, A, B, C, U, V, W, A2, B2, C2, X1, Y1, Z1, X2, Y2, Z2, S, S1, S2 and S3). Any axis used as axis variable must be specified in Word/Format table as "Macro" or "Conditional".
9. Since VERICUT supports only 12 axes, the first 12 axes can refer to corresponding axes in VVERICUT. However the NC program can reference any Siemens 840D axis if it is specified in Word/Format table.
10. '[' and '(' must be defined as Left Precedence.
11. ']' and ')' must be defined as Right Precedence.
12. Everything beyond the semi-colon will be ignored.
13. For more details see: Job planning 1-4 Programming Manual, 03/2006 Edition or later.

The concept of axis variable is:

1. Axis name can be stored and referenced without being a string.
2. Numerical array elements can be referenced by axis name instead of index (where X is equivalent to index = 0).
3. Axis variable or constant can be used in logical expression and in function argument list.

VERICUT is enhanced to support these features. Following are some examples.

; Axis definitions:

```
DEF AXIS _AAX = Y ;Y axis assigned to variable _AAX
DEF AXIS _AAY = X ;X axis assigned to variable _AAY
DEF AXIS _AAZ = Z
DEF AXIS _ALLAX[12] = SET(X,Y,Z,A) ;_ALLAX[0-3] initialized to
X,Y,Z,A
```

; supporting definitions

```
DEF REAL ABC
DEF STRING $P_AXNAMS[12] ; string array
DEF REAL $P_AVAL[12] ; REAL array
```

VERICUT 7.0 Release Notes

```
_AAX = _AAY ; X assigned to _AAX
_AAX = X ; X constant assigned to _AAX (no change)
ABC = 5.5
$P_AXNAMS[X]="X" ; string "X" assigned to string array element
$P_AXNAMS[0]
$P_AVAL[Y] = 12.5 ; value 12.5 assigned to array element
$P_AVAL[Y] (or $P_AVAL[1])
_ALLAX[5] = _ALLAX[2] ; Z is assigned to axis array element
_ALLAX[5]
_AAY = _ALLAX[1] ; Y is assigned to _AAY
$P_AVAL[_ALLAX[2]] = 22.5 ; value 22.5 is assigned to $P_AVAL[Z]
(or $P_AVAL[2])
$P_AVAL[_AAX] = 2.5 * $P_AVAL[Y] + $P_AVAL[_ALLAX[2]] ;
$P_AVAL[X] value is 53.75
ABC = 2.5 * $P_AVAL[Y] + $P_AVAL[_ALLAX[2]] ; ABC value is 53.75
```

```
IF (ABC == 5.0); condition not satisfied
ABC = 3.3
ENDIF
```

```
IF (X == _AAX AND Y == _ALLAX[1]); condition satisfied
ABC = 2.2 ; ABC value is 2.2
ENDIF
```

```
GEOAX (ABC, _AAX); function call with axis argument as a variable name
GEOAX (ABC, Y); function call with axis argument as a constant
```

; axis variables and constants in frames

```
$P_UIFR[2] = CTRANS(_AAX,22.5, _ALLAX[1],22.5, _ALLAX[2],145)
$P_UIFR[3] = CTRANS(X,45+22.5*COS(30), Y,22.5, Z,145+22.5*SIN(-
30)):CROT(_AAZ,45):CROT(_ALLAX[1],30)
```

A new conditional function, SiemensAXISCond is added to support AXIS constants in logical expressions when an IF statement is processed. All axes used in the NC program and referred also in logical expressions should be defined in Word/Format table using SiemensAXISCond conditional function and format specifically for particular axis.

Support is added for Siemens 840D AC command for I/J/K on a G2/G3 command while mirroring.

Support is added for processing Okuma turning cycles with Cutter Compensation.

The SyncValue macro is enhanced to apply to all input channels. If a text value of "ALL" is passed, the current input channel will sync with all other input channels.

VERICUT 7.0 Release Notes

New macros MSWriteToFile, MSCloseFile, and MSOpenFile are added to provide a more generic application of the existing FanucDprint, FanucPclose, FanucPopen macros and work in exactly the same way.

New macros AxisMotion and AxisMachineMotion are added to enable redirecting to the appropriate macro implementation (XAxisMotion for AxisMotion with OT="X", AAxisMachineMotion for AxisMachineMotion with OT="A", etc) with the override value passed along. Valid Override Text values are "A", "B", "C", "A2", "B2", "C2", "U", "V", "W", "X", "Y", "Z". Macro AxisMotion should not be used when you need to specify the override text "-" to select the motion direction in absolute mode.

A new override value, 99, is added to macro WorkingPlane2AbcType to provide a "Universal" type similar to the "Universal" type previously implemented for the Ijk2AbcType macro.

A new macro, TravelLimitErrorReporting, is added to enable specifying whether or not travel limit errors are reported during rapid motion.

A new macro, CirclArcAngle, is added to support G2 and G3 with a pitch and angle in Heidenhain MillPlus helical motion.

Three new macros, SetRelWorkCoordFrom, SetRelWorkCoordTo, and SetRelWorkCoord are added to support advance features like RPCP, that require that the work offset table to be relative (ex: Tool to Stock) when the work offset is created from within the NC program.

The G-Code 5-axis motion break-up, created for material removal, is enhanced so that intermediate points have the correct (matrix) orientation in space.

Words A40=, B40= and C40= are added to the HeimPlus Library control and call the IgnoreMacro macro, to prevent them from being interpret as A rotate 40 degrees, etc.

Support for Cycle 204 is added to the Heid530 Library control.

Word CCA is added to the Heid530 Library control and calls the TangentialAngle macro.

G84.1 rigid tapping support is added to the Acramatic 2100 Library control (acr2100.ctl).

Support is added for Polar Interpolation, for situations where C is not at zero, to the Fanuc 15t Library control (fan15t.ctl).

Support is added for Heidenhain Mill Plus G02/G03 helical motion with a pitch.

Miscellaneous

Batch Wizard is enhanced to enable specifying a Working Directory where output files are to be written to.

Batch Wizard is enhanced to enable re-sizing the Main window and the Editor window to be able to see the entire command line.

NC Program Review is enhanced to display the tool path trace in a Machine/Cut Stock View when Animate Machine When Stepping Back is toggled "on" in the Properties window.

All APT Sample files are moved to a separate directory.

Setup Plan is enhanced so that any visible coordinate systems are shown in the Setup Plan report picture capture.

The right mouse button menu for models is enhanced so that Visible and Delete are no longer next to each other.

The tilde symbol “~” is now recognized by VERICUT as a valid APT character.

An option is added to the VERICUT installer to VERICUT Limited shortcuts to be created.

The Display HOSTID utility is enhanced to show up to ten devices and now uses Sentinel LM 8.2.1.

Support is added for 32 bit and 64 bit Vista.

The .bat files in the /commands/ directory are enhanced to enable the recognition of variables defined prior to calling the .bat file.

The tool display in Tool Manager and in Machine views are now updated when the Model Tolerance value changes.

Many sample files are enhanced and new sample files are added.

VERICUT Help has been restructured to focus on the need to use The Project Tree for setting up VERICUT projects.

CATIA V5-to-VERICUT Interface (CATV5)

The CATV5 "Options" dialog is enhanced to enable setting different chordal deviation values for design, stock and fixture models. This is not a tolerance per model, but a value per component type, and applies to all setups.

CATV5 is enhanced to enable selecting STL files used in the CATProcess file as STOCK, FIXTURE, or DESIGN and passing the file reference and position through to VERICUT.

A "Merge to tool library referenced by setup template" option is added to CATV5 to enable merging tools from the current setup with those in the tool library referenced by the Setup Template.

CATV5 is enhanced to enable the Design component to follow the Cut Stock from one setup to the next. If the Design component is selected for the first “active” setup, it is attached to the Stock component and will follow the Cut Stock from one setup to the next. If the Design component is selected in any other setup, it is attached to the Fixture component and will NOT be passed from one setup to the next.

VERICUT 7.0 Release Notes

A check box labeled "Generate one tool library for all setups" is added to CATV5. When toggled "on" (checked), all tools with unique IDs from all setups (active or not) are placed in one library, with its name the same as the new project (but with extension ".tls").

The CATV5 "Options" dialog is enhanced to enable specifying a subsystem for the Work Offset table.

CATV5 is enhanced to keep track of the last "Generate tool library from CATIA information"/"Use specified TDM Tool list" setting as the default setting when accessing a new CATProcess file for the first time.

CATV5 is enhanced to enable passing Tool Holder IDs from the Process Resource List through to VERICUT.

CATV5 is enhanced to enable merging the tools from the current CATProcess file into the tool library referenced by setup template.

MasterCAM-to-VERICUT Interface (MCAMV)

Support is added for MCAMV on Windows XP64.

A French mcRes.local file is now provided with the MCAMV installation.

MCAMV is enhanced to work with MasterCAM X4.

NX-to-VERICUT Interface (NXV)

(formerly known as the Unigraphics-to-VERICUT Interface (UGV))

NXV is enhanced to enable specifying sub-programs.

NXV is enhanced to enable specifying the job subroutines that are to be in the VERICUT session.

NXV is enhanced to support new NX3 tool definitions.

NX4.bat, NX5.bat, and NX6.bat files are added to the commands directories.

Problems resolved in V 7.0

Verification

A problem causing a strange display of a particular facing head in both the Machine and Workpiece views is fixed.

A collision between the Z-axis model and the Cut Stock in a specific VERICUT project file is now correctly reported.

Invalid Tool/Stock collision errors and bad material removal are corrected for a two specific tools with unusual cutter shapes, used for unusual 5-axis motion.

NC Program Review now updates the tool location correctly during a circular motion in a specific VERICUT project file.

Unexpected VERICUT termination no longer occurs during material removal for a specific VERICUT project file using a VERICUT Solid model that was created with a very small tolerance.

A false holder collision is no longer reported for a specific tool holder profile that drops down inside of the cutter profile.

A collision between the holder and the Cut Stock during a drilling cycle in a specific VERICUT project file, displaying only a Workpiece view is now correctly reported.

VERICUT now processes the same number of loops that the machine does for a specific VERICUT user file.

Collisions between the holder and the cut stock are now correctly reported for tool assemblies that have the tool shank inside of concave bottom cutter.

Unexpected VERICUT termination no longer occurs for situations where VERICUT processing runs out of memory due to an invalid cutter profile.

It is now possible to construct a coordinate system on a Cut Stock using the Construct > Circle method. In some cases the pick will fail to resolve a CSYS and you will be prompted to try picking again in a different location.

Control file changes are now saved correctly from inside a specific multi-setup project file.

A lathe threading tool used in a specific VERICUT project file no longer flip-flops around the Z-axis, when the Animation Speed Slider is used to slow down the simulation.

The maximum character limit for IP files in the File Summary > Copy Files window is increased to enable using longer directory paths.

Volume is now correctly reported in the tooling section of a VERICUT Report when FastMill is being used.

Control Settings are now correctly displayed when working with IP files.

VERICUT 7.0 Release Notes

Project file models are now included in the File Summary when an encrypted machine file is being used.

Stop At: End of each Setup now works correctly when the X-Caliper window is open.

Adding a model to a translucent component now displays the model as translucent.

Setting up a VERICUT Report to display the Tool Diameter in the Tool Summary Table now correctly retrieves and displays the tool's diameter value instead of the tool's radius value.

The Field of View Angle slider on the View Attributes window: OpenGL Settings tab, Display Options tab is now correctly activated when Perspective View is toggled on.

Unexpected VERICUT termination no longer occurs when exporting a cut stock as STL file with the Reduce Triangle feature toggled on.

Driven Point values and display are now correct for round inserts.

Having the X-Caliper window open no longer deactivate the Project Tree right mouse button shortcut menus.

Collisions between the holder and the cut stock are now correctly reported when the Animation Speed slider is at 100% for a specific tool assembly that has a tool with an unusual shank shape.

The Cut Stock is now displayed correctly in both the Machine/Cut Stock and Workpiece view regardless of whether the spindle is spinning or not for a specific VERICUT project file.

The jaws no longer spin out of round, on a lathe using programmable jaws, once the spindle is turned on and the jaws are moved.

False Holder/Cut Stock collision errors are no longer reported when the Animation Speed slider is set at 100% for a specific VERICUT project file.

Material removal is now correct for a specific Parametric Cutter used in a specific VERICUT project file.

An error message is now output when trying to use a circular probe motion. Only linear probe motions are supported.

Mirroring a Sweep solid model relative to a coordinate system, now displays correctly when the project file is saved and then re-opened.

The X-Caliper Air Distance value is now correct when measuring after a specific roughing operation.

Material is no longer left behind for a specific tool used for 5-axis cuts around the inner and outer periphery of a pocket.

Unexpected VERICUT termination no longer occurs when outputting a VERICUT report file in HTML format using Japanese characters.

Shank profile removing material errors are now correctly output for a positive Z motion using a specific tool having two shanks. One is the non-cutting portion of the cutter above

VERICUT 7.0 Release Notes

the flute length. The other is a small disk placed in the concave bottom portion of the cutter.

Warning messages, including the Line Number that triggered the warning, is now displayed in NC Program Review message area to enable clicking on a warning message and have VERICUT jump to the location of the warning in the NC program listing and in the graphics area in the same way that clicking on an Error message does.

The collision is now correctly reported when the Collision Detect table record is set to check for collisions between X component and the Z Table component (with sub components) and the Near Miss value is set to zero.

The VERICUT session no longer “hangs” for a specific VERICUT project file when Scan NC Program Files is toggled on in the G-Code Settings window (now the Project Tree, Configure Setup menu: G-Code tab).

Unexpected VERICUT termination no longer occurs after loading a specific IP file and pressing Play without first pressing Rewind NC Program.

A 64 bit VERICUT session no longer hangs when processing a specific VERICUT project file.

Machine Simulation

The machine model visibility, for a specific machine, is now correct when the VERICUT project file is opened a second time.

A collision between a holder and the Stock, with the Stock spinning, for a specific VERICUT project file is now detected correctly.

Using the DwellSeconds macro in a sync job no longer causes a bad move for the head opposite the one that invoked the DwellSeconds macro.

The Driven Point is now correctly positioned on the selected insert of a multi-insert lathe tool regardless of the insert’s position in the tree.

The modified tool list displayed in Tool Manager now reverts back to the original condition when a Save Tool File As action is cancelled.

Milling Tool Wizard no longer permits manually changing the Gage Point when Automatic Gage Offset Z is active. An error message is now output under this condition.

Turret Setup now displays tools in the correct orientation after editing a previously setup turret.

Tool Manager

Support is added for alpha-numerical tools on tool chain when only a cross reference tool list is used.

Holders created as parametric cones now display correctly in the Workpiece view.

X-Caliper

Feature/History records are now correct for helical holes.

The Volume values returned are now correct when using multiple setups.

AUTO-DIFF

AUTO-DIFF results no longer vary for situations where AUTO-DIFF was performed in a prior setup.

During constant gouge check, the color-marked design model indicating gouges/excesses in one setup is now positioned correctly in the next setup.

G-Code Processing

Axis array values can now be successfully used to define a Siemens 840D DEF real array.

New macros RestorePrevMotionType, RestoreSavedMotionType, and SetSavedMotionType are added to support the need to process a G28 command in rapid mode.

A new conditional, TosnucCondVWord is added to enable sharing an axis address as a variable tag.

VERICUT no longer slows, or hangs up, while processing large (large number of records) 2-axis lathe G-Code NC programs when “Replace Material When Stepping Back” is toggled on.

VERICUT now produces the correct circular motion and material removal for situations when in polar interpolation mode, with cutter comp set "on-default to zero" and C rotary changes direction.

The Tool Tip Y value in the Status window is now correct for situations where “Allow Motion Beyond Limit” is toggled off.

Variable round off differences no longer cause incorrect “IF” check results.

Siemens frame functions, CROT and CTRANS now work correctly after an IP file is “merged”.

Material removal when processing a G03 XW circle with ActivateAxis is now correct.

The NewCycleLogic macro is enhanced so that it is now set globally, eliminating the need for it to be called at start of processing for every subsystem.

The “universal” conversion type 99, for the Ijk2AbcType, and the WorkingPlane2AbcType macros are enhanced to use the spindle axis if no tool is loaded.

The “universal” conversion type 99, for the WorkingPlane2AbcType macro is enhanced to ignore the "spindle orient" rotary axis when determining the angles to calculate.

NumSubroutineSequence and similar macros now work with blank text entries.

Rotary Travel Limit Error values are now correct the first time that they are output.

A new Override Text value, “Order”, is added to macros NumSequence and SequenceStartEnd to support a NUM controls ability to create a loop where the Start Sequence block number is higher than the End Sequence block number. The macro NumSequence would pass this ORT value to macro SequenceStartEnd. If macro SequenceStartEnd is entered with ORT = "Order" the sequence start block number and sequence end block number are entered exactly in the order specified in the NC program block.

Miscellaneous

Installation of the Import/Export modules for STEP no longer requires the installation of the optional CATIA support.

The Tool Use graph now correctly displays the program record number.

The Output File Name can now be typed in, as well as using the Browse function, when creating PolyFix Converter batch script with Batch Wizard.

VERICUT is modified to add the directories containing the project and setup template files to the list of directories searched for the machine and control files. This should resolve the problem for NXV and for the other CAM interfaces when VERICUT is started from one of the CAM interfaces and the machine and control files passed to VERICUT do not have directory paths.

VERICUT now consistently uses a yellow background when a menu option field requiring a mouse pick in the VERICUT graphics area is activated.

Coordinate System markers and Axes display are now consistent whether you are in OpenGL mode or in non-OpenGL mode.

Long delays are no longer experienced when loading a specific VERICUT Solid file (.vct) with 64 bit VERICUT.

VERICUT now prompts you to save the project file when only color settings have been changed.

Collision reporting is now consistent regardless of the Animation Slider position for tools that use a single point to define a concave bottom.

Using the new “Pause” feature, VERICUT can now be stopped in the middle of a large rotary motion without killing the process.

Using special characters in Control Notes no longer causes errors.

Creating a report while the Project Tree is open no longer interferes with graphical model selection.

Motion display is now consistent between the Machine view and the Workpiece view when simulating multi-channel machines.

Invalid errors are no longer output for tapping cycles when “Bottom Only” is active.

NC programs can now be activated/inactivated from within the project tree.

CATIA V5-to-VERICUT Interface (CATV5)

Prior CATV5 settings are now correctly retained when CATV5 is opened to start work on a new CATProcess file.

GibbsCam-to-VERICUT Interface (GIBBSV)

The \windows64\ folder now contains the correct Gibbsv.dll.

MasterCAM-to-VERICUT Interface (MCAMV)

MCAMV now works correctly with MasterCAM X4.

NX-to-VERICUT Interface (NXV)

Unexpected NXV termination no longer occurs when it is used with NX4 and 32 bit VERICUT on a 32 bit computer.

NXV, when NX4 is operating in a Team Center environment, now correctly fills the Table List, the 'From' List and the 'To' List pull down lists in the Options window.

NXV no longer outputs a series of error messages when Output Files is selected when the NXV window fields are only partially filled in.

Valid inserts are now created when NX User Defined tools (turning) are passed through NXV to VERICUT.

NXV now successfully passes Work Offset From/To settings to VERICUT.

The NXV "Merge Tools Into Setup Template Tool Library" feature now works correctly.

NXV is enhanced to handle a specific special NX tool definition.

New Macros in V7.0

The following new Macros are added for V7.0.

AxisMachineMotion
AxisMotion
AxisPriorityGroup
CheckVolumeAboveDepthLimit
CircleArcAngle
CollisionPreCheckOnOff
CompositeValue
ConvertArrayIndex
CycleTurnThreadStartAngle
DebugComponentAxis
GlobalVariables
MatchParensInComment
McallSubroutineOn
MSCloseFile
MSOpenFile
MSWriteToFile
RelationalOffsetRegisterName
RestorePrevMotionType
RestoreSavedMotionType
RollerFormCollisionIgnore
RotationPlaneXMirror
RotationPlaneYMirror
RotationPlaneZMirror
SetCycleFeedRate
SetCycleFeedrate2
SetSavedMotionType
TravelLimitErrorReporting
ViperCourseDist
ViperTowCut
ViperTowPinch
ViperTowPrefeed
ViperTowPrefeedDist
XRelationalOffsetCompName
YRelationalOffsetCompName
ZRelationalOffsetCompName

New Conditionals in V7.0

The following new Conditionals are added for V7.0.

TosnucCondVWord

New Functions in V7.0

The following new Functions are added for V7.0.

ISFILE

NOT

Macros not yet included in the documentation

AxisMotion

Function — MOTION

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Text

Text = the axis to be moved.

Sets the axis to be moved ("A", "B", "C", "A2", "B2", "C2", "U", "V", "W", "X", "Y", "Z") then calls the appropriate axis motion macro passing along the Override Text value. For example, if Override Text = B, then macro BAxisMotion is called.

NOTE: Macro AxisMotion should not be used when you need to specify the override text "-" to select the motion direction when in the "absolute" mode.

Also see: AAxisMotion, BAxisMotion, CAxisMotion, A2AxisMotion, B2AxisMotion, C2AxisMotion, UAxisMotion, VAxisMotion, WAxisMotion, XAxisMotion, YAxisMotion and ZAxisMotion

AxisMachineMotion

Function — MOTION

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Text

Text = the axis to be moved.

Similar to AxisMotion except that it sets the axis to be moved ("A", "B", "C", "A2", "B2", "C2", "U", "V", "W", "X", "Y", "Z") then calls the appropriate axis machine motion macro, passing along the Override Text value. For example, if Override Text = B, then macro BAxisMachineMotion is called.

Also see: AAxisMachineMotion, BAxisMachineMotion, CAxisMachineMotion, A2AxisMotion, B2AxisMachineMotion, C2AxisMachineMotion, UAxisMachineMotion, VAxisMachineMotion, WAxisMachineMotion, XAxisMachineMotion, YAxisMachineMotion and ZAxisMachineMotion

CheckVolumeAboveDepthLimit

Function — MISCELLANEOUS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

Value = the maximum allowable volume of material that can be removed by the portion of the cutter above the Maximum Cut Depth
positive value = enables this feature
negative value or 0 = disable this feature

When this macro is called, a “special” volume removed by each cut is calculated by VERICUT. This “special” volume is base on the material removed by the portion of the cutter above the Maximum Cut Depth. When the “special” volume is greater than or equal to the specified value, a Warning message is output to the Logger. Enter a positive value, representing the maximum allowable volume in the Override Value field, to enable this feature. Entering a negative value, or 0, in the Override Value field disables this feature.

This feature only works for “standard” VERICUT (not FastMill). If this feature is enabled when FastMill is active, VERICUT will internally turn off FastMill mode.

If OptiPath is turned on while this feature is enabled, the OptiPath results will be based only on the material removed by the portion of the cutter below the Maximum Cut Depth.

Use of this macro requires the following:

1. **Check Cutting Limits** in the Project Tree, Setup Configure menu: Motion tab must be toggled “on”.
2. Maximum Cut Depth on the OptiPath window: Limits tab must be enabled.

CircleArcAngle

Function — MOTION

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

Value = the arc angle specified as a full angle of rotation

Use to enable specifying a helical/circular motion by a full angle of rotation. For example, the Heidenhain 530 control uses the “B5=” field to specify the arc angle as shown below:

```
G01 Z9. F100  
X30 Y30 F500  
G2 I40 J20 K-8 B5=420
```

CollisionPreCheckOnOff

Function — MISCELLANEOUS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

0 = Off (default)

1 = On

If the collision pre-check option is turned on, we will first do the collision check using an automatically created simplified primitive. If this check indicates that there might be a collision, the collision check will be done with the original primitives.

Currently the default for the collision pre check is Off. Theoretically, if the pre check always causes the collision check to be done with the original primitives, then it would be faster not doing the pre-check. If collision checking performance is slow on the job that you are working on, you can try turning the pre-check on, and see if this helps the performance.

CompositeValue

Function — MISC

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Text, Value

Text = composite format : variables to which the individual components are assigned

Value = the composite value

Use this macro to breakdown a composite value into its individual components, and store each component in the variable specified. The text value contains the composite format (for example: 2 3*.2), followed by a ':', followed by a series of variable names (up to 10) indicating where to store the individual components. All arguments are separated by spaces.

If the breakdown of the composite value does not result in an individual component for a specific variable, then that variable will be set to zero as shown in EXAMPLE 2 below.

EXAMPLE 1:

Override Value = 1234567.89

Override Text = 2 3*.2 : VAR1 VAR2 VAR3 VAR4 (Note the space between the 2 and the 3)

2 3*.2 : VAR1 VAR2 VAR3 VAR4 is interpreted as:

Put the first 2 characters of the composite value (12) in VAR1.

Put the next 3 characters of the composite value (345) in VAR2.

Put all remaining characters up to the decimal point (67) in VAR3.

Put the characters after the decimal point, in this case up to 2 (89) in VAR4.

The variables would be set as follows:

VAR1 = 12

VAR2 = 345

VAR3 = 67

VAR4 = 89

EXAMPLE 2:

Override Value = 12.1

Override Text = 2 3*.2 : VAR1 VAR2 VAR3 VAR4

2 3*.2 : VAR1 VAR2 VAR3 VAR4 is interpreted as:

Put the first 2 characters of the composite value (12) in VAR1.

Put the next 3 characters of the composite value (none) in VAR2.

Put all remaining characters up to the decimal point (none) in VAR3.

Put the characters after the decimal point, in this case up to 2 (1) in VAR4.

The variables would be set as follows:

VAR1 = 12

VAR2 = 0

VAR3 = 0

VAR4 = 1

ConvertArrayIndex

Function — EVENTS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

0 = do not apply the AXIS variable conversion

1 = apply the AXIS variable conversion to index (default)

This macro is used to enable/disable the conversion of the array index, specified as AXIS variable, or the constant in the argument string specified with the **SetDynamicVars** macro. If this macro is not used, or called with value of 1, all array references in argument string are decoded when the AXIS variable/constant is used. If a value of 0 is specified, the old parsing logic is used where the array element can only be referenced by numerical index.

Also see: **SetDynamicVars**

CutterCompOnOffSwitches

Function — CUTTER COMPENSATION

Status — ACTIVE

Comment — Added V6.2.2

Valid Inputs — Text, Value

This macro is used to set the flags which determine how to process the turning on and turning off of Cutter Compensation.

Value: Specifies the option you want for a specific scenario as defined by the Text argument. The valid options are specific to the specific scenario selected. Valid Options are:

Text = 1

Value = 1 → Ramp On

Value = 2 → Immediate

Text = 2

Value = 1 → Ramp On with Active Plane Motion

Value = 2 → Immediate

Value = 3 → Ramp on with Any Motion

Text = 3

Value = 1 → Ramp On with Active Plane Motion

Value = 2 → Immediate

Value = 3 → Ramp On with Any Motion

Text = 11

Value = 1 → Adjust the input coordinates to match the physical location of the tool

Value = 2 → Do not Adjust the input coordinates to match the physical location of the tool

Text: Specifies which flag (scenario) to set:

1 = Turning on Cutter Compensation with motion within the plane

2 = Turning on Cutter Compensation with no motion

3 = Turning on Cutter Compensation with the only motion being outside of the plane

11 = Turning off Cutter Compensation

CycleTurnDiameter**Function** — TURNING CYCLES**Status** — ACTIVE**Comment** — Added V6.2.2**Valid Inputs** — None

Macro **CycleTurnDiameter** sets the turning cycle type to diameter. The current position is used as the diameter start point. When X and/or Z words are included on the block they define the endpoint of the canned diameter. When U and/or W words are included on the block, the associated U-W values are interpreted as signed incremental values from the start point to the end point. The **XaxisIncreMotion** and **ZaxisIncreMotion** macros must be called when U and W are used to define the canned diameter endpoint.

The **CycleTurnDiameter** macro is modal until canceled by one of the following macros: **MotionRapid**, **MotionLinear**, **MotionCW**, **MotionCCW**, **MotionNurbs**, **MotionPoly**, or **Motion3DCircle**.

The G90, canned diameter cycle remains active allowing additional blocks of the G90, cycle to be simulated until canceled by another G-code.

A G90, word address is required to call the **CycleTurnDiameter** macro. The associated **CycleTurn*** macros *DO NOT* have to be called for each block.

For example:

```
G90 X2.5Z-2R0.1
X2.4
X2.3
X2.2
X2.1
X2.0
G00X3.0Z1.0
```

CycleTurnTaper**Function** — TURNING CYCLES**Status** — ACTIVE**Comment** — Added V6.2.2**Valid Inputs** — Value

Specifies the distance of taper in X-axis direction for a diameter canned cycle (signed value, radius).

CycleTurnThreadOff

Function — TURNING CYCLES

Status — ACTIVE

Comment — Added V6.2.2

Valid Inputs — Text, Value

The macro enables you to define a set of word/value pairs in the OVERRIDE TEXT field which are used to signify when the turning cycle modal is turned off. For example, the word/value pairs entered in the OVERRIDE TEXT field might be as follows: "G00, G01, G02, G03". This provides full control for cancelling turning cycle modals.

CycleTurnThreadStartAngle

Function — TURNING CYCLES

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

Value = the thread starting orientation

This macro sets the thread starting orientation (0 to 359.999 degrees) for multiple threads in a general thread turning cycle.

DebugComponentAxis

Function — MISCELLANEOUS

Status — SPECIAL

Comment — Added V6.2.2 - This macro is intended for internal CGTech use only.

Valid Inputs — Text

This macro prints the location of the specified component.

GlobalVariables

Function — VARIABLES

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Text

This macro defines the ranges for global variables. NOTE: This macro only applies to positive integer variables.

The format for the TEXT value is:

-30, 50-70, 90, 300-400, 500-

The above line would define the following variables as global:

- All variables 30 and below.
- 50 through 70
- 90
- 300 through 400
- All variables 500 and above

This macro now makes SetMaxSubsystemVar and SetMinSubsystemVar obsolete.

MatchParensInComment

Function — MISCELLANEOUS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

0 = a comment defined by parentheses will close on first ')' (default)

1 = a comment will not be closed unless *all* opening parentheses "(" are matched by closing parentheses ")"

This macro is used to specify how VERICUT should determine when the end of a comment is reached. It should be called at the start of processing.

If "(" is defined as a "Begin Comment" word in the Word Format window: the comment will end on the first ")" if the Override Value = 0, or the matching ")" if the Override Value = 1; however the comment will always end on End of Block.

EXAMPLE 1

"N0004 (THIS COMMENT (1.000) HAS MATCHED PARENTHESES)"

the comment will end on the first ")" if the Override Value = 0,

"N0004 (THIS COMMENT (1.000))"

or by the second ")" if the Override Value = 1.

"N0004 (THIS COMMENT (1.000) HAS MATCHED PARENTHESES)"

EXAMPLE 2

"N0004 (THIS (COMMENT (1.000) HAS UNMATCHED PARENTHESES) SO WHAT HAPPENS?"

the comment will end on the first ")" if the Override Value = 0,

"N0004 (THIS (COMMENT (1.000))"

End of Block if the Override Value =1.

"N0004 (THIS (COMMENT (1.000) HAS UNMATCHED PARENTHESES) SO WHAT"

If "(" is defined as a "Multiline Comment" word in the Word Format window, the End of Block is irrelevant:

EXAMPLE 3

"N0004 (THIS (COMMENT (1.000) HAS UNMATCHED PARENTHESES) SO WHAT HAPPENS?)"

the comment will end on the first ")" if the Override Value = 0,

"N0004 (THIS (COMMENT (1.000)"

or on the matching ")" if the Override Value = 1.

"N0004 (THIS (COMMENT (1.000) HAS UNMATCHED PARENTHESES) SO WHAT HAPPENS?)"

McallSubroutineOn

Function — SUBROUTINES

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Text, Value

This macro is used to save the subroutine name and retain the parameters upon returning from the subroutine. McallSubroutineOn should be called when a Siemens 840D **MCALL** *subroutine name* command is encountered. It should be followed by a call to the **Siemens840DProcCall** macro. An **MCALL** command on a block by itself should call the **CycleSubroutineOff** macro.

How it works:

MACALL *subroutine name* passes the subroutine name and parameters to the McallSubroutineOn macro which saves the subroutine name and parameters so that they are available upon returning from the subroutine. Siemens840DProcCall macro invokes the call to the subroutine. When an **MCALL** command is the only thing on a block, the **CycleSubroutineOff** macro is called to cancel the above actions.

EXAMPLE:

```
X50 Y0 Z55
MCALL CYCLSUB(60,50,2,35)
Y30
MCALL
```

In the above example, the CYCLSUB(60,50,2,35) subroutine is called twice before the **MCALL** block cancels the above sequence.

MSCloseFile

Function — MISCELLANEOUS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — None

This macro is designed to support commands similar to the Fanuc PCLOS command. This macro closes the external text file opened using macro MSOpenFile. This macro should be called after processing the last MSWriteToFile command, or before the end of the NC program. Macros MSOpenFile and MSCloseFile should always be used in pairs.

MSOpenFile

Function — MISCELLANEOUS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Text

This macro opens an external text file for the output provided by the MSWriteToFile macro. The Override Text field is used to specify the name of the output file. If the file name is specified without a path, it will be located in the VERICUT Working Directory. If the specified file exists, the formatted string output by MSWriteToFile is appended to its contents. This macro must be used before any MSWriteToFile macro is processed. Note that only one external file can be opened at any time. The file should be closed, using macro MSCloseFile, after the last MSWriteToFile macro, or before the end of the NC program. Macros MSOpenFile and MSCloseFile should always be used in pairs.

MSWriteToFile

Function — MISCELLANEOUS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — None

This macro is used to format, and output, NC program variables similar to the Fanuc DPRNT command. The only input data for this macro is the text following the DPRNT command. The variables and their format are specified in DPRNT statement. See Fanuc 15i programming manual page 659 – 663 for details. The Fanuc system variable #7000 (bit 7) can be used to control the current format design (see Fanuc manual for details). The formatted string is output to an external file opened using macro MSOpenFile.

RelationalOffsetRegisterName

Function — TOOL_OFFSETS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Text

This macro has been created as an alternative way to specify the component in which a relational offset should be applied. Rather than calling one of the RelationalOffsetCompName macros, this macro can be called to specify the Register name that you want the offset to apply to. For example: If your machine is defined with both a X and a U register component, both moving along the X axis, if you call this macro with an OT=U, then the X relational offset will be applied to the U register component.

If a text value of “RESET” is specified, the direct specification of the registers used for X, Y, Z Relational offsets will be deleted, and the software will apply X to X, Y to Y and Z to Z.

RestoreMotionType

Function — MOTION

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — None

This macro restores the motion type to the type of the last non-rapid motion. This macro is only applicable when in RAPID mode, and you want to switch back to the previous non-RAPID mode. If the current motion type is set to something other than RAPID, this macro does not restore a previous mode, and will keep the current motion type.

RestorePrevMotionType

Function — MOTION

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — None

This macro is used to restore the previous motion type.

NOTE: Commands like ToolChange, CyclesExecute and CyclesCancel may change the motion type.

RestoreSavedMotionType

Function — MOTION
Status — ACTIVE
Comment — Added V7.0
Valid Inputs — None

This macro restores the "saved motion type". The "saved motion type" is set by calling the **SetSavedMotionType** macro.

RollerFormCollisionIgnore

Function— TAPE LAYING
Status — ACTIVE
Comment — Added V7.0
Valid Inputs — Value

0 = turn off the ignore flag
1 = turn on the ignore flag

This macro enables, and disables, the normal collision checking that would occur between the active tool (Roller) and the active stock (Form). It does this by modifying the ignore flag on the corresponding entry in the collision table.

NOTE: The corresponding entry must already exist in the collision table.

A value of 0 turns off the ignore flag, and a value of 1 turns on the ignore flag.

NOTE: The ignore flag was implemented in V7.0. It is being back ported into V6.2.x to support this command, but without the corresponding GUI update to the collision table.

RotationPlaneXMirror

RotationPlaneYMirror

RotationPlaneZMirror

Function — MIRROR
Status — ACTIVE
Comment — Added V7.0
Valid Inputs — none

These macros are used to add mirroring support to Siemens 840D \$P_PFRAME. These macros are equivalent to the **MirrorX**, **MirrorY** and **MirrorZ** macros where the mirror axis value is 0. When one of these macros is called, the corresponding axis will be reversed. Circular interpolation is not supported when mirror is activated by one of these macros.

SetCycleFeedrate

Function — DRILL CYCLES

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

This macro sets the feedrate which is used during a drill cycle while moving from the Rapid plane to the final Depth level. This Feedrate is independent of the normal feedrate used when not executing a drill cycle. The default value is zero, which means to use the normal feedrate. This value is modal. This means, it might need to be reset to zero when the drill cycle is defined (G81, G82, ...).

SetCycleFeedrate2

Function — DRILL CYCLES

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

Sets the secondary feedrate which is used during specific drill cycles while moving from the Final Depth level back to the Rapid plane. This Feedrate is independent of the normal feedrate used when not executing a drill cycle. The default value is zero, which means to use the main cycle feedrate (See SetCycleFeedrate). This value is modal. This means, it might need to be reset to zero when the drill cycle is defined (G81, G82, ...).

NOTE: This feedrate only applies to BORE, BORE DRAG, and TAP cycles.

SetSavedMotionType

Function — MOTION

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — None

This macro is used to set the "saved motion type" which is used by the RestoreSavedMotionType macro.

TravelLimitErrorReporting

Function — MISCELLANEOUS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

Value = error reporting mode

Use to define the travel limit error reporting mode. An override value of "0" produces error reporting for all motion (default). An override value of "1" prohibits error reporting during a rapid motion.

ViperCourseDist

Function — Tape

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

This macro is specific to the Cincinnati Viper Tape machine. It defines the instantaneous distance for the current course (since starting to lay this piece of tape).

NOTE: Although this appears to be a fairly generic value, internally, this value is important to various calculations specific to Viper.

ViperTowCut

Function — Tape

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

This macro is specific to the Cincinnati Viper Tape machine. It defines the tows that are currently being cut, or has been cut. Tows that are currently being cut will run out in distance specified by the **ViperTowPrefeedDist** macro.

For example: T2063

Each bit is associated with a tow. 2063 converted into base 16 is: 80F This value corresponds to 4 tows on the low side, and 1 tow on the high side.

ViperTowPinch

Function — Tape

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

This macro is specific to the Cincinnati Viper Tape machine. It defines the tows that are to be restarted. The tow will actually start being laid in the distance specified with the **ViperTowPrefeedDist** macro.

For example: Sa2048

Each bit is associated with a tow. 2048 converted into base 16 is: 800. This value corresponds to tow 12.

ViperTowPrefeed

Function — Tape

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Text

This macro is specific to the Cincinnati Viper Tape machine. It defines the distance that a tow should be pre-fed prior to the "headon" command. Subtracting the specified value from the **ViperTowPrefeedDist** will give you the distance the head will need to travel before the tow actually starts laying tape.

The specific tow is referenced by A-M, O-Z, and a-g

Example: Text value = "A2.9B2.9C.5D2.5E2"

ViperTowPrefeedDist

Function — Tape

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Value

This macro is specific to the Cincinnati Viper Tape machine. It sets the distance from the point where the tape is cut to the point where tape is pressed on to the part. The default is 2.9 inches.

See also: **ViperTowCut**.

VirtualZAxisADynamic

Function — Miscellaneous

Status — Active

Comment — Added V6.2.2

Valid Inputs — Value

0 = On

1 = Off

This macro turns on, and off, the dynamic calling of the **VirtualZAxisARotary** macro. If the ABC angles are being calculated from IJK points, then the **VirtualZAxisARotary**

macro needs to be called after the A angle has been set. This is done by calling this macro. A value of 0 turns this feature on, a value of 1 turns this feature off.

VirtualZAxisARotary

Function — Miscellaneous

Status — Active

Comment — Added V6.2.2

Valid Inputs — None

Establish a virtual Z axis using rotation planes. The input angle is defined by the local A-axis value. The **AAxisMotion** macro is used to specify incremental/absolute, and must be called prior to calling this macro. This angle defines the angle at which the real Z-axis is offset from the virtual orthogonal Z axis.

XAxisMultiplier

Function — MOTION

Status — ACTIVE

Comment — Added V6.2.2

Valid Inputs — Value

Value = Multiplier to be applied to the **XAxisMotion** macro

This macro defines a multiplier that is used by the **XAxisMotion** macro. This multiplier is currently only used by the **XAxisMotion** macro. Typically, this macro will be called when transitioning between programming in radius mode and diameter mode.

If you have a .5 multiplier on the X word or the Word/Address entry, then this macro would be pass a value of 2 when changing to radius mode, and a 1 when changing to diameter mode.

If you do not have a multiplier on the X word or the Word/Address entry, then this macro would be pass a value of 1 when changing to radius mode, and a .5 when changing to diameter mode.

The default value is zero, which means to ignore the multiplier.

NOTE: If a multiplier is defined, and Debug Macro Arguments are turned on, you will see the following results:

Debug: MACRO: XAxisMotion, WORD:X, TEXTSTR=5., VALUE=5

Debug: ADJUSTED X VALUE=2.5 (XAxisMultiplier)

XRelationalOffsetCompName

Function — TOOL OFFSETS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Text

Specifies the name of the component to which the "X offset" is applied for offsets specified as relational (using Select From/To Locations).

YRelationalOffsetCompName

Function — TOOL OFFSETS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Text

Specifies the name of the component to which the "Y offset" is applied for offsets specified as relational (using Select From/To Locations).

ZRelationalOffsetCompName

Function — TOOL OFFSETS

Status — ACTIVE

Comment — Added V7.0

Valid Inputs — Text

Specifies the name of the component to which the "Z offset" is applied for offsets specified as relational (using Select From/To Locations).

Functions not yet included in the documentation

ISFILE (string)

It is used to check if the specified string refers to an existing file in the system. If file is found the function returns a 1 (TRUE), otherwise the function returns 0 (FALSE). The function checks the current working directory, project directory and library trying to find specified file. The default file extension is MPF. The string argument can contain the file path if necessary.

NOT value

It is used to check if an argument (value) is equal to 0 and returns 1 (TRUE), otherwise returns 0 (FALSE).

VERICUT 7.0 Release Notes



9000 Research Drive, Irvine, CA 92618 • www.cgtech.com • Tel: (949) 753-1050 • Fax: (949) 753-1053 • Email: support@cgtech.com